Computing and Informatics, Vol. 44, 2025, 915–932, doi: 10.31577/cai_2025_4_915

MODELING AND VERIFICATION OF CHINESE WALL POLICY BASED ON PETRI NETS WITH DATA

Hanqian Tu, Dongming Xiang*

Department of Computer Science and Technology
Zhejiang Sci-Tech University
310018 Hangzhou, China
e-mail: alex_thq@163.com, flysky_xdm@163.com

Liang Qi

Department of Computer Science and Technology
Shandong University of Science and Technology
266590 Qingdao, China
e-mail: qiliangsdkd@163.com

Guanjun Liu

Department of Computer Science
Tongji University
201804 Shanghai, China
e-mail: liuguanjun@tongji.edu.cn

Abstract. Information security is an important issue in the design and development of information systems. As a well-known information security policy, Chinese Wall policy concerns the conflict of interest among sensitive information items. Since it is widely applied in many fields, it is important to explore the verification methods. Petri nets are a widely used formal method in the modeling and verification of information systems, and they are suitable for verifying Chinese Wall policy due to the capability of characterizing the concurrency. Particularly, some stud-

* Corresponding author

ies utilize colored Petri nets for modeling and verification of Chinese Wall policy. However, they do not characterize data operations including read, write and delete, which may affect the verification results. In this paper, we utilize Petri nets with data (PD-nets) to model and verify this policy. Specifically, we propose PD-nets for Chinese Wall policy to depict the control-flows, data-flows and data operations of information systems and introduce configurations and reachability graphs to describe the running states. We give theorems to prove the correctness of our method. Based on these theorems, we develop an algorithm to detect the violations of Chinese Wall policy. Furthermore, a case study is presented to show the effectiveness of our method, especially in modeling data operations and verifying their relevant CW policy.

Keywords: Petri net, information security, model checking, Chinese Wall policy, reachability graph

Mathematics Subject Classification 2010: 68-Q60

1 INTRODUCTION

Information security is a crucial issue in information systems and business processes [1]. When it is compromised, the system reliability is also jeopardized. Derived from British laws concerning conflict of interest in business and finance, *Chinese Wall (CW) Policy* [2] relates to the confidentiality and integrity of information security [3]. Unlike traditional multi-level security policies (e.g., Bell-LaPadula model [4]), CW policy focuses on the conflict of interest among information items, and restricts access to them. Specifically, CW policy prohibits the same individual from accessing sensitive information items in conflict of interest [5]. For example, Figure 1 shows the violation of CW policy, where there exist two competing companies A and B. An individual named Alex has access to two sensitive information items a and b that are in conflict of interest, which potentially leads to unfair competition or bribery.

CW policy is crucial in the information security of many research fields, e.g., cloud services and distributed systems. Tsai et al. [6] propose a centralized control mechanism in cloud computing based on CW policy so as to eliminate the possible Inter-VM Attacks from competitors. Basu et al. [7] present a formal cloud model based on Z-notation, and apply CW policy to design secure cloud-specific operations. Alqahtani et al. [8] employ the principles of CW policy, and introduce a strategy to store and audit conflict of interest classes for cloud services. Anupa and Sekaran [9] implement CW policy in workflow management systems with role-based access control, and illustrate how to apply CW policy to various layers in cloud computing service models. Fehis et al. [10] propose a new CW policy model for distributed systems to deal with some previous limitations.

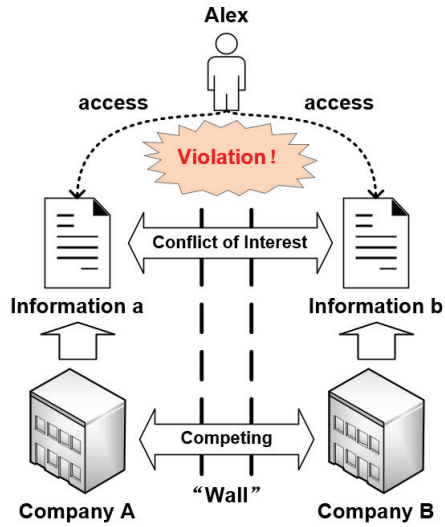


Figure 1. The violation of CW policy

Since CW policy is widely applied in the information security, its verification becomes significant in information systems. As a commonly used model-checking [11] method, Petri nets [12] are well-suited for modeling and verifying CW policy due to their capability of characterizing parallelism, concurrency and synchronization [13, 14]. Zhang et al. [3] employ colored Petri nets (CPN) to model CW policy, and utilize the coverability graph for formal analysis. Huang and Kirchner [15] propose a CPN-based approach to verify modular security policies, and apply it in the design verification of CW policy. Tu et al. [16] use unfolding techniques of CPNs to alleviate the state space explosion problem in detecting information leakage against CW policy. In general, the existing methods depict the data-flows with colored Petri nets. However, they do not consider data operations, which can have an effect on the verification of CW policy. This is crucial because data operations may change the states of data, e.g., modifying values, incorporating new information or deleting certain content. These changes can significantly affect users' access to information. Therefore, it is imperative to account for data operations when verifying CW policy to ensure information security. By contrast, Petri nets with Data (PD-nets) [17] introduce the formalization of data operations (e.g., read, write and delete), and thus properly describe both the control-flows and data-flows of information systems.

In this paper, we present a novel method for modeling and verifying CW policy based on PD-nets. Figure 2 shows the architecture of our method. The main contributions are summarized as follows.

1. We incorporate some elements (e.g., subjects, sources and conflict of interest)

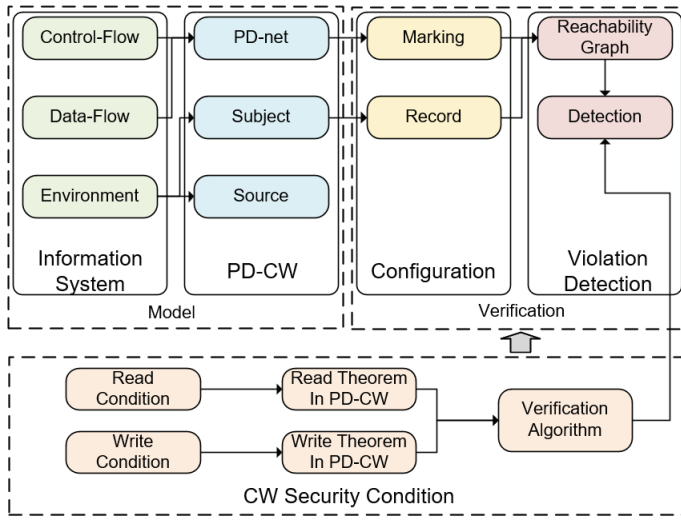


Figure 2. The architecture of our method

of CW policy into PD-nets, and define a new model (PD-CW) to formalize the control-flows, data-flows and data operations of information systems;

2. We introduce configurations and reachability graphs to characterize the state space of PD-CW, which take into account both system states and access records; and
3. We propose some theorems, which involves the reading and writing security of CW policy in PD-CWs, to check whether PD-CW complies with CW policy. Moreover, an algorithm is developed to detect the violations within PD-CWs.

This paper is organized as follows. Section 2 introduces some preliminaries. Section 3 presents our method for modeling and verifying CW policy. Section 4 gives a case study. Section 5 concludes this paper.

2 PRELIMINARIES

Some definitions related to PD-nets and CW policy are given in this section.

2.1 PD-Nets

When modeling and verifying CW policy, both control-flows and data-flows of information systems should be considered. However, ordinary Petri nets can only describe the control-flows. In order to describe data-flows, we introduce *D-net* and *PD-net*.

Definition 1 (D-net [17])**.** A tuple $N = (P, T, F, D, Read, Write, Delete)$ is a *net with data* (*D-net*), where:

1. (P, T, F) is a *net* [18], where the set of places P and transitions T are finite and disjoint, and $F \subseteq (P \times T) \cup (T \times P)$ is the arc set;
2. D is a finite set of data elements;
3. $F \subseteq (P \times T) \cup (T \times P)$ is a flow relation.
4. $Read : T \to 2^D$ is a label function of reading data;
5. $Write : T \to 2^D$ is a label function of writing data; and
6. $Delete : T \to 2^D$ is a label function of deleting data.

For each node $x \in P \cup T$, its *pre-set* and *post-set* are denoted by ${}^\bullet x = \{y|(y, x) \in F\}$ and $x^\bullet = \{y|(x, y) \in F\}$, respectively. A *marking* M is a function: $P \to \mathbb{N}$ where $\mathbb{N}$ is the set of natural numbers. A D-net N with a initial marking M_0 is a *Petri net with data* (*PD-net*) [17], denoted as $\Sigma = (N, M_0)$. Figure 3 a) shows a PD-net, where a, b and c are three data elements. The labels next to the transitions indicate the data operations, e.g., t_0 writes a and t_4 reads b. The initial marking of this PD-net can be denoted as a vector $M_0 = [1, 1, 0, 0, 0, 0, 0, 0]$.

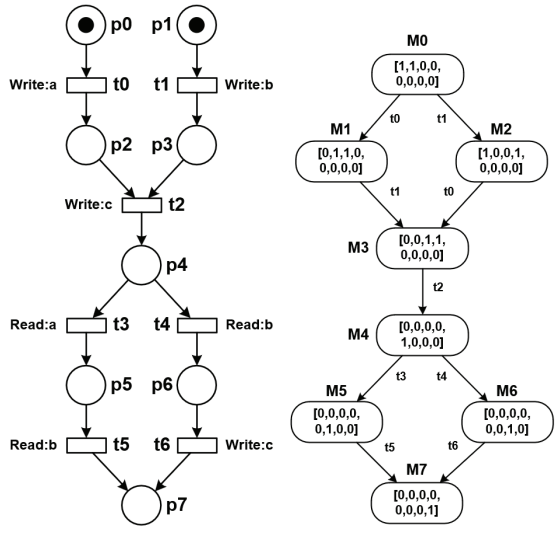


Figure 3. a) a PD-net; b) reachable markings of a)

A transition $t \in T$ is *enabled* at a marking M if $\forall p \in^\bullet t$: $M(p) > 0$, denoted as $M[t\rangle$. An enabled transition t *fires* at M and a new marking M' is generated after

firing, denoted as $M[t\rangle M'$, where $\forall p \in P$:

$$M'(p) = \begin{cases} M(p) + 1, & \text{if } p \in^\bullet t \setminus t^\bullet, \\ M(p) - 1, & \text{if } p \in t^\bullet \setminus^\bullet t, \\ M(p), & \text{otherwise.} \end{cases} \quad (1)$$

A sequence of transitions σ is *enabled* at a marking M, denoted by $M[\sigma\rangle$, if the transitions in σ can fire sequentially, and σ is called *firing sequence*. After the firing of σ at M, it can generate a new marking M', denoted as $M[\sigma\rangle M'$. A marking M' is *reachable* from M if there exists a firing sequence σ such that $M[\sigma\rangle M'$. For example, Figure 3 b) shows some markings of 3 a), where M_0 is the initial marking. According to the firing rule, M_{1-7} are reachable from M_0, e.g, M_4 can be reached by firing the sequence $\sigma = [t_0, t_1, t_2]$ at M_0. We say that M_{0-7} are *reachable markings* of the PD-net in Figure 3 a).

2.2 Chinese Wall Policy

The main goal of CW policy is to separate sensitive information from competing *sources* (e.g., companies, corporations and organizations) [6]. In CW policy, an *object* [2] o refers to an information or data item of a certain source, and its source is denoted as $CD(o)$. For example, companies A and B in Figure 1 are sources of objects a and b, respectively. A *subject* [2] refers to an individual, group or entity involved in the information interaction and system activities. Thus, subjects usually have access to objects depending on the running states of information systems. The set containing all the objects that a subject s has access to is denoted as $PR(s)$.

Due to the security requirement of CW policy, a subject cannot have access to two objects from two competing sources. We say that these two objects are in *conflict of interest* [2]. This relationship among objects can be described by a binary symmetric relation *CIR* [19], where $(o_1, o_2) \in CIR$ indicates that o_1 and o_2 are in conflict of interest.

In order to prevent information leakage against CW policy, the following conditions are given to illustrate which objects can be read or written to by a subject.

Definition 2 (CW Security Condition [3])**.** Let S and O be the set of subjects and objects, respectively.

1. *CW-Simple Security Condition*: A subject $s \in S$ can read an object $o \in O$ if and only if one of the following requirements holds:
   - $\exists o' \in PR(s) : CD(o) = CD(o')$; or
   - $\forall o' \in PR(s) : (o, o') \notin CIR$.
2. *CW-* Security Condition*: A subject $s \in S$ can write to an object $o \in O$ if and only if both of the following requirements hold:

- s can read o; and
- $\forall o' \in PR(s) : CD(o) = CD(o')$.

The CW-Simple Security Condition and CW-* Security Condition indicate the reading and writing restrictions of subjects in CW policy, respectively. If either of these two conditions are violated, CW policy is considered violated. Figures 4, 5 show the examples of violating CW-Simple Security Condition and CW-* Security Condition, respectively. In Figure 4, the subject reads both objects a and b that are in conflict of interest, which violates CW policy. In Figure 5, a subject reads the object a and writes its information to c. Then, another subject reads objects b and c, where a and b are in conflict of interest. Since c contains the information of a, this subject can indirectly read a and b, which violates CW policy.

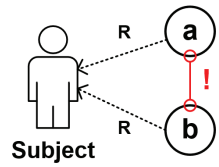


Figure 4. The violation of CW-Simple Security Condition

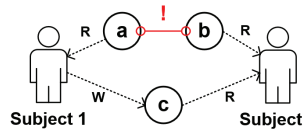


Figure 5. The violation of CW-* Security Condition

3 MODELING AND VERIFICATION OF CW POLICY

Based on PD-nets, we explore the method for modeling and verifying CW policy in this paper. Specifically, we propose a PD-CW model, define its configuration and reachability graph, and develop an algorithm to verify CW policy.

3.1 PD-Net for CW Policy

PD-nets can describe the control-flows, data-flows and data operations of information systems. However, we still need the formalization of subjects and sources so as to verify CW policy. Therefore, we propose *PD-net for CW Policy* (*PD-CW*) based on PD-nets, which involves subjects, sources and *CIR* for the verification of CW policy. This model indicates the performers of system activities, so that we can analyze the behaviors of subjects and verify CW policy.

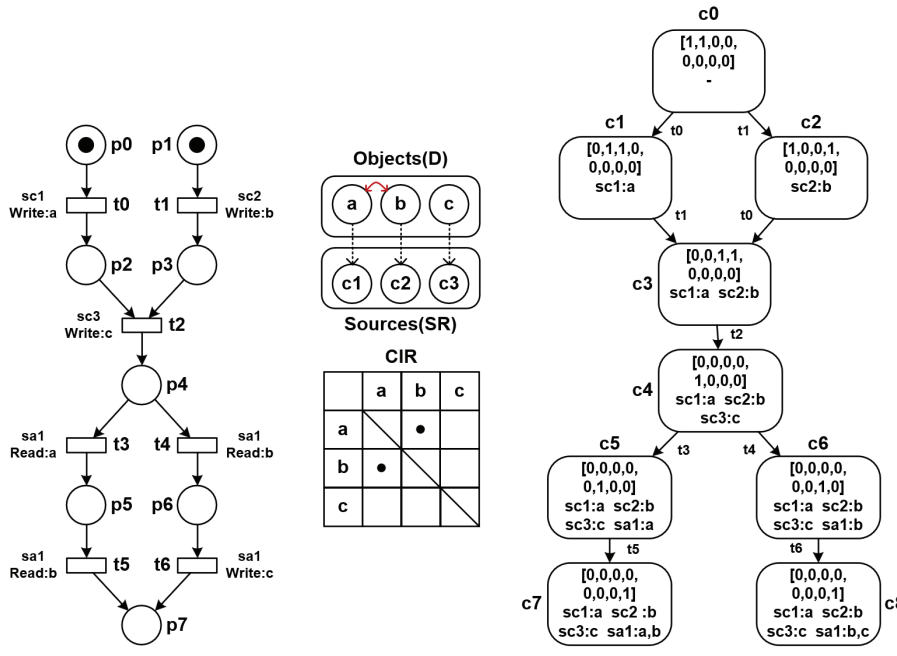


Figure 6. a) a PD-CW; b) objects, sources and CIR of a); c) the reachability graph of a)

Definition 3 (PD-net for CW Policy, PD-CW)**.** A tuple $(\Sigma, S, T_S, SR, CD, CIR)$ is a *PD-net for CW policy* (*PD-CW*), where:

1. Σ is a PD-net;
2. S is a finite set of subjects;
3. $T_S : T \to S$ specifies the subject of each transition;
4. SR is a finite set of sources;
5. $CD : D \to SR$ specifies the source of each data element; and
6. $CIR \subseteq D \times D$ indicates the conflict of interest relation.

Notice that, data elements are referred to as objects in PD-CW. In the following, we also refer to the elements in D as objects for convenience. In the formalization of PD-CW, a subject s has access to an object o after the firing of a transition t such that $T_S(t) = s \wedge o \in Read(t) \cup Write(t)$. Besides, the writing operations in PD-CWs are considered as appending rather than overwriting data elements. Thus, the writing operations do not affect the data validity.

For example, Figure 6 a) shows a PD-CW, where $S = \{sc_1, sc_2, sc_3, sa_1\}$ is the subject set. The object set D, source set CD and CIR are shown as Figure 6 b), where $(a, b) \in CIR$.

3.2 Configuration and Reachability Graph

Subjects may have access to different objects at the same marking due to different firing sequences, which brings difficulties for the verification of CW policy. For example, the PD-CW in Figure 6 a) will eventually reach the marking $[0, 0, 0, 0, 0, 0, 0, 1]$. However, we cannot infer which sequence is fired at this marking. As a result, it is difficult to trace the access records of subjects. In order to handle this problem, we introduce *configuration* to describe the dynamic behaviors of PD-CW, which includes the access records of subjects.

Definition 4 (Configuration)**.** A *configuration* of a PD-CW is a tuple $c = (M, PR)$, where:

1. M is a marking of PD-CW; and
2. $PR : S \to 2^D$ records the objects that subjects have access to at M.

Based on the concept of configurations, we can not only describe the running states of information systems, but also record the access rights of subjects. Notice that, the PD-CW is assumed free of data errors in this paper, especially missing-data errors and data inconsistency. To ensure this, some techniques for data error detection [17, 20] can be applied to PD-CW models before verifying CW policy.

The initial configuration c_0 of a PD-CW is (M_0, PR_0), where M_0 is the initial marking and PR_0 satisfies that $\forall s \in S : PR_0(s) = \varnothing$. The firing of a transition t at a configuration $c = (M, PR)$ generates a new configuration $c' = (M', PR')$ such that $M[t\rangle M'$ and $PR'(T_S(t)) = PR(T_S(t)) \cup \mathit{Read}(t) \cup \mathit{Write}(t)$. Furthermore, the firing of a transition sequence σ at a configuration $c = (M, PR)$ generates a new configuration $c'' = (M'', PR'')$ such that $M[\sigma\rangle M''$ and $\forall t \in \sigma : PR''(T_S(t)) = PR(T_S(t)) \cup \mathit{Read}(t) \cup \mathit{Write}(t)$. A configuration c is called a *reachable configuration* if it can be generated by the firing of a transition sequence at the initial configuration. Building on this ground, we give the definition of *reachability graph* of PD-CWs.

Definition 5 (Reachability Graph)**.** A *reachability graph* of a PD-CW is a tuple (C, E, δ, c_0), where:

1. C is a set of reachable configurations;
2. $E \subseteq T$ is a transition set;
3. $\delta : C \times E \to C$ denotes the transition relation; and
4. c_0 is the initial configuration.

We provide Algorithm 1 for generating the reachability graph of PD-CW, where the function $\mathit{Enabled}(M)$ is to get all the enabled transitions at the marking M, and $\mathit{Fire}(M, t)$ is to calculate the generated marking after the firing of t at the marking M.

Figure 6 c) shows the reachability graph of Figure 6 a). There exist nine configurations, where c_1 is a configuration at the marking $[0, 1, 1, 0, 0, 0, 0, 0]$, and sc_1 has access to a, denoted as $PR(sc_1) = \{a\}$.

Algorithm 1 The algorithm for Generating Reachability Graph

Require:
A PD-CW $(\Sigma, S, T_S, SR, CD, CIR)$;
Ensure:
The reachability graph (C, E, δ, c_0);
$c_0 := (M_0, PR_0)$;
$C := E := \varnothing$;
$SC := \{c_0\}$;
while $SC \neq \varnothing$ **do**
  Get a configuration $c = (M, PR)$ from SC;
  if $c \notin C$ **then**
    $C := C \cup \{c\}$;
    for each $t \in Enabled(M)$ **do**
      $E := E \cup \{t\}$;
      $M' := Fire(M, t)$;
      $PR' := PR$;
      $PR'(T_S(t)) := PR(T_S(t)) \cup Read(t) \cup Write(t)$;
      $c' := (M', PR')$;
      $\delta(c, t) := c'$;
      $SC := SC \cup \{c'\}$;
    end for
  end if
  $SC := SC \setminus \{c\}$;
end while
Output (C, E, δ, c_0);

3.3 Verification of CW Policy

The reachability graph of PD-CWs contains all required information for the verification of CW policy, so that we can analyze system states and access records to verify CW policy. An intuitive idea is that, we can check all the reachable configurations and identify those ones which violate the CW security conditions. For example, c_7 and c_8 in Figure 6 c) are two configurations which violate CW policy. However, illegal access can introduce the illegal access record that violates CW policy, impacting all subsequent configurations. Resolving illegal access ensures that all subsequent configurations remain compliant. Therefore, our target is to find the illegal access. Based on this idea, we propose Theorems 1 and 2 based on Definition 2 to check whether a PD-CW model complies with CW policy.

Theorem 1. Let $(\Sigma, S, T_S, SR, CD, CIR)$ and (C, E, δ, c_0) be a PD-CW and its reachability graph, respectively. The PD-CW violates the CW-Simple Security Condition if and only if there exists a reachable configuration $c = (M, PR)$ and a transition $t \in E$ satisfying:

1. t is enabled at M; and
2. $\exists o \in PR(T_S(t)) \cup Read(t) \cup Write(t), o' \in Read(t) \cup Write(t) : (o, o') \in CIR$.

Proof.

Sufficiency ($\Rightarrow$) Firing the transition t indicates that the subject $s = T_S(t)$ has access to objects o and o', which are in conflict of interest and lead to the violation of CW-Simple Security Condition.

Necessity ($\Leftarrow$): If CW-Simple Security Condition is violated, there must be a subject s having access to two objects o and o' in conflict of interest at certain configurations. Since we define $PR_0(s) = \varnothing$ at the initial configuration, there must exist a transition t and a configuration c such that $s = T_S(t)$ and s has access to o and o' after firing t, where $o \in PR(T_S(t)) \cup Read(t) \cup Write(t)$ and $o' \in Read(t) \cup Write(t)$.

□

Theorem 2. Let $(\Sigma, S, T_S, SR, CD, CIR)$ and (C, E, δ, c_0) be a PD-CW and its reachability graph, respectively. The PD-CW violates the CW-* Security Condition if and only if there exists a reachable configuration $c = (M, PR)$ and a transition $t \in E$ such that:

1. t is enabled at M; and
2. $\exists o \in PR(T_S(t)) \cup Read(t) \cup Write(t), o' \in Write(t) : CD(o) \neq CD(o')$.

Proof.

Sufficiency ($\Rightarrow$): Firing the transition t indicates that the subject $s = T_S(t)$ writes to the object o' when it has access to o, where $CD(o) \neq CD(o')$. Thus, it leads to the violation of CW-* Security Condition.

Necessity ($\Leftarrow$): If CW-* Security Condition is violated, there must be a subject s writing to an object o' while having access to another object o at certain configurations, where $CD(o) \neq CD(o')$. Since we define $PR_0(s) = \varnothing$ at the initial configuration, there must exist a transition t and a configuration c such that $s = T_S(t)$, s writes to o' and has access to o after firing t, where $o \in PR(T_S(t)) \cup Read(t) \cup Write(t)$ and $o' \in Write(t)$.

□

A *violation* is a pair (c, t), which indicates that CW policy is violated if the transition t is fired at the configuration c. In fact, it refers to the illegal access that violates CW policy. For example, there are two violations (c_5, t_5) and (c_6, t_6) in Figure 6, which violate the CW-Simple Security Condition and CW-* Security Condition, respectively.

Based on Theorems 1 and 2, we develop Algorithm 2 to verify the CW policy, and detect all the violations within PD-CWs. If its output SV is an empty set,

the PD-CW complies with CW policy. Otherwise, the PD-CW violates CW policy, and all the violations are recorded in SV. The time complexity of Algorithm 2 is $O(|C| \times |E| \times |D|)$.

Algorithm 2 The algorithm for Verifying CW Policy

Require:
A PD-CW $(\Sigma, S, T_S, SR, CD, CIR)$;
The reachability graph (C, E, δ, c_0);
Ensure:
The set of violations SV;

```
1: SV := ∅;
2: for each c = (M, PR) ∈ C do
3:   for each t ∈ Enabled(M) do
4:     for each o ∈ PR(T_S(t)) ∪ Read(t) ∪ Write(t) do
5:       if ∃o' ∈ Read(t) ∪ Write(t) : (o, o') ∈ CIR then
6:         SV := SV ∪ {(c, t)};
7:         Break;
8:       end if
9:       if ∃o' ∈ Write(t) : CD(o) ≠ CD(o') then
10:        SV := SV ∪ {(c, t)};
11:        Break;
12:      end if
13:    end for
14:  end for
15: end for
16: Output SV;
```

4 CASE STUDY

The process and PD-CW model of booking flight tickets [21] are shown in Figure 7 and Figure 8 a), respectively. A customer wants to book a ticket on a third-party platform. After the customer asks for booking, the airline A first quotes the ticket price. Subsequently, the customer can decide to accept the price or not. If he/she rejects the price, the airline B then quotes its price. Similarly, the customer can decide whether to accept it. But if he/she rejects the second quoted price, the booking order will be canceled. The security requirement is that, neither of the airlines has access to the quoted price from the other airline. Otherwise, it may lead to malicious competition. Due to the fact that both of the airlines only have access to their own quoted price, the security requirement is the same as CW policy. The objects, sources and CIR are shown as Figure 8 b), where pa and pb are respectively quoted prices of airlines A and B, and they are in conflict of interest.

In order to verify CW policy, we first generate the reachability graph of Figure 8 a) by Algorithm 1, as shown in Figure 8 c). The reachability graph contains

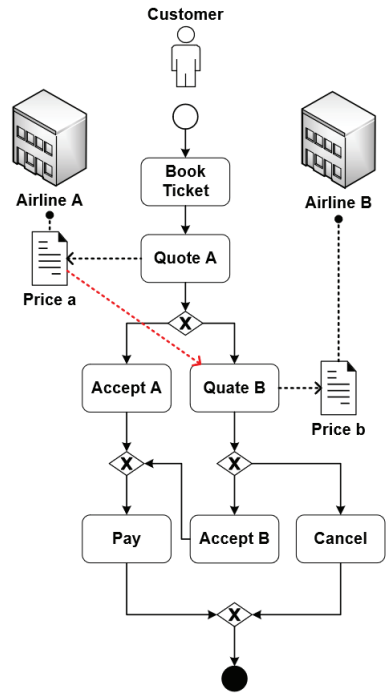
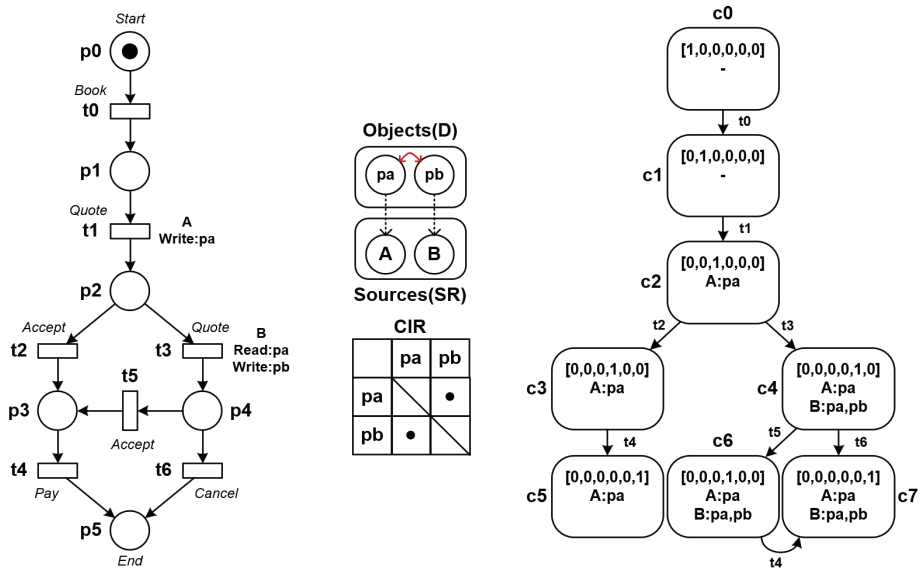


Figure 7. The process of booking flight tickets

all the running information of the business process of airlines. Afterwards, we can detect the violations of CW policy within the PD-CW by Algorithm 2. The result shows that (c_4, t_4) is a violation, i.e., firing the transition t_4 at the configuration c_4 leads to the violation with CW policy. Due to the fact that $(pa, pb) \in CIR$ and $CD(pa) \neq CD(pb)$, both the CW-Simple Security Condition and CW-* Security Condition are violated. In other word, CW policy is violated when the airline B quotes the price, which is caused by its illegal access to the price of the airline A.

Furthermore, we conduct a scale comparison between the CPN-based approach and PD-CW-based approach in three scenarios from previous work, including *FlightBook* [21], *FinancialAgent* [3] and *InsuranceClaim* [22]. The result is demonstrated as Table 1. In these three scenarios, the net structure scale of PD-CW is smaller than that of CPN, while the reachable graph scale is larger than that of CPN. That is because, PD-CWs utilize configurations to distinct different access records at the same marking, while CPNs incorporate extra places to trace the record. As a consequence, the space costs of CPNs and PD-CWs are similar. However, the PD-CW can depict and analyze the data operations in the scenarios, while the CPN cannot. In general, our approach holds certain advantages.



a) b) c)

Figure 8. a) A PD-CW of booking flight tickets; b) objects, sources and CIR of a); c) the reachability graph of a)

Scenarios	CPN					PD-CW				
	Net			RG		Net			RG	
	$\|P\|$	$\|T\|$	$\|F\|$	Nodes	Arcs	$\|P\|$	$\|T\|$	$\|F\|$	Nodes	Arcs
FlightBook	18	14	34	36	84	12	14	28	64	96
FinancialAgent	26	14	46	64	128	16	14	30	81	144
InsuranceClaim	18	10	28	25	50	10	10	20	36	60

Table 1. The Result of Scale Comparison in three scenarios

5 CONCLUSION

CW policy is a widely applied security policy, and it is related to the confidentiality and integrity of information security. Therefore, exploring its verification is of significant importance. As a widely used mathematical formalism, Petri nets are suitable for the verification of CW policy. Previously, some studies utilize CPNs for modeling and verification of CW policy due to their capability of depicting the data-flows. However, they do not consider data operations including read, write and delete, which can affect the verification of CW policy. In this paper, we propose a novel method for modeling and verifying CW policy based on PD-nets. We first give the definition of PD-CW to formalize the control-flows, data-flows and data operations of information systems. Subsequently, we introduce the configurations

and reachability graphs to depict the running states and access records. Ultimately, we provide two theorems to characterize the violation of CW policy in PD-CW, and develop an algorithm to detect the violations. A case study of booking flight tickets shows the effectiveness of our method. In addition, a comparison of scales shows that our method has distinct advantages. However, our method has two limitations. On the one hand, the PD-CW does not take into account the possible impact of data on the running of systems, such as guard functions. On the other hand, the state space of PD-CW increases due to the application of configurations.

In future work, we plan to carry out the following studies:

1. Develop tools for modeling and verifying CW policy. We plan to design and implement tools for supporting the visual modeling of PD-CW and automatic verification of CW policy;
2. Alleviate the state space explosion problem. As known to all, the methods based on reachability graph, which generally utilize the interleaving semantics, are prone to the state space explosion problem. The unfolding techniques [23] seem promising to alleviate the problem, and improve the efficiency of verifying CW policy; and
3. Generalize our method to the workflow systems [24]. Workflow systems involve the business logic and interaction of business processes. WFD-nets [25], which consider both the control-flows and data-flows, are suitable for modeling and verification of CW policy in workflow systems. The main problem is how to handle guard functions [26].

Acknowledgement

This work is supported by the Natural Science Foundation of China under Grant Nos. 62472388, 62002328, and the Fundamental Research Funds of Zhejiang Sci-Tech University Grant No. 24232123-Y.

REFERENCES

[1] Tu, H.—Xu, Y.—Shao, J.—Xiang, D.—Liu, G.: Checking Non-Interference Based on Unfolding Techniques of Petri Nets. 2024 International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2024, pp. 1–6, doi: 10.1109/ICNSC62968.2024.10760036.

[2] Brewer, D. F. C.—Nash, M. J.: The Chinese Wall Security Policy. Proceedings 1989 IEEE Symposium on Security and Privacy, 1989, pp. 206–214, doi: 10.1109/SECPRI.1989.36295.

[3] Zhang, Z.—Hong, F.—Liao, J.: Modeling Chinese Wall Policy Using Colored Petri Nets. The Sixth IEEE International Conference on Computer and Information Technology (CIT'06), 2006, pp. 162–162, doi: 10.1109/CIT.2006.123.

[4] Zhang, R.—Liu, G.—Kang, H.—Wang, Q.—Tian, Y.—Wang, C.: Improved Bell–LaPadula Model with Break the Glass Mechanism. IEEE Transactions on Reliability, Vol. 70, 2021, No. 3, pp. 1232–1241, doi: 10.1109/TR.2020.3046768.

[5] Kessler, V.: On the Chinese Wall Model. In: Deswarte, Y., Eizenberg, G., Quisquater, J. J. (Eds.): Computer Security – ESORICS 92. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 648, 1992, pp. 41–54, doi: 10.1007/BFb0013891.

[6] Tsai, T. H.—Chen, Y. C.—Huang, H. C.—Huang, P. M.—Chou, K. S.: A Practical Chinese Wall Security Model in Cloud Computing. 2011 13th Asia-Pacific Network Operations and Management Symposium, IEEE, 2011, pp. 1–4, doi: 10.1109/APNOMS.2011.6076992.

[7] Basu, S.—Sengupta, A.—Mazumdar, C.: Modelling Operations and Security of Cloud Systems Using Z-Notation and Chinese Wall Security Policy. Enterprise Information Systems, Vol. 10, 2016, No. 9, pp. 1024–1046, doi: 10.1080/17517575.2016.1183264.

[8] Alqahtani, S. M.—Gamble, R.—Ray, I.: Auditing Requirements for Implementing the Chinese Wall Model in the Service Cloud. 2013 IEEE Ninth World Congress on Services, 2013, pp. 298–305, doi: 10.1109/SERVICES.2013.44.

[9] Anupa, J.—Sekaran, K. C.: Securing Cloud Workflows Using Aggressive Chinese Wall Security Policy. 2014 First International Conference on Networks & Soft Computing (ICNSC2014), IEEE, 2014, pp. 85–91, doi: 10.1109/CNSC.2014.6906714.

[10] Fehis, S.—Nouali, O.—Kechadi, M. T.: A New Distributed Chinese Wall Security Policy Model. Journal of Digital Forensics, Security and Law, Vol. 11, 2016, No. 4, Art. No. 11, doi: 10.15394/jdfsl.2016.1434.

[11] Clarke, E. M.: Model Checking. In: Ramesh, S., Sivakumar, G. (Eds.): Foundations of Software Technology and Theoretical Computer Science (FSTTCS 1997). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1346, 1997, pp. 54–56, doi: 10.1007/BFb0058022.

[12] Qi, L.—Su, Y.—Zhou, M.—Abusorrah, A.: A State-Equation-Based Backward Approach to a Legal Firing Sequence Existence Problem in Petri Nets. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 53, 2023, No. 8, pp. 4968–4979, doi: 10.1109/TSMC.2023.3241101.

[13] Liu, G.—Zhou, M.—Jiang, C.: Petri Net Models and Collaborativeness for Parallel Processes with Resource Sharing and Message Passing. ACM Transactions on Embedded Computing Systems (TECS), Vol. 16, 2017, No. 4, Art. No. 113, doi: 10.1145/2810001.

[14] Liu, G.: Petri Nets: Theoretical Models and Analysis Methods for Concurrent Systems. Springer Nature Singapore, 2022, doi: 10.1007/978-981-19-6309-4.

[15] Huang, H.—Kirchner, H.: Formal Specification and Verification of Modular Security Policy Based on Colored Petri Nets. IEEE Transactions on Dependable and Secure Computing, Vol. 8, 2011, No. 6, pp. 852–865, doi: 10.1109/TDSC.2010.43.

[16] Tu, H.—Xiang, D.—Ding, Z.—Liu, G.: Detecting Information Leakage Against Chinese Wall Policy Based on the Unfolding Technique of Colored Petri Nets. IEEE Transactions on Computational Social Systems, 2024, doi:

10.1109/TCSS.2024.3461812.

[17] Xiang, D.—Liu, G.—Yan, C.—Jiang, C.: Detecting Data Inconsistency Based on the Unfolding Technique of Petri Net. IEEE Transactions on Industrial Informatics, Vol. 13, 2017, No. 6, pp. 2995–3005, doi: 10.1109/TII.2017.2698640.

[18] Moutinho, F.—Gomes, L.: Asynchronous-Channels Within Petri Net-Based GALS Distributed Embedded Systems Modeling. IEEE Transactions on Industrial Informatics, Vol. 10, 2014, No. 4, pp. 2024–2033, doi: 10.1109/TII.2014.2341933.

[19] Lin, T. Y.: Chinese Wall Security Model and Conflict Analysis. Proceedings 24th Annual International Computer Software and Applications Conference (COMPSAC2000), IEEE, 2000, pp. 122–127, doi: 10.1109/CMPSAC.2000.884701.

[20] Xiang, D.—Lin, S.—Wang, X.—Liu, G.: Checking Missing-Data Errors in Cyber-Physical Systems Based on the Merged Process of Petri Nets. IEEE Transactions on Industrial Informatics, Vol. 19, 2023, No. 3, pp. 3047–3056, doi: 10.1109/TII.2022.3181669.

[21] Atluri, V.—Chun, S. A.—Mazzoleni, P.: A Chinese Wall Security Model for Decentralized Workflow Systems. Proceedings of the 8th ACM Conference on Computer and Communications Security (CCS '01), 2001, pp. 48–57, doi: 10.1145/501983.501991.

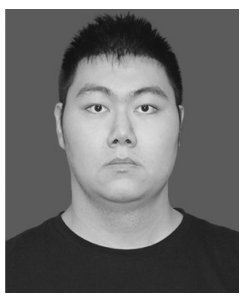
[22] Knorr, K.: Dynamic Access Control Through Petri Net Workflows. Proceedings 16th Annual Computer Security Applications Conference (ACSAC'00), IEEE, 2000, pp. 159–167, doi: 10.1109/ACSAC.2000.898869.

[23] Rouabah, Y.—Li, Z.: The Unfolding: Origins, Techniques, and Applications Within Discrete Event Systems. Mathematics, Vol. 11, 2023, No. 1, Art. No. 47, doi: 10.3390/math11010047.

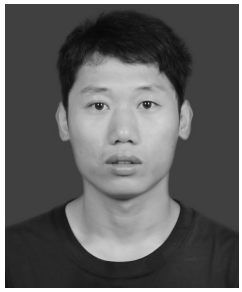
[24] Liu, G.—Reisig, W.—Jiang, C.—Zhou, M.: A Branching-Process-Based Method to Check Soundness of Workflow Systems. IEEE Access, Vol. 4, 2016, pp. 4104–4118, doi: 10.1109/ACCESS.2016.2597061.

[25] Zhao, F.—Xiang, D.—Liu, G.—Jiang, C.: Behavioral Consistency Measurement Between Extended WFD-Nets. Information Systems, Vol. 119, 2023, Art. No. 102274, doi: 10.1016/j.is.2023.102274.

[26] Xiang, D.—Liu, G.—Yan, C.—Jiang, C.: A Guard-Driven Analysis Approach of Workflow Net with Data. IEEE Transactions on Services Computing, Vol. 14, 2019, No. 6, pp. 1650–1661, doi: 10.1109/TSC.2019.2899086.

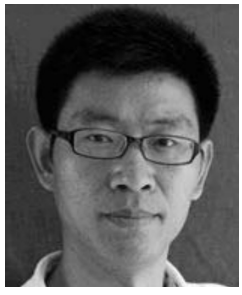


Hanqian Tu is currently working toward his Master's degree in the Department of Computer Science and Technology, Zhejiang Sci-Tech University. His research interests include model checking, Petri net, and system security.



Dongming Xiang received his Ph.D. degree in computer science and technology from the Tongji University, Shanghai, China, in 2018. He is currently Associate Professor with the Department of Computer Science and Technology, Zhejiang Sci-Tech University. He has authored over 30 papers including TII, TSC, TCSS, JAS, and ICPADS. His research interests include model checking, Petri net, formal methods, business process management, and service computing.

Liang Qi received his Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China in 2017. From 2015 to 2017, he was a Visiting Doctoral Student at the New Jersey Institute of Technology. He is currently Associate Professor in the Shandong University of Science and Technology. He has published 100+ papers in journals and conference proceedings, including the IEEE TITS, IEEE/CAA JAS, IEEE TSMCS, IEEE TCSS, IEEE TASE, IEEE TCYB, IEEE TNSE, IEEE TIP, IEEE IoT-J, IEEE TAI, IEEE RA-L, and IEEE SPL. He received the Best Student Paper Award-Finalist in the 15th IEEE International Conference on Networking, Sensing and Control (ICNSC'2018) and the Best Paper Award-Finalist in the 3rd IEEE International Conference on Automation in Manufacturing, Transportation and Logistics (iCaMaL2023). His current research interests include manufacturing systems, Petri nets, optimization, machine learning, and intelligent transportation. He is currently serving as an Associate Editor for IEEE Transactions on Intelligent Transportation Systems.



Guanjun Liu received his Ph.D. degree in computer software and theory from the Tongji University, Shanghai, China, in 2011. He was Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2011 to 2013. He was Post Doctoral Research Fellow with the Humboldt University Berlin, Germany, from 2013 to 2014, supported by the Alexander von Humboldt Foundation. He is currently Professor with the Department of Computer Science and Technology, Tongji University. He has authored over 90 papers including 23 papers in IEEE/ACM Transactions and two books. His research interests include Petri net theory, model checking, Web service, workflow, discrete event systems, and information security.