A Novel Noise Obfuscation Model and Its Strategies for Effective and Efficient Privacy Protection in Cloud Computing

by

Gaofeng Zhang

B.Eng. (Hefei University of Technology)

M.Eng. (Hefei University of Technology)

A thesis submitted to

Faculty of Information and Communication Technologies
Swinburne University of Technology

for the degree of
Doctor of Philosophy

April 2013

To my parents and my friends

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma, except where due reference is made in the text of the thesis. To the best of my knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Gaofeng Zhang

April 2013

Acknowledgements

I sincerely express my deepest gratitude to my principle coordinating supervisor, Professor Yun Yang, for his experienced supervision and continuous encouragement throughout my PhD study. And I want to show my most honest appreciation to my coordinating supervisor, Associate Professor Jinjun Chen, for his productive supervision and passionate encouragement during the past more than three years. Without their consistent supports, I would not have been able to complete this thesis.

I thank Swinburne University of Technology and the Faculty of Information and Communication Technologies for offering me a full Research Scholarship throughout my doctoral program. I also thank the Research Committee of the Faculty of Information and Communication Technologies for providing me with financial support to attend conferences.

My thanks also go to staff members, research students and research assistants at SUCCESS for their help, suggestions, friendship and encouragement, in particular, Professor Chengfei Liu, Alan Colman, Gillian Foster, Jianxin Li, Rui Zhou, Hai Huang, Jiajie Xu, Xiaoyuan Xie, Jing Gao, Minyi Li, Wei Dong. Of course, previous and current members of our group: Qiang He, Xiao Liu, Dong Yuan, Wenhao Li, Dahai Cao, Xuyun Zhang, Chang Liu, Feifei Chen, Jofry Sutanto and Antonio Giardina.

Last but not least, I am deeply grateful to my parents Xianshu Zhang and Yun Gao for raising me up, teaching me to be a good person, and supporting me to study abroad, and their understandings, encouragements, sacrifices and help.

Abstract

Cloud computing is a novel market-oriented computing paradigm which can manage various IT resources and provide virtual scalable IT services under its openness and virtualisation features. Hence, cloud customers can save huge capital investments in their own infrastructure by deploying or utilising these IT services through cloud. Due to this outsourcing, it is a natural concern for cloud customers about how to protect their privacy because they do not have much control inside cloud. Without related privacy protection, customers may lose the confidence in and desire to take cloud computing into practice eventually. Therefore, as one of the most important issues for both academia and industry in cloud computing, cloud privacy protection is a joint research frontier for both cloud computing and privacy protection. For instance, due to the openness and virtualisation features, various malicious service providers may exist in cloud environments out of cloud customers' control. Meanwhile, these customers are quite hard to distinguish these malicious ones for the same reason. As a result, some of these malicious service providers can collect these customers' service data, such as service requests or communication logs, and then deduce their privacy without authorisation or permission. Therefore, certain technical actions should be taken to protect their privacy automatically at client side. That is the cloud privacy protection at client side which is one essential aspect of the entire cloud privacy protection.

In this regard, as a promising cloud privacy protection approach at client side, noise obfuscation can protect customer privacy without service providers. For example, it injects noise service requests into real service requests. As a result, malicious service providers are hard to distinguish which are real ones so that related customer privacy can be protected in general. Actually, in this thesis, noise obfuscation has to investigate how to withstand various privacy risks and concerns

in opaque and complex cloud environments. That is to make the noise obfuscation approach effective and functional in cloud computing. Besides, the pay-as-you-go style of cloud computing makes the cost of noise obfuscation as a key privacy concern about the efficiency of privacy protection. Generally speaking, to apply the noise obfuscation approach for privacy protection in cloud computing, based on existing noise obfuscations, we need to improve noise obfuscation in both the effectiveness and efficiency of privacy protection to match the opaque and complex cloud environments.

By now, current noise obfuscations are preliminary and isolated for cloud privacy protection. In other words, they are inadequate to consider these various privacy risks and concerns in the opaque and complex cloud computing. In this regard, to promote noise obfuscation and improve cloud privacy protection, this thesis proposes a novel noise obfuscation model for cloud privacy protection and a series of novel strategies for effective and efficient noise obfuscation in cloud computing. By investigating the limitations of conventional noise obfuscation related research on these privacy risks and concerns, this novel model can provide a systematic and comprehensive support for privacy protection at client side to improve the cloud privacy protection. In this novel model, there are three major components: noise pre-processing component – to process customers' requirements on noise obfuscation; noise generation component – to generate noise data effectively and efficiently; and noise utilisation component – to utilise noise obfuscation into opaque and complex cloud environments. Based on these components, we propose a suite of innovative strategies to deal with several serious privacy risks and concerns systemically and comprehensively during noise obfuscation functions for cloud privacy protection. Simulation comparisons and quantitative evaluations are presented to demonstrate that our novel model and innovative strategies can significantly improve the noise obfuscation approach in both effectiveness and efficiency of cloud privacy protection.

Specifically, in the noise pre-processing component, a novel privacy-leakage-tolerance based noise enhancing strategy is proposed to bridge customers' privacy requirements and noise obfuscations. With this strategy, the customer-set privacy-leakage-tolerance can enhance noise obfuscation by controlling the noise set's

creation, and improve the efficiency of noise obfuscation. In the noise generation component, to deal with the probability fluctuation privacy risk, a novel time-series pattern based noise generation strategy is presented to conceal fluctuations of occurrence probabilities by time-series patterns' generation and forecasting. Similarly, to deal with the association analysis privacy risk, a novel association probability based noise generation strategy is proposed to conceal association probabilities under the association probability model for noise obfuscation. In the noise utilisation component, noise utilisation focuses on multiple cloud services scenarios based on the former two components which focus on single cloud service scenarios. Furthermore, in the ethical multiple services case, a novel correlation based noise injection strategy is designed to combine ethical cloud services together to improve the effectiveness of cloud privacy protection on noise obfuscation. Likewise, in the unethical multiple services case, a novel common set based noise cooperation strategy is created to withstand the privacy risk that unethical cloud services could share customer private information and break existing noise obfuscations. Briefly, to protect cloud customer privacy systematically and comprehensively, various novel strategies are invented for concealing customer privacy effectively and efficiently under various privacy risks and concerns in opaque and complex cloud environments.

The major contribution of this research is that we propose a novel noise obfuscation model for improving cloud privacy protection by systematically and comprehensively withstanding privacy risks and concerns in opaque and complex cloud environments. Specifically, a suite of novel strategies including noise pre-processing strategies, noise generation strategies and noise utilisation strategies, have been designed and developed. Corresponding comparisons and quantitative evaluations have shown that these innovative strategies can obtain great improvements on the effectiveness or efficiency of privacy protection on noise obfuscation. In summary, by deploying our innovative model and its novel strategies, noise obfuscation can better support cloud privacy protection effectively and efficiently.

The Author's Publications

Book:

1. Xiao Liu, Dong Yuan, **Gaofeng Zhang**, Wenhao Li, Dahai Cao, Qiang He, Jinjun Chen and Yun Yang, *The Design of Cloud Workflow Systems*. Springer, ISBN: 978-1-4614-1932-7, 2012.

Book Chapter:

2. Xiao Liu, Dong Yuan, **Gaofeng Zhang**, Jinjun Chen and Yun Yang, SwinDeW-C: A Peer-to-Peer Based Cloud Workflow System. *Handbook of Cloud Computing*, pages 309-332, Springer, ISBN: 978-1-4419-6523-3, 2010.

Journal Articles:

3. **Gaofeng Zhang**, Yun Yang and Jinjun Chen, *A Historical Probability based Noise Generation Strategy for Privacy Protection in Cloud Computing*, Journal Computer and System Sciences (JCSS), Elsevier, vol.78, issue 5, pp. 1374- 1381, September 2012, ISSN: 0022-0000.

4. **Gaofeng Zhang**, Yun Yang, Dong Yuan and Jinjun Chen, *A Trust-based Noise Injection Strategy for Privacy Protection in Cloud*, Software: Practice and Experience (SPE), Wiley, vol. 42, pp. 431-445, April 2012. ISSN: 1097-024X.

5. Dong Yuan, Yun Yang, Xiao Liu, **Gaofeng Zhang** and Jinjun Chen, *A Data Dependency Based Strategy for Intermediate Data Storage in Scientific Cloud Workflow Systems,* Concurrency and Computation: Practice and Experience (CCPE), Wiley, vol. 24, issue 9, pp. 956-976, August 2010. ISSN: 1532-0634.

Conference Papers:

6. **Gaofeng Zhang**, Xuyun Zhang, Yun Yang, Chang Liu and Jinjun Chen, *An Association Probability based Noise Generation Strategy for Privacy Protection in Cloud Computing*, presented at the 10th International Conference on Service Oriented Computing (ICSOC 2012), pp. 639-647, November 12-16, 2012, Shanghai, China.

7. **Gaofeng Zhang**, Yun Yang, Xuyun Zhang, Chang Liu and Jinjun Chen, *Key Research Issues for Privacy Protection and Preservation in Cloud*, presented at the 2012 International Conference on Cloud and Green Computing (CGC2012), pp. 47-54, November 1-3, 2012, Xiangtan, China.

8. **Gaofeng Zhang**, Yun Yang, Xiao Liu and Jinjun Chen, *A Time-Series Pattern based Noise Generation Strategy for Privacy Protection in Cloud Computing*, presented at the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 458-465, May 13-16, 2012. Ottawa, Canada.

9. Xiao Liu, Yun Yang, Dong Yuan, **Gaofeng Zhang**, Wenhao Li and Dahai Cao, *A Generic QoS Framework for Cloud Workflow Systems*, 2011 International Conference on Cloud and Green Computing (CGC2011), pp. 713-720, December 12-14, 2011, Sydney, Australia.

Table of Contents

List of Figures

List of Algorithms

Chapter 1
Introduction

This thesis focuses on the privacy issue in the cloud. This is an extremely important issue for customers to deploy applications and utilise IT services in cloud computing. The goal of the novel research reported in this thesis is that cloud customers can improve the privacy protection performance and establish confidence for cloud computing in business markets. Generally speaking, as a promising privacy protection approach at client side, noise obfuscation for privacy protection in cloud computing consisting of noise pre-processing strategies, noise generation strategies, noise utilisation strategies and the noise obfuscation model is designed and developed. Experimental evaluation demonstrates that our work can help to enhance privacy protection effectively in cloud computing, meanwhile related noise cost can be reduced significantly or controlled reasonably in the pay-as-you-go cloud environments in terms of efficiency.

This chapter introduces the background and key issues of this research. It is organised as follows. Section 1.1 gives a brief introduction to privacy protection in cloud computing. Section 1.2 discusses some privacy challenges in cloud computing, and outlines the research in this thesis—noise obfuscation. Finally, Section 1.3 presents an overview for the remainder of this thesis.

1.1 Introduction to Privacy Protection in Cloud Computing

Cloud computing is a novel concept for IT services [70]. In brief, it is described as a promising framework for delivering IT services based on distributed system and

service computing [63]. IT resources, like computing, storage and communication, can be collected, packed and distributed in business markets to be provided to cloud customers with a pay-as-you-go fashion in IT infrastructure [2, 70]. It means that these customers can save huge capital investments and maintenance consumptions on their own hardware and software [63]. In this regard, both academia and industry have paid significant attentions to cloud computing which is viewed as a promising platform to control the current increasing cost on IT infrastructure and related concerns about energy consumption.

From the nature and basic concepts of cloud computing, it is clear that the important openness and virtualisation features of cloud environments require more attentions on privacy protection to provide a safe and secure business model and environment to all related roles in the cloud [79]. For example, cloud customers could upload and deploy their data and applications in the cloud, or access and utilise cloud services with related service agreements. In other words, all customers' data and service processes are in these virtualised and open cloud environments physically. Without related privacy protection, the 'unsafe' cloud environments can destroy the confidence for cloud computing from users, enterprises and governments [22, 62]. As a result, in this case, the cloud's main advantage—cost-saving would be quite hard to persuade cloud potential customers and supporters to take it into practice [97]. That is why privacy protection in cloud computing is chosen as the main topic of this thesis to support cloud's development in both views of academia and industry. In this thesis, as a promising computing architecture, cloud computing requires privacy protection to investigate and deal with all actual and potential privacy risks and concerns which are brought by cloud computing, or existing ones depraved by cloud computing [75].

Based on the significance of privacy protection in cloud computing [53], we can discuss this topic in some dimensions to provide a general picture about cloud privacy protection. The first one is the time dimension: On one hand, privacy protection is a 'old' problem which exists quite a long time [8]. And there are some common mathematical basis to be utilised consistently [71, 1]. Hence, privacy protection in cloud computing cannot be investigated without existing privacy protection issues and work [62]. On the other hand, cloud privacy protection is a quite challenging problem in current academia and industry. Cloud computing

promises itself as a novel service paradigm to reorganise existing IT infrastructure elements [2]. In this open and virtualised environment, privacy protection requires necessary modifications in views of customers, engineers and managers, comprehensively. Hence, novel strategies or approaches have to be invented to enhance privacy protection in cloud computing and support the current fast development of cloud computing [62, 82]. Therefore, due to the 'old' and 'novel' features, privacy protection in cloud computing means broad and variable privacy challenges for both researchers and engineers from the perspective of time dimension.

The other dimension is the space dimension, or the location view. Based on Internet and high-speed communications, cloud computing provides powerful, green and smart IT services to remote customers or terminals [47]. Hence, privacy protection in cloud computing has to consider the disparity of real locations of services or customers. For example, one cloud storage service provider in one country and one cloud computing service provider in another country may cooperate to fulfil a cloud service process to a cloud customer in the third country, and privacy protection in cloud computing should analyse related risks and deploy corresponding approaches to pursue reasonable privacy protection and obey different regulations and policies in these countries. In this regard, United States [19] and European Union [37] have different views in privacy laws. Besides, not only the real locations, the 'virtualised' locations in virtualised cloud environments also require privacy protection approaches to keep data and processes safe in terms of complex cloud service processes. For example, these different cloud service providers may have different privacy policies to protect their customers' privacy, or even some of them do not consider this. In other words, cloud privacy protection has to coordinate to pursue a systemic and comprehensive protection. Hence, privacy protection in cloud computing requires effective and efficient approaches to be analysed and utilised in every location, regardless real or virtualised. It is another aspect of the broad and variable privacy challenges in cloud computing, from the perspective of space dimension.

Generally speaking, privacy protection in cloud computing is a complicated topic for customers, engineers and managers in cloud environments, with various privacy approaches to withstand broad privacy challenges. In this thesis, we focus

on one specific privacy protection approach—noise obfuscation, and utilise it for privacy protection in cloud computing to support cloud's current and future blossoming promoting. In the next section, we will introduce the noise obfuscation approach based on the privacy challenge analysis in cloud computing.

1.2 Privacy Challenges in Cloud Computing

Due to the openness and virtualisation features, privacy protection in cloud computing has to consider and withstand various privacy risks and concerns to keep privacy secure and safe. In this section, we discuss some serious privacy challenges in cloud computing, and introduce the noise obfuscation for privacy protection as the key topic of this thesis.

1.2.1 Some Key Privacy Challenges in Cloud Computing

As discussed before, cloud privacy protection is a crucial issue to promote cloud computing into practice. Hence, some key privacy challenges should be emphasised and highlighted:

1) Privacy distribution under data isolation in cloud platforms [85]: it is obvious that data isolation in cloud computing is quite important for privacy protection. For example, one malicious attacker can deploy his/her applications in a public cloud platform which is utilised by many other customers in the meantime, and he/she may use some malicious codes or strategies to access other customers' applications or data without authorisation, because current cloud platform managements may distribute more than one customer to use one same physical machine or storage disk. And data isolation focuses on dealing with this. For cloud service providers, it is a powerful tool to manage their platforms and provide healthy, safe and attractive cloud services for their customers with cloud privacy protection. Besides, data isolation is quite complex in terms of different cloud data, applications and service levels in cloud platforms systematically. Therefore, it is a serious and important issue in cloud privacy protection.

2) Privacy distribution under specific polices: as we discussed before, privacy protection has to consider related laws and regulations [28]. But in different

countries, these policies on privacy could be quite different. Hence, to follow these different policies, privacy distribution is a serious topic to be considered in cloud computing. For instance, building multiple data centres in different countries is a reasonable solution for a multinational cloud enterprise to improve its service performance in different countries. Private information and data in these data centres have to consider the synchronous strategies and policies to match different laws and regulations in terms of data secureness, backup, authorisation, access control, data duplication and destruction, and so on. Without this, cloud enterprises may fall into law troubles and lose opportunity to be promoted in the blossoming stage of cloud computing. Hence, privacy distribution in cloud computing is a much more serious and complex challenge than ever before, from the perspective of privacy protection.

3) Privacy protection at client side [73, 42]: it is clear that cloud computing is a virtualised service environment. Hence, for cloud customers, it is quite hard to distinguish 'malicious' service providers in cloud environments. Besides, to obtain a powerful, green and smart cloud service performance, cloud service providers could cooperate to fulfil a cloud service. This makes cloud customers even harder to monitor and protect their private information during these complicated cloud service processes. Hence, it is a serious threat to customer privacy in the view of cloud computing. They may lose their confidence in cloud computing under some extreme conditions. In this regard, these cloud customers need to be equipped with some technical actions on their own to protect their privacy without the support from service providers. That is privacy protection at client side. Under this topic, this thesis focuses on the noise obfuscation for privacy protection in cloud computing, which will be introduced next.

1.2.2 Noise Obfuscation for Cloud Privacy Protection

Based on the former discussions, under the key privacy challenge—privacy protection at client side, we introduce the noise obfuscation approach for privacy protection in cloud computing in this subsection. Let us start with a motivating example:

One customer, who often travels to one city in Australia, say 'Sydney', and checks the weather report regularly from a weather service in cloud environments

before departure. The frequent appearance of service requests about the weather report for 'Sydney' can reveal the privacy that the customer usually goes to 'Sydney'. But if a system aids the customer to inject other requests like 'Perth' or 'Darwin' into the 'Sydney' queue, the service provider cannot distinguish which ones are real and which ones are 'noise' as it just sees a similar style of service request. These requests prevent from revealing the location privacy of the customer. In such cases, the privacy can be protected by noise obfuscation in general. That is a basic idea of noise obfuscation for cloud privacy protection. And we will use this example frequently to illustrate noise obfuscation details in the following chapters. We discuss noise obfuscation in common cloud environments in this thesis as the motivating example, and it is obvious that this approach can be utilised in some other application areas, such as military systems. In those cases, some specific requirements can take noise obfuscation into different considerations and criteria, which could be viewed as a specialised process of noise obfuscation for cloud privacy protection.

In brief, a large number of unknown and malicious service providers may exist in open and virtualised cloud environments. Such service providers may collect service information from customers to analyse and deduce customers' privacy without their permission. For service providers, it is a common phenomenon to collect their customers' information, like service requests. From large to small firms, they often use them to analyse customers' behaviour, habits, and other private information [81]. Most ethical ones have adequate self-control to use the information by following certain policies and regulations, but some others may abuse this in unethical ways, especially in open and virtualised environments like cloud computing. Because the openness and virtualisation features make customers hard to distinguish and verify service providers and service processes, it is a serious privacy risk for cloud customers.

Existing major privacy protection mechanisms and approaches have not considered this situation thoroughly, hence cannot aid customers to withstand such type of privacy risks in cloud computing. Therefore, customers should be protected by taking certain technical actions for their privacy automatically at client side without participation of service providers. Noise obfuscation is an effective and promising approach in this regard. The key advantage is that this approach does not

need cooperation or assistance from service providers. Besides, compared to other privacy protection approaches at client side, noise obfuscation offers a better practicality in terms of the effectiveness and efficiency which will be discussed in Chapter 2. Hence, we investigate noise obfuscation in this thesis to improve the privacy protection in cloud computing.

Besides, in this thesis about the noise obfuscation for privacy protection in cloud computing, we focus on common customers' privacy without specific data structures or types. For instance, these service requests from customers to service providers may have some private information to fulfil service processes. And this information could be individual data items, like the location information in the motivating example. For other private data with complex data structures, noise obfuscation can be modified based on this thesis, according to data structure and knowledge representation on customer privacy.

In brief, in this thesis, we focus on how to use noise obfuscation effectively and efficiently to protect privacy in cloud computing. Hence, we firstly present a novel noise obfuscation model for privacy protection in cloud computing to execute the whole procedure of the noise obfuscation function for cloud privacy protection systematically and comprehensively, which gives a general description of the following steps. Then, in the noise pre-processing step of this model, a novel noise pre-processing strategy uses privacy-leakage-tolerance to link customers' privacy requirements and noise obfuscation functions together to guide noise obfuscation. After that, in the step of noise generation of this model, we propose two novel noise generation strategies to deal with two serious privacy risks—probability fluctuation and association analysis, respectively. Besides, these noise pre-processing and generation strategies are adequate to be executed sequentially as the single noise obfuscation process in the scenario of single service, without further noise utilisation strategies. Lastly, in the noise utilisation step which is a necessary step of this model to match the scenario of multiple services, we present two novel noise utilisation strategies to utilise these single noise obfuscation processes with noise pre-processing and noise generation in the cases of ethical multiple services and unethical multiple services. That is the main process of our novel noise obfuscation for privacy protection in cloud computing. In short, this thesis improves the noise obfuscation approach significantly and utilises it in cloud privacy protection by our

novel noise obfuscation model with a suite of novel strategies. We will introduce these steps in the following chapters one by one in detail.

1.3 Overview of This Thesis

In this thesis, the main topic is to systematically and comprehensively discuss noise obfuscation for privacy protection in cloud computing. Due to the nature of privacy protection, different risks and concerns in cloud privacy protection have to be discussed, and different strategies need to be designed and presented accordingly. And the effectiveness and efficiency of privacy protection on noise obfuscation can be improved in cloud computing. The thesis structure is depicted in Figure 1-1.

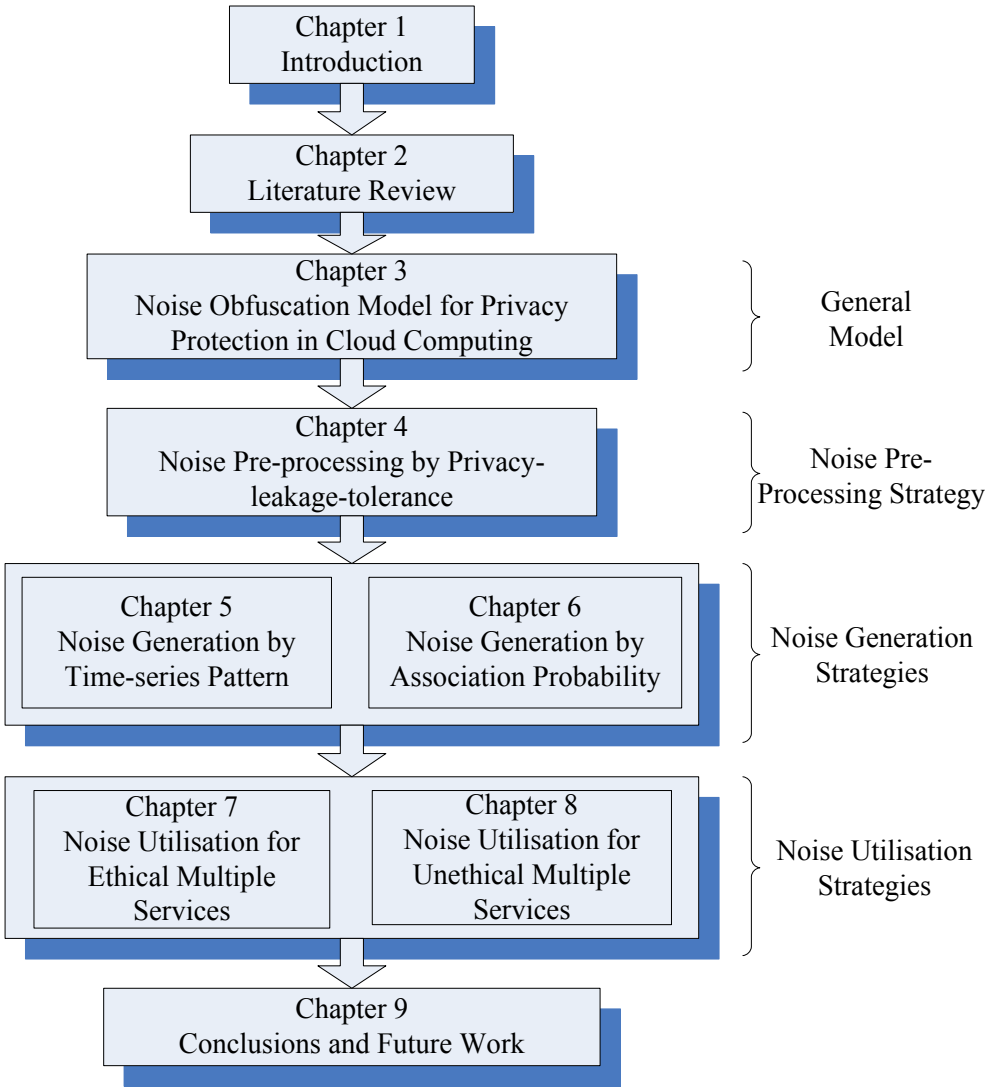


Figure 1-1 Thesis Structure

In Chapter 2, we introduce the related work to this research. We start from introducing privacy protection at service side, especially existing well-researched privacy approaches which can be utilised in cloud computing. Then, we introduce some representative work about privacy protection at client side to discuss the noise obfuscation approach for privacy protection in cloud computing, and point out the significance and practicality of noise obfuscation. At last, we introduce some other work about time-series analysis, association analysis, trust model, and so on which are important foundations for our work in different views.

In Chapter 3, we firstly demonstrate our novel noise obfuscation model for privacy protection in cloud computing. In this chapter, we focus on this noise obfuscation model to abstract and organise noise pre-processing strategies, noise generation strategies and noise utilisation strategies which will be introduced in detail in the following chapters, and provide a common procedure of noise obfuscation in cloud computing including: the noise pre-processing component, the noise generation component and the noise utilisation component. In one word, this model is the general framework of the noise obfuscation approach for privacy protection in cloud computing in this thesis. Before we present this model in this chapter, we discuss the relationship between noise obfuscation and other privacy protection approaches in cloud computing, and point out noise obfuscation approach's position and significance for privacy protection in cloud computing. By the way, in this model, the noise pre-processing component and noise generation component focus on single service in cloud privacy protection, and the noise utilisation component focuses on multiple services in cloud privacy protection based on the previous two components.

In Chapter 4, we develop noise pre-processing for single service by privacy-leakage-tolerance as the first step of our novel noise obfuscation model—the noise pre-processing component. In this chapter, we use a customer-set boundary to require and manage the noise obfuscation function in terms of cloud privacy protection. Under this privacy concern, the customer-set boundary—the privacy-leakage-tolerance can link customers' privacy requirements and noise obfuscation functions to pursue a better privacy protection performance. Hence, we present a novel privacy-leakage-tolerance based noise enhancing strategy as the noise pre-processing strategy in the noise pre-processing component.

In Chapter 5, we start to introduce the noise generation for single service as the second step of our novel noise obfuscation model—the noise generation component. In this chapter, we propose a novel time-series pattern based noise generation strategy. To withstand the privacy risk about fluctuations of occurrence probabilities, this strategy utilises time-series patterns to abstract past fluctuations of occurrence probabilities in service data and forecasts future fluctuations of occurrence probabilities. After that, the noise data generated by this strategy can conceal these future fluctuations effectively and obtain a better effectiveness of privacy protection on noise obfuscation in terms of probabilities' fluctuations.

In Chapter 6, we further discuss the noise generation for single service in terms of another privacy risk: Some privacy attackers may focus on association relations and probabilities among service data as the privacy they wanted, which have not been considered by existing noise obfuscations. In this chapter, we propose a novel association probability based noise generation strategy to operate the noise generation process in terms of association probabilities. This strategy investigates this association analysis risk, and generates noise data by concealing association probabilities. Under this chapter (Chapter 6) and the pervious chapter (Chapter 5), the noise generation component in the noise obfuscation model can be operated.

In Chapter 7, we start with investigating noise utilisation for ethical multiple services as a part of the last step of our novel noise obfuscation model—the noise utilisation component. In this chapter, we focus on noise utilisation to deal with the privacy concern about ethical multiple services and present a novel correlation based noise injection strategy. In this strategy, the correlation model and the noise injection architecture are utilised to cooperate single noise obfuscation processes together to improve the effectiveness of privacy protection in terms of noise utilisation. The single noise obfuscation process is operated by the previous noise pre-processing component and noise generation component, which will be introduced in detail in Chapter 3.

In Chapter 8, we further investigate noise utilisation for unethical multiple services in terms of the privacy risk: it is possible that these unethical multiple services share customers' private data and break noise obfuscation. To deal with this privacy risk, we present a novel common set based noise cooperation strategy. In this strategy, the common set creation model is presented to provide the solution to

utilise single noise obfuscation processes. Generally speaking, the effectiveness of privacy protection on noise obfuscation can be improved by this strategy in the case of unethical multiple services in terms of noise utilisation. Under this chapter (Chapter 8) and the previous chapter (Chapter 7), the noise utilisation component in the novel noise obfuscation model can be operated.

Finally, in Chapter 9, we summarise our new model and strategies presented in this thesis, major contributions of this research, and consequent further research work.

Chapter 2
Literature Review

This chapter reviews the existing privacy protection work related to noise obfuscation for cloud privacy protection in this thesis. This chapter is organised as follows: Section 2.1 gives a general introduction of current privacy protection at service side. Section 2.2 reviews the privacy protection at client side including the noise obfuscation approach. Section 2.3 reviews other supporting work related to our novel noise obfuscation model, such as time-series analysis, association analysis and so on.

2.1 Privacy Protection at Service Side

In this section, we introduce the current work about privacy protection at service side. Generally speaking, privacy protection at service side is the dominating and mature research parts in cloud privacy protection. Under this topic, on one hand, novel and interesting ideas in cloud privacy protection have been presented continuously with the speedy development of cloud computing; on the other hand, existing mature privacy protection approaches are under revision and modification to match new cloud environments. Therefore, in this section, we will introduce some representative privacy protection approaches and issues at service side to support a whole picture of cloud privacy protection.

In brief, many and more researchers are starting to produce and/or have produced remarkable research on privacy protection related to cloud environments.

Some of them focus on a whole consideration on privacy protection: such as Huang *et al.* [94] discuss privacy protection in the value-added context-aware cloud. Similarly, Simoens *et al.* [57] present a biometric encryption system in privacy protection of biometric search area. And Neisse *et al.* [74] investigate trust and promote data security in cloud environments. Some of them focus on novel and interesting ideas, such as Itani *et al.* [89] discuss the "privacy as a service" idea to push cloud privacy protection into practice.

Privacy-Preserving Data Mining *(PPDM)* reveals a kind of privacy leakage in the minutiae [71]. To protect customers' privacy, Evfimievski *et al.* [4] use a randomisation operator to investigate and discuss the process of association rule mining. Besides, differential privacy [58] is to cope with this searching privacy preserving situation, particularly due to large scale and uneven distributions in structural data.

Similarly, Privacy-Preserving Data Publish (*PPDP*) has a wide utilised field in data publish of service web [12, 13]. In general, a SuLQ framework [10] considers privacy-aware statistical databases by improving the bounds on noise required for privacy. In the case with considering a trade-off between privacy and utility [86], *PPDP* has been enhanced to match the pay-as-you-go style of cloud computing.

Different from *PPDM* and *PPDP*, Privacy-preserving Information Retrieval (*PIR)* utilises another approach to protect privacy, which mainly prevents database operators from knowing users' interested records. Chor *et al.* [14] have a conclusion that, to get a perfect protection, a user has to query all the entries in database when dealing with a single server framework. Besides, Beimel *et al.* [5] and Goldberg *et al.* [46] apply information theories to take *PIR* in practice.

Proxy and anonymity network to protect customers' privacy have been widely discussed. The major goal is to keep anonymity or "invisibility" in a complex or "dangerous" network condition. For example, onion routing [24] and its successor *TOR* [29] provide a kind of sophisticated privacy protection scenarios, making it difficult for attackers to trace the customer via network traffic analysis. In social networks [9] and encrypted communications [78], anonymous network can protect privacy by identity anonymity. Besides, a hierarchical identity-based cryptography [59] can achieve mutual authentication in hybrid clouds, which can be viewed as an important basis to deploy anonymous network in cloud environments.

MapReduce [51] is a popular programming platform in cloud environments. Privacy protection in MapReduce has been considered to deal with some privacy risks: Word search could be enhanced by privacy-preserving in cloud computing [33], and access control [48] is another important topic for privacy protection in MapReduce. The hybrid approach [56] can promote cloud data-intensive instances to be more practical in terms of the combination of private cloud and public cloud. Besides, other cloud application platforms, like Hadoop, can be enhanced by privacy protection in terms of fixing system flaws, too [44, 45, 66].

Different to cloud programming platforms, in cloud management platforms, privacy protection focuses on some mature cloud environments [36, 66]. For example, in Amazon Elastic Compute Cloud (EC2) [31], Bugiel *et al.* [83] present one type of image attack which focuses on extracting sensitive information caused by unaware users, which needs more attentions in terms of privacy protection.

In the area of Virtual Machine (VM) which is a key supporting component of cloud computing, privacy protection is necessary to be considered. At a high level, an in-VM measuring framework [68] for increasing VM's security & privacy in cloud computing has been discussed to collect different VMs sources together based on trust mechanisms. Besides, identity management [85] and cost on privacy protection [17] can be viewed as important issues to influence the VM's privacy protection. At a low level, one approach [3] has been investigated to obtain a strong isolated computing to keep information secure based on specific hardware. Similarly, one kind of hypervisor attack surface enabling guest VMs can threat privacy in cloud computing, and be addressed by a strict user model [49].

As analysed in Section 1.2, various malicious service providers may exist in cloud environments. Some of them may record customers' service requests and collectively deduce customers' private information. Therefore, customers' privacy needs to be protected without service providers. This is the scenario that we focused on in this thesis.

Briefly, *PPDM* is not an ideal choice to address the scenario because it is out of customers' control, hence not suitable for protecting customers' privacy focused by this thesis. *PIR* and *PPDP* mainly work at service provider side, hence have the similar problem. Proxy and anonymity network need service provider's cooperation to enable such access, and have to face a possibility that cannot enable this access in

complex cloud environments. Other privacy protections in cloud computing, about MapReduce, cloud management platforms and VMs, focus on cloud management and cannot match this scenario well, like other privacy protection approaches at service side. Hence, in this thesis, we need to investigate privacy protection at client side in cloud computing which will be introduced in the next section.

2.2 Privacy Protection at Client Side

In the preceding section, privacy protection approaches at service side have been introduced in terms of cloud computing and pointed out their unsuitability for the privacy protection scenario in this thesis. Hence, in this section, we introduce representative privacy protection approaches at client side to match the scenario.

As introduced before, at cloud client side, privacy protection considers how to use or deploy cloud services safely depending on clients own. Hence, based on this semi-honest condition, some instructive approaches and ideas have been discussed, including some qualitative methods [27]. This kind of approaches is a promising research area in cloud privacy protection to give cloud customers confidence in terms of cloud client side.

Secure computation starts to combine a bunch of nodes mistrusted each other to complete a task together by cryptography [7]. Based on the theoretical analysis [8], the optimisation on the efficiency has been discussed [38], especially in some specific situations [60, 67]. From the perspective of cloud privacy protection, it is a promising approach to build a privacy protection system at client side in cloud computing. But by now, a significant efficiency improvement is necessary for secure computation to be practical in cloud computing [38, 95].

Similar to secure computation, homomorphic encryption extends the usage of encryption and decryption in the view of outsourcing in cloud computing [20]. And some papers try to take it into practice [21], such as in the view of efficiency [64]. Besides, [69] uses bilinear aggregate signature and public key based homomorphic authenticator to improve practicality. Generally speaking, with the improvement on efficiency and versatility, these cryptography approaches can protect privacy effectively at client side in cloud environments [20, 72].

In brief, cryptograph approaches for cloud privacy protection, including secure computation and homomorphic encryption, have to face a common efficiency problem to be utilised practically in cloud computing, although they have a strong mathematical basis. Specifically, these cryptograph approaches require adequate computing capability at client side to execute related computing processes, and this 'fat' client in cloud computing is a violation of the basic idea of cloud computing, where cloud customers should transfer major tasks in cloud environments and keep clients 'thin'. Without this, the cost-saving of cloud would be impaired for these cryptograph approaches. Besides, these cryptograph approaches need supports from service providers to execute these related encryption and decryption processes, and this does not match the privacy protection scenario in this thesis very well.

By keeping a light-weight style of client side, noise obfuscation is another widely adopted approach for protecting private information at client side. It is clear that noise obfuscation can be utilised by cloud customers to keep their privacy safe on their own without support from cloud service providers. For example, Ardagna *et al.* [18] discuss the location privacy protection in a mobile environment, and present a solution based on different obfuscation operators. By a similar mechanism [54], private information can be analysed deeply for privacy protection. Besides, Perron *et al.* [35] investigate noise utilisation in wireless conditions as a type of data security in the communication area. Especially, noise insertion builds on the ground of information theory to conceal the characters of information [50]. Ye *et al.* [77] investigate noise injection in privacy-aware searching by formulating noise injection problem as a mutual information minimisation problem. And a common model is presented in terms of obfuscation-based private web search [34]. Zhang *et al.* [42] present a historical probability based noise generation strategy to improve the efficiency of privacy protection and obtain a promising cost-saving for privacy protection in cloud environments. And a trust based noise injection strategy is presented to discuss influences of complex relations in cloud computing on noise obfuscation schemes [41]. In short, compared to cryptograph approaches, noise obfuscation presents another promising approach to protect privacy at client side: obfuscating private information by noise information, instead of covering it directly.

In summary, privacy protection at client side has attracted more attentions in cloud environment than ever before. As an important approach in this area, noise

obfuscation is envisaged to perform well to enhance cloud privacy protection at client side, and support a comprehensive privacy protection in cloud computing. This is the major focus of this thesis. As discussed before, noise obfuscation has to face various privacy risks and concerns in cloud computing. Hence, we will present a general noise obfuscation model in Chapter 3 and corresponding related strategies to address them in Chapters 4, 5, 6, 7 and 8.

2.3 Other Supporting Work for Noise Obfuscation

In this section, we introduce some other supporting work for noise obfuscation, including operating model in distributed systems, trust and privacy risk evaluation, time-series analysis, association analysis and intersection attack. For our novel noise obfuscation model, these areas are important references to support some parts of it. Hence, they are necessary to be discussed in the literature review.

Operating model in distributed systems is a useful tool to describe different functions in distributed systems. For instance, Yang *et al.* [98] utilise grid and peer-to-peer technologies to model workflow systems by Swinburne Decentralised Workflow for Grid (*SwinDeW-G*) [84]. Similarly, Liu *et al.* [93] present a peer-to-peer based cloud workflow system to operate instance intensive cloud workflow applications. Besides, comparable ideas can be utilised in different areas to organise complex processes in distributed environments [87, 61]. In this thesis, we consider an general operating model for noise obfuscation in cloud computing—the novel noise obfuscation model for privacy protection in cloud computing, and use this model to organise all noise obfuscation processes, components and strategies in this thesis to be an entirety for privacy protection in cloud computing. We will discuss it in Chapter 3 in detail.

About trust and privacy risk evaluation in cloud, Neisse *et al.* [74] start to investigate trust in cloud environments to promote data security. Besides, the interoperability in cloud computing could be enhanced by trust based on heterogeneous domains and trust recommendation [90]. Accountability [76] is also an important aspect considered by trust and privacy risk evaluation in cloud computing. In brief, trust and privacy risk evaluation in cloud computing can make

cloud services and customers perform better in these opaque environments in different views. We will consider it as a reference in Chapters 4 and 7 in detail. In Chapter 4, to aid cloud customers to guide noise obfuscation, we utilise privacy risk evaluation to support the noise pre-processing by privacy-leakage-tolerance. And in Chapter 7, to incorporate single noise obfuscation processes among ethical cloud services, trust can be viewed as a valuable reference to present the correlation model to connect these services with single noise obfuscation processes in our novel correlation based noise injection strategy.

About time-series analysis, an online algorithm [30] has been used for segmenting time series in mining time-series databases. In another area, Shi *et al.* [32] investigate the aggregation of time-series data and present a group of *PSA* algorithms to protect each source's privacy, when the data aggregator is untrusted. In scientific workflow activities, Liu *et al.* [91] present a time-series pattern based algorithm to forecast duration intervals. In this thesis, to address a probability fluctuation privacy risk, the time-series pattern is an effective tool to forecast "future" occurrence probabilities based on past data probabilities in the situation with probability fluctuations. And that is our novel time-series pattern based noise generation strategy as the noise generation strategy. We will utilise the time-series pattern in Chapter 5 in detail.

In the area of association analysis, [4] and [6] start to consider privacy protection in association rules mining, and they are useful references for our novel noise generation strategy in terms of association probability in this thesis. For example, these association probabilities of past service data can be utilised to manage noise generation, and be concealed to prevent service providers from revealing them by association rules mining. To deal with the association analysis privacy risk, we will utilise the association analysis by our novel association probability based noise generation strategy in Chapter 6 in detail.

About intersection attacks, data publish and data mining areas are the main fields to be discussed for privacy protection. For example, the composition attack [80] is one typical intersection attack in the view of database. During these attacks, some multidimensional adversarial knowledge [11] can be utilised to quantify the breach of private information and improve algorithms to sanitise data with external knowledge. Specifically, Malin *et al.* [15] investigate this problem in the situation of

inferring genotype from clinical phenotype. Briefly, the malicious intersection private data analysis is a significant privacy risk and we have to consider it in the unethical multiple services case in terms of noise utilisation in our noise obfuscation model. To address the unethical multiple services case, we will utilise it as a key reference of our novel common set based noise cooperation strategy in Chapter 8 in detail.

In summary, these supporting issues can be utilised in our research. They are valuable references to be investigated to support the main topic of this thesis—noise obfuscation for systematically and comprehensively supporting cloud privacy protection.

2.4 Summary

In this chapter, the literature for the recent studies related to the privacy protection in cloud computing has been analysed. Firstly, we introduced and discussed current successful privacy protection approaches at service side, and pointed out these approaches can not deal with all privacy protection scenarios addressed in cloud computing, especially the one in this thesis. Then, we introduced existing privacy protection approaches at client side, and indicated that the noise obfuscation approach is a suitable approach to deal with the scenario in this thesis. Meanwhile, based on some other supporting approaches including operating model in distributed systems, trust and privacy risk evaluation, time-series analysis, association analysis and intersection attack, for cloud privacy protection we can develop novel and promising noise obfuscation model and strategies for protecting privacy in cloud computing as detailed in the subsequent chapters.

Chapter 3

Noise Obfuscation Model for Privacy Protection in Cloud Computing

As introduced before, this thesis focuses on the noise obfuscation approach for privacy protection in cloud computing. Therefore, in this chapter, we present a novel general model of noise obfuscation function for privacy protection in cloud computing—the noise obfuscation model. This novel noise obfuscation model is to abstract the whole novel noise obfuscation philosophy to enhance privacy protection in cloud computing. Based on the model in this chapter, the technical details about noise obfuscation including noise pre-processing strategies, noise generation strategies and noise utilisation strategies, will be described in the subsequent chapters. In one word, the novel noise obfuscation model in this chapter is the foundation of this entire thesis.

This chapter is organised as follows. Section 3.1 introduces an overview of privacy protection in cloud computing. Section 3.2 presents the novel noise obfuscation model for privacy protection in cloud computing. Section 3.3 introduces the design of the first component in this model, i.e. noise pre-processing. Section 3.4 introduces the design of the second component, i.e. noise generation. Section 3.5 introduces the design of the last component, i.e. noise utilisation. Section 3.6 introduces our SwinCloud cloud computing infrastructure briefly as the simulation environment. Finally, Section 3.7 summarises this chapter.

3.1 Privacy Protection Overview in Cloud Computing

As discussed before, noise obfuscation is one kind of cloud privacy protection approaches. Hence, an overview about privacy protection in cloud computing expresses a comprehensive picture of cloud privacy protection to underline noise obfuscation as a novel and promising approach, which was partly discussed in our former work [40].

Firstly, the cloud architecture [75] has been discussed before, as depicted in Figure 3-1. And there are various levels in this architecture.

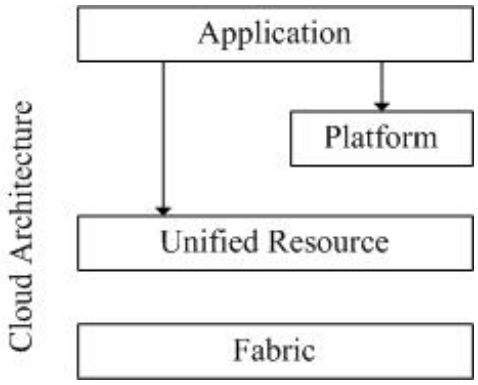


Figure 3-1 Cloud Architecture

Based on this architecture, cloud service levels can be presented in the view of cloud customers, as shown in Figure 3-2.

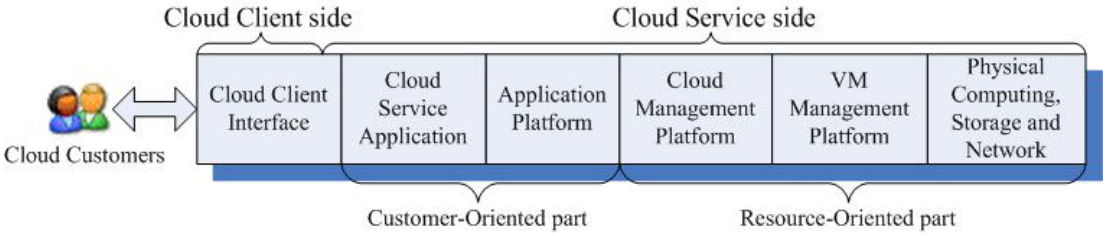


Figure 3-2 Cloud Service Levels in the view of Cloud Customers

In Figure 3-2, from the perspective of cloud customers, it is a suitable view to classify privacy protection approaches in cloud environments by cloud service levels. Under these levels, various cloud privacy protection approaches can be organised to describe the overview.

Besides, similar to security in cloud computing [65], as the two main roles in cloud environments, cloud customers and cloud services providers have to

investigate privacy protection in cloud service processes among them. Hence, as discussed in Chapter 2, we classified cloud privacy protection into client side and service side.

Accordingly, in Figure 3-3, we can draw an overview of cloud privacy protection, similar to other systematic analysis in different areas [52, 25].

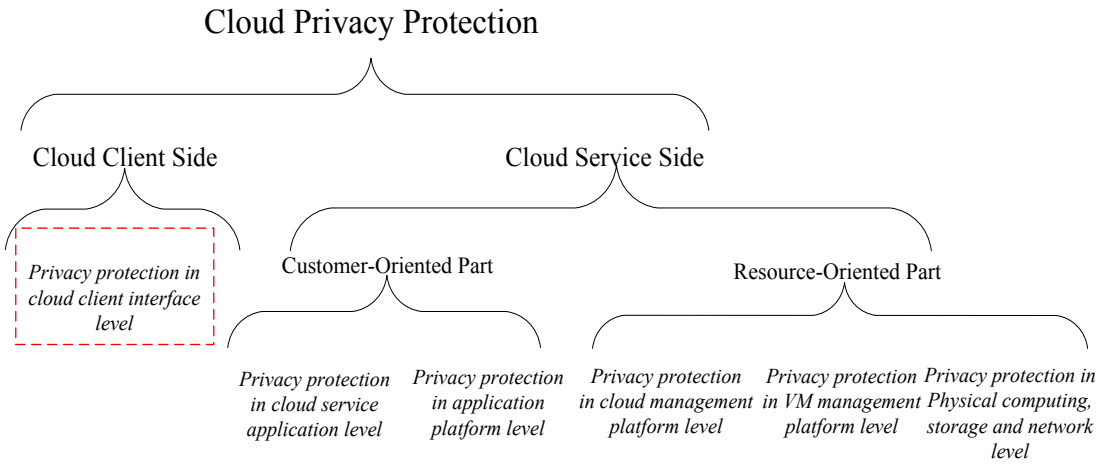


Figure 3-3 Privacy Protection in Cloud Environments

In Figure 3-3, based on these roles—cloud customers and cloud service providers, we can divide cloud service levels into two parts: cloud client side and cloud service side, respectively. Hence, two kinds of privacy protection approaches can be discussed: privacy protection approaches at cloud service side, and privacy protection approaches at cloud client side. As introduced before, at cloud client side, this thesis focuses on noise obfuscation which is located in the box on the left—privacy protection in cloud client interface level.

In this thesis, privacy protection considers how to use cloud services safely dependent on clients own. Hence, based on this semi-honest condition, some instructive approaches and ideas have been discussed, including some qualitative methods [27]. In Chapter 2, we introduced some current representative privacy protection work at client side. They are secure computation, homomorphic encryption and noise obfuscation. As discussed in Chapter 2, compared to noise obfuscation, secure computation and homomorphic encryption have to face the low efficiency problem which can limit them in cloud computing.

In summary, noise obfuscation is a key and promising approach to support powerful, comprehensive and efficient privacy protection in complex and

unpredictable cloud environments. Moreover, we will introduce our novel noise obfuscation model for privacy protection in cloud computing with its components in the following sections.

3.2 Noise Obfuscation Model for Privacy Protection

In the preceding section, we realised that noise obfuscation is a key and promising approach in cloud privacy protection. Now, we present our novel noise obfuscation model for privacy protection in cloud computing, and give a general picture about our novel noise obfuscation for privacy protection in cloud computing.

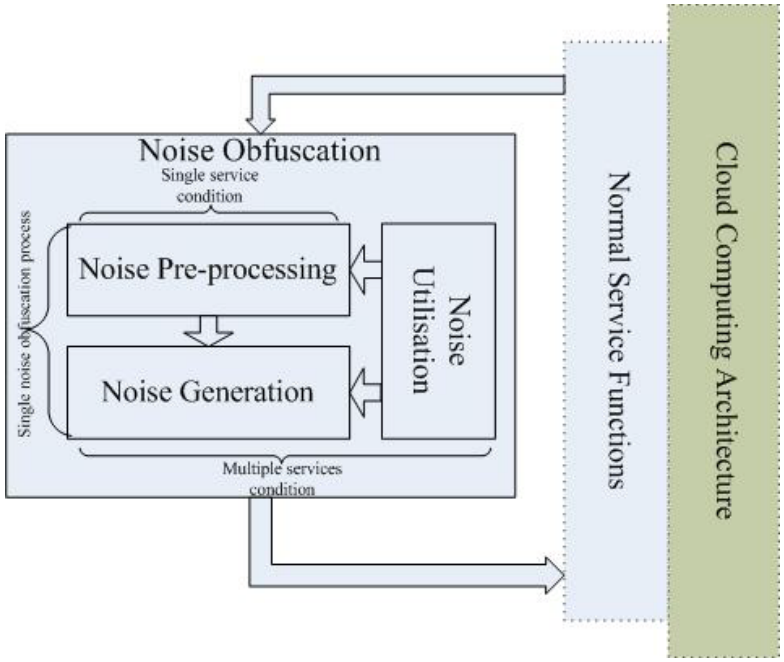


Figure 3-4 Noise Obfuscation Model

In Figure 3-4, our novel noise obfuscation model for privacy protection in cloud computing is presented. The main process of this model is that: there are three main components which can be executed successively in noise obfuscation functions: "Noise Pre-processing", "Noise Generation" and "Noise Utilisation". The "Noise Pre-processing" component focuses on some pre-setting work in noise obfuscation functions, such as connecting to customers' privacy requirements by privacy-leakage-tolerance described in this thesis. The "Noise Generation" component is the core part of noise obfuscation and in charge of how to generate effective and efficient noise data after noise pre-processing to withstand various privacy risks. For

example, fluctuations of occurrence probabilities can be analysed by malicious service providers to break existing noise obfuscation, and a novel noise generation strategy should be designed to conceal these fluctuations for privacy protection. The former two components only consider the single service scenario and are executed sequentially as a single noise obfuscation process in this single service scenario. Hence, the "Noise Utilisation" component considers how to utilise these single noise obfuscation processes together and connect them in the multiple services scenario. In other words, it has to execute the former two components in this multiple services scenario. For instance, when unethical multiple services may cooperate together to share private information and break existing noise obfuscations, to address this kind of privacy risks, noise utilisation has to consider how to utilise single noise obfuscation processes efficiently and effectively in the unethical multiple services case. In general, when these three components are combined together, the entire novel noise obfuscation functions for privacy protection in cloud computing can be operated and functional.

In parallel to noise obfuscation functions, normal service functions focus on their operation and execution in cloud environments. On one hand, these normal cloud service functions are the privacy protected targets of noise obfuscation. In other words, theses cloud service functions decide noise obfuscation in terms of privacy protection, not the other way round. On the other hand, cloud computing architecture is the basis of these normal service functions. Hence, these normal cloud service functions have some typical cloud features, like openness and virtualisation, or all possible cloud management and service delivery.

In this model and the thesis, we focus on noise obfuscation itself. Hence, the "Normal Cloud Service Functions" and the "Cloud Computing Architecture" are external supporting factors for noise obfuscation and out of the scope of this thesis.

In brief, we introduce our novel noise obfuscation model for privacy protection in cloud computing. It combines and assembles three main components—"Noise Pre-processing", "Noise Generation" and "Noise Utilisation", and organises them to be an entirety to client-side protect privacy for normal cloud service functions in cloud environments. Besides, existing noise obfuscations [77, 42] mainly focus on one part of the model: noise generation.

In the following sections, we will introduce these components one by one in

detail.

3.3 Noise Pre-processing Component

In this section, we introduce the noise pre-processing component in our noise obfuscation model, which is the first step of the single noise obfuscation process and the whole noise obfuscation function. As introduced before, this component operates itself as a necessary way to connect between noise obfuscation and cloud customers who use noise obfuscation to protect their privacy in cloud computing. In this thesis, we focus on privacy-leakage-tolerance as the main way to require noise obfuscation functions by cloud customers from the perspective of privacy protection.

Generally speaking, privacy-leakage-tolerance can decide the noise set's creation to control noise generation processes in the next step—noise generation. Specifically, for cloud customers, noise obfuscation has to meet some customers' privacy requirements, such as a probability of privacy leakage which they can accept. That is privacy-leakage-tolerance. Besides, a noise set is a set of all possible noise data during the process of noise generation. Hence, the noise set is the key premise of noise generation, which will be illustrated in the following chapters. In this component, it is suitable to link the customers' privacy requirements and noise obfuscation functions by privacy-leakage-tolerance. It can decide the creation of the noise set to improve the efficiency of privacy protection, due to the more accurate requirements than without it. That is the main idea of the noise pre-processing component.

In this component, as shown in Figure 3-5, we can use the specific noise pre-processing strategy to operate this main idea, which is the novel privacy-leakage-tolerance based noise enhancing strategy. In this strategy, privacy-leakage-tolerance can be set by cloud customers firstly, and then the noise set, which includes all possible noise data used by the noise obfuscation function, can be decided and created from this privacy-leakage-tolerance by the noise set creation model. We will introduce this strategy in detail in Chapter 4.

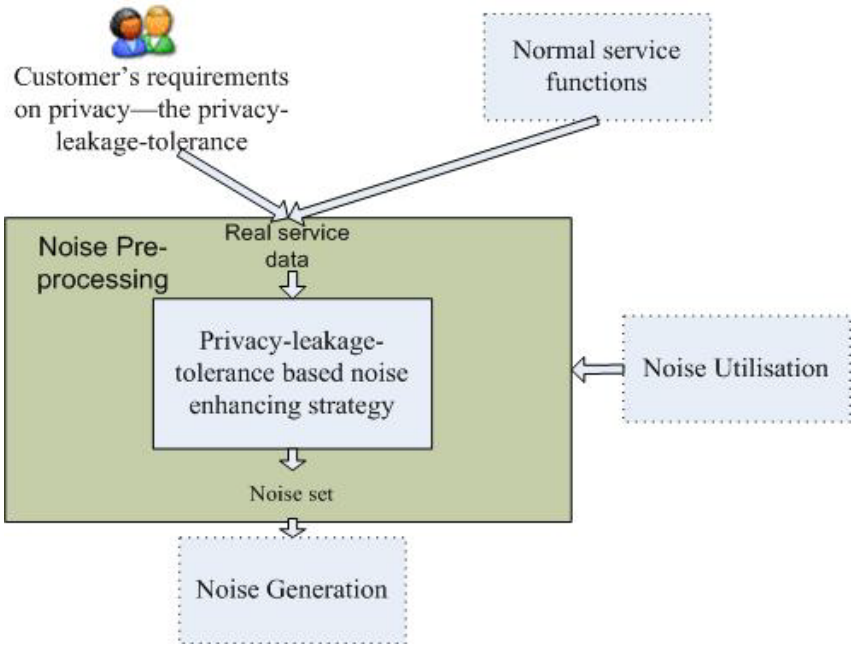


Figure 3-5 Noise Pre-processing Component

Besides, in this figure, we can see that this component could be directed by the noise utilisation component to support the multiple services scenario, which will be introduced in Section 3.5.

Briefly, the noise pre-processing component uses the noise pre-processing strategy to generate the noise set from the customer requirement on privacy (privacy-leakage-tolerance) to support the noise generation component. It is the first step of the single noise obfuscation process which is the basis of the whole noise obfuscation function.

3.4 Noise Generation Component

In the preceding section, we introduced the noise pre-processing component to obtain the noise set as the first step in the single noise obfuscation process. Hence, in this section, we can utilise the noise set in the noise generation component to generate efficient and effective noise data, such as service requests. For the single service scenario, this noise generation component and the previous noise pre-processing component can be executed as a single noise obfuscation process. Hence, as discussed before, the noise generation component is the key part of the single

noise obfuscation process and the whole noise obfuscation function.

In Figure 3-6, we discuss the input and output of this noise generation component: the results of the noise pre-processing component—the noise set is the main input of this component, and the output of this component are noise data which can be utilised by normal service functions to protect privacy in cloud computing. In short, the noise generation component focuses on how to generate noise data to obtain efficient and effective privacy protection in the single noise obfuscation process, based on the noise set.

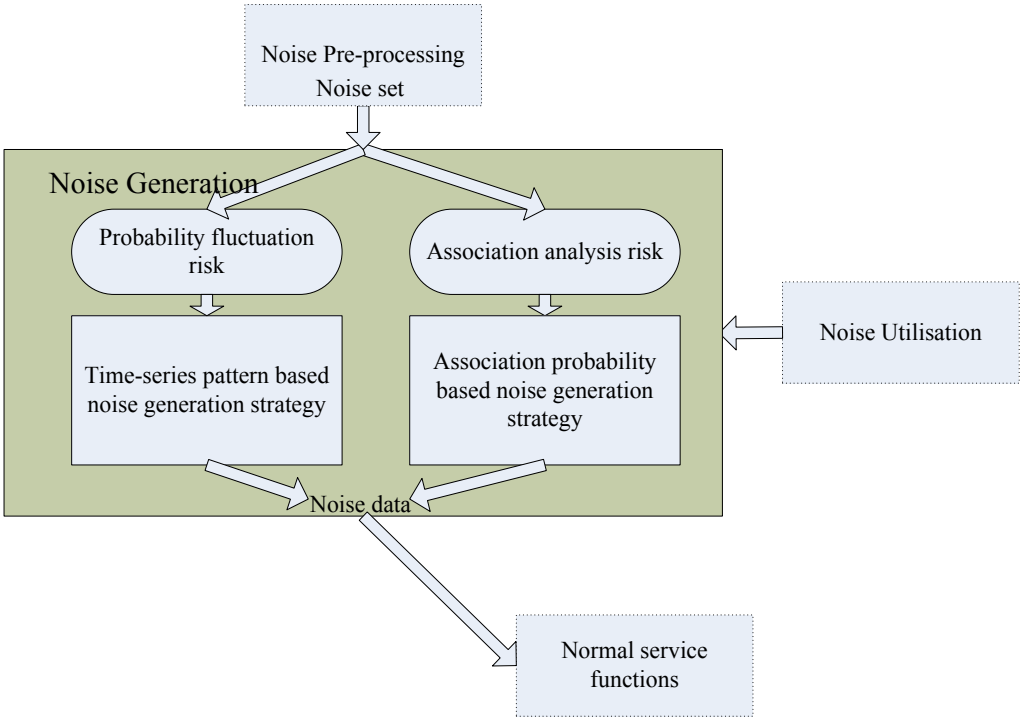


Figure 3-6 Noise Generation Component

In this component, we design some noise generation strategies to obtain effective and efficient noise generation by addressing some privacy risks. And these are two serious privacy risks for noise generation considered in this thesis: the probability fluctuation privacy risk and the association analysis privacy risk.

The probability fluctuation privacy risk means that malicious attackers can break existing noise obfuscations by analysing these fluctuations of occurrence probabilities to obtain customer privacy. Hence, for this serious privacy risk, we design the novel time-series pattern based noise generation strategy. In short, this noise generation strategy can analyse past real service data with fluctuations of occurrence probabilities, and generate noise data to conceal these fluctuations by

time-series patterns' creation and forecasting. We will introduce this strategy in detail in Chapter 5.

The association analysis privacy risk means that these association probabilities among different real service data are the private information which could be found by malicious privacy attackers, and should be protected. Hence, we design the novel association probability based noise generation strategy. This strategy can analyse past association probabilities of real service requests, and focus on concealing these association probabilities by the association probability model. We will introduce this strategy in detail in Chapter 6.

Generally speaking, these two privacy risks represent two main ways to break existing noise generation for privacy attackers: one uses external features of service data, such as time-series; another one uses internal features of service data, such as association probability. For specific noise obfuscation, these risks have to be analysed and addressed under specific requirements.

Besides, just like the noise pre-processing component, in Figure 3-6, we can find out that this component could be directed by the noise utilisation component to support the multiple services scenario, too. And, in Section 3.5 about the noise utilisation component, we can utilise these two components together as a single noise obfuscation process and combine several single noise obfuscation processes to match the multiple services scenario.

In summary, this noise generation component can generate effective and efficient noise service data, such as noise service requests, based on the noise set which can be obtained by the previous noise pre-processing component. And this component and the noise pre-processing component can make up the single noise obfuscation process to execute the noise obfuscation function in the single service scenario. This component is the core step of the single noise obfuscation process and the whole noise obfuscation function.

3.5 Noise Utilisation Component

In the preceding sections, we introduced the noise pre-processing component and the noise generation component for the single noise obfuscation process with different

strategies. Hence, in this section, we can utilise these strategies to support the noise utilisation component by matching privacy concerns or risks in multiple services scenarios. This is the noise utilisation component which is an essential part of the whole noise obfuscation function.

In Figure 3-7, we show that this component analyses privacy concerns and risks in multiple services scenarios, and utilises several single noise obfuscation processes together to protect privacy in the multiple services scenario. In other words, this component manages the former two components to obtain efficient and effective privacy protection in the cloud multiple services scenario.

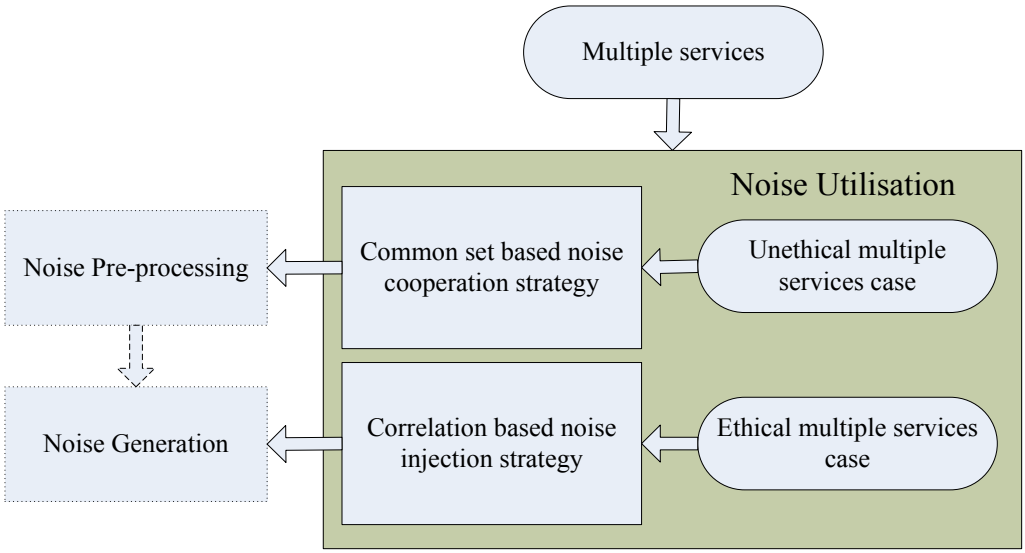


Figure 3-7 Noise Utilisation Component

In this component, we design some novel noise utilisation strategies to operate this noise utilisation function. These strategies focus on noise utilisation processes during normal cloud service functions. Hence, these are two different cases for noise utilisation: the ethical multiple services case and the unethical multiple services case.

The ethical multiple services case means that these ethical cloud services can cooperate together to withstand privacy attackers—other malicious service providers in terms of noise obfuscation. For this ethical multiple services case, we design the novel correlation based noise injection strategy to match the privacy concern. In short, this strategy can analyse correlations among different ethical cloud service providers by the correlation model, and pursue effective noise utilisation based on managing single noise obfuscation processes. We will introduce this strategy in detail in Chapter 7.

The unethical multiple services case means that cloud customers have to deal with unethical services which could cooperate together to break privacy protection, like noise obfuscation. Hence, for this unethical multiple services case, we design the novel common set based noise cooperation strategy. In general, this strategy can analyse this privacy risk, and execute the noise utilisation component by the common set to keep noise obfuscation effective in this case. We will introduce this strategy in detail in Chapter 8.

In summary, this noise utilisation component can utilise single noise obfuscation processes to protect privacy in the multiple services scenario. In other words, it can be executed based on guiding several single noise obfuscation processes which are composed of noise pre-processing components and noise generation components. In the view of cloud privacy protection, it is a necessary step to deal with these multiple services scenarios for noise obfuscation.

3.6 Simulation Environment

Our novel noise obfuscation for privacy protection in cloud computing proposed in this thesis is being gradually implemented as system components in our SwinCloud [98, 99, 55, 92].

The SwinCloud infrastructure is depicted in Figure 3-8.

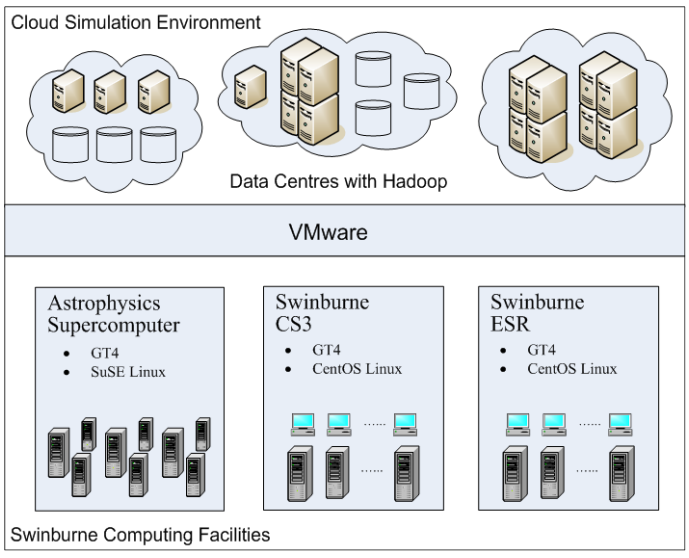


Figure 3-8 SwinCloud Infrastructure

SwinCloud is a cloud computing simulation test bed. It is built on the computing

facilities in Swinburne University of Technology and takes advantage of the existing SwinGrid system. For example, the Swinburne Astrophysics Supercomputer Node comprises 145 nodes of Dell Power Edge 1950 nodes each with: 2 quad-core Clovertown processors at 2.33 GHz (each processor is 64-bit low-volt Intel Xeon 5138), 16 GB RAM and 2 x 500 GB drives. We install VMWare [88] to offer unified computing and storage resources. By utilising the unified resources, we set up data centres that can host applications. In the data centres, Hadoop [44] is installed that can facilitate MapReduce [37] computing paradigm and distributed data management.

Generally speaking, SwinCloud is the cloud simulation environment for our novel noise obfuscation for privacy protection in cloud computing. In the following chapters, we will use the cloud environment as the basic platform to simulate cloud privacy attacking and protection to illustrate our noise obfuscation, without internal details about this environment.

3.7 Summary

In this chapter, we have presented an overview about our novel noise obfuscation for privacy protection in cloud computing, and proposed the novel noise obfuscation model showing a general view of our novel noise obfuscation for privacy protection in cloud computing. In this model, three components discussed can be utilised to fulfil the whole noise obfuscation functions: noise pre-processing, noise generation and noise utilisation. Besides, the first two components can be executed sequentially as the single noise obfuscation process, and the last component can be executed based on this process. In each component, some strategies have been described briefly while the details will be presented in subsequent chapters of this thesis. We have also briefly introduced the SwinCloud cloud computing simulation test bed which serves as the simulation environment for our noise obfuscation strategies.

Chapter 4

Noise Pre-Processing by Privacy-Leakage-Tolerance

As discussed in Chapter 3, the noise pre-processing component is the first step in the single noise obfuscation process and the novel noise obfuscation model. In this component, cloud customers could utilise noise pre-processing strategies to create the noise set for noise generation in the next step, based on the customer-set privacy-leakage-tolerance. Hence, in this chapter, we introduce our novel noise pre-processing strategy: the privacy-leakage-tolerance based noise enhancing strategy. By facilitating this noise pre-processing strategy, cloud customers can manage noise obfuscation in terms of the efficiency of privacy protection by the creation of the noise set. The customer-set privacy-leakage-tolerance is the key factor to create the noise set in noise pre-processing to connect the probability of privacy leakage and the noise obfuscation functions. This is the main idea of the noise pre-processing component in our novel noise obfuscation model.

Briefly, under the privacy concern about the bridge between customers' requirements and noise obfuscation functions, the novel privacy-leakage-tolerance based noise enhancing strategy can be presented as the main issue of this chapter. This strategy investigates the creation of the noise set by the customer-set privacy-leakage-tolerance. Hence, the following noise generation and the whole noise obfuscation function can be more accurate for privacy protection in the pay-as-you-go fashion for cloud computing. Besides, the comparison results demonstrate that our strategy can enhance noise obfuscation performance in terms of the efficiency of

privacy protection.

This chapter is organised as follows. Section 4.1 presents the background of the strategy. Then, Section 4.2 proposes the Privacy-leakage-Tolerance based Noise Injection Model (*PTNIM*). Section 4.3 presents the novel Privacy-leakage-Tolerance based Noise set Creation Model (*PTNCM*). Section 4.4 proposes our novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) for privacy protection in cloud computing. Section 4.5 discusses simulation and evaluation to illustrate this strategy. Finally, Section 4.6 summarises the chapter.

4.1 Background of the Strategy

Generally speaking, privacy protection is critical as one of the most concerned issues in cloud computing [70, 79, 22, 62]. Based on data obfuscation [23], noise obfuscation is an effective approach for cloud privacy protection. To fulfil different requirements on noise obfuscation, different kinds of noise strategies have been presented [77, 42, 43]. But currently, existing noise obfuscations do not consider and investigate the impact from a customer-defined privacy-leakage-tolerance in terms of noise pre-processing.

Actually, it is a natural concern that a cloud customer evaluates the privacy leakage risk with some boundaries or tolerances before he/she uses services in cloud environments, no matter automatically or not. Hence, for cloud customers, these customer-defined boundaries or tolerances on the probability of privacy leakage are important issues to evaluate and manage noise obfuscation processes in terms of privacy protection. In other words, these specific privacy-leakage-tolerances can give cloud customers specific choices on noise obfuscation in terms of noise pre-processing. For instance, a cloud service provider may have a low ‘privacy risk’ for a customer, which means that the cloud customer or the client may ‘give’ the service provider a high privacy-leakage-tolerance. And it is an evaluation by this customer. Noise obfuscation could use this tolerance to guide the specific noise obfuscation process by controlling the volume of noise data utilised. Hence, the customer or noise obfuscation could reduce the volume of noise data injected into real ones in noise obfuscation processes under this boundary or tolerance. And he/she can obtain

a lower cost on noise data with a reasonable effectiveness of privacy protection based on this tolerance. In a word, for a service provider with a 'high' privacy-leakage-tolerance, less (or ever no) noise data need to be utilised by noise obfuscation in terms of noise pre-processing; and for a service provider with a 'low' privacy-leakage-tolerance, more noise data should be utilised by noise obfuscation in terms of noise pre-processing.

But existing noise obfuscations do not consider this so far. As a result, they have to utilise a large number of noise data to obtain a reasonable level of privacy protection without a specific privacy-leakage-tolerance, which means a higher cost on noise data in cloud environments.

To address this, considering existing noise obfuscations [42, 43], a noise set is a key issue to generate noise data, and connect a privacy-leakage-tolerance to a specific single noise obfuscation process. It includes all possible noise data which could be utilised in noise obfuscation, such as in noise service requests, based on the customer-set privacy-leakage-tolerance. At service side, malicious service providers cannot distinguish which requests are real ones based on this set. Hence, the size of this noise set can describe the intensity of noise obfuscation to some extents. It means cost-saving at customers' wishes. That is the main idea of this strategy.

Hence, we propose our novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) for privacy protection in cloud computing. Based on existing noise generation strategies, we firstly analyse the privacy-leakage-tolerance from cloud customer as the customer-set privacy risk boundary. Then, we present a novel privacy-leakage-tolerance based noise set creation model to describe this noise set's creation process. Finally, we present our novel *PTNES* as the noise pre-processing strategy to improve the efficiency of privacy protection in the noise pre-processing component.

Based on the previous weather forecast service example described in Section 1.2.2, we can take a motivating example to describe this strategy. One customer, who often travels to one city in Australia, like 'Sydney', checks the weather report regularly from a weather forecast service in cloud environments before departure. The regular appearance of service requests about the weather report for 'Sydney' can reveal the privacy that the customer usually goes to 'Sydney'. But if a system aids the customer by injecting other requests like 'Melbourne', 'Perth' or 'Darwin'

into the ‘Sydney’ queue, the service provider cannot distinguish which ones are real and which ones are ‘noise’ as it just sees a similar style of service request. These requests should be responded and would not reveal the location privacy of the customer. In such cases, the privacy can be protected by noise obfuscation in general. Considering the privacy-leakage-tolerance in this strategy, if the customer has a high privacy-leakage-tolerance for the service provider, the customer may only need a small request set, like one set with two options: ‘Sydney’ and ‘Melbourne’ to conceal this privacy from the service provider. In other words, the high tolerance can give the service provider some ‘trust’ from the customer in terms of noise obfuscation. Therefore, ‘Perth’, ‘Darwin’ and so on are ‘useless’ for enhancing the effectiveness of privacy protection but incur some extra unnecessary cost. Hence, reducing this unnecessary cost based on the privacy-leakage-tolerance is the main motivation of this strategy.

In the following sections, firstly, we plan to introduce the Privacy-leakage-Tolerance based Noise Injection Model (*PTNIM*) to support our novel *PTNES*. Secondly, the creation of the noise set is introduced, and the novel Privacy-leakage-Tolerance based Noise set Creation Model (*PTNCM*) can summarise it. Thirdly, our novel *PTNES* is presented. Lastly, some simulation evaluations can illustrate this strategy from the perspective of privacy protection.

4.2 *PTNIM*: Privacy-Leakage-Tolerance based Noise Injection Model

In this section, we introduce the Privacy-leakage-Tolerance based Noise Injection Model (*PTNIM*) to support *PTNES*. As introduced before, *PTNES* is a noise pre-processing strategy in the noise pre-processing component of our novel noise obfuscation model. And the noise injection model is necessary to execute these noise obfuscation processes. From [77, 41, 42, 43], different noise injection models are built for different noise obfuscation strategies, respectively. Accordingly, we present *PTNIM* in Figure 4-1 based on the former work.

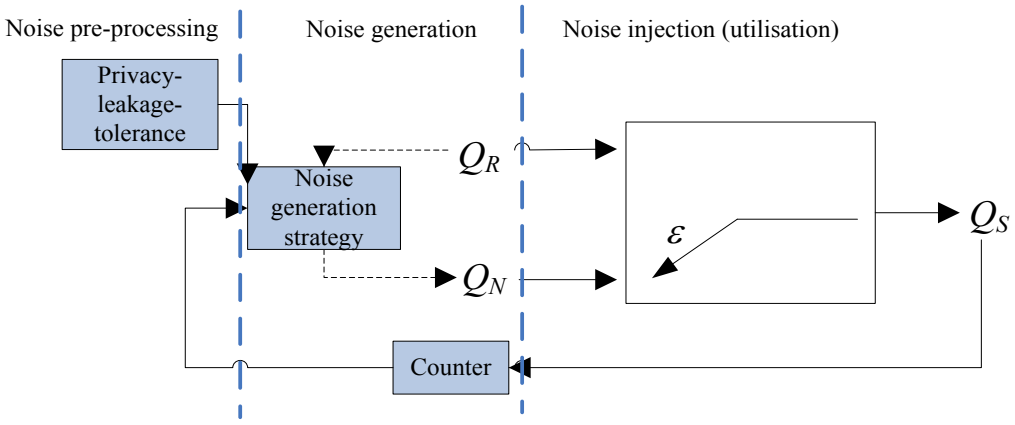


Figure 4-1 *PTNIM*: Privacy-leakage-Tolerance based Noise Injection Model

Denotations in Figure 4-1 are listed as follows:

Q_R: queue of customer's real service requests which are to be protected.

Q_N: queue of noise service requests which are to be injected into Q_R.

Q_S: queue of final service requests composing of Q_R and Q_N.

Q: a set of all service requests, and $Q = \{q_1, q_2, ..., q_i, ..., q_n\}$. Every service request in Q_R, Q_N and Q_S is from this set. Hence, in the view of service providers, service requests in the queue of final service requests Q_S could be from real requests Q_R or noise requests Q_N.

ε: probability for injecting Q_N into Q_R, and $\varepsilon \in [0,1]$. We call it noise injection intensity.

The overall working process of the model is to inject Q_N into Q_R based on ε so that we can get Q_S. We need to utilise past Q_R and Q_S to generate Q_N by 'noise generation strategy' and then apply them into noise obfuscation. As a part of noise pre-processing, 'privacy-leakage-tolerance' guides 'noise generation strategy' by the creation of noise set, and we would detail these processes in the following sections. Besides, noise injection intensity ε is an important parameter from 'noise generation strategy'.

With this *PTNIM*, we can present the novel Privacy-leakage-Tolerance based Noise set Creation Model (*PTNCM*) in the next subsection to support our novel *PTNES*.

4.3 *PTNCM*: Privacy-Leakage-Tolerance based Noise Set Creation Model

In this section, we present the key part of our novel *PTNES*—Privacy-leakage-Tolerance based Noise set Creation Model (*PTNCM*). This model can be presented in two aspects. Firstly, the privacy leakage risk evaluation under noise obfuscation can be analysed to compare with the privacy-leakage-tolerance. Then, the Privacy-leakage-Tolerance based Noise set Creation Algorithm (*PTNCA*) can be proposed to describe the creation procedure based on the previous subsection.

4.3.1 Privacy Leakage Risk Evaluation

In this subsection, we investigate the evaluation problem of privacy leakage risk under noise obfuscation. Based on existing noise obfuscation work [77, 42, 43], to evaluate the privacy leakage risk under noise obfuscation, we discuss the original set of all service requests:

$$Q = \{ q_1, q_2, ..., q_i, ..., q_n \} \qquad \textbf{Formula 4-1}$$

Based on this set, we can present a map $f : q_i \to d_i$ to obtain data items d_i in service requests q_i. In the example of Section 1.2.2, this city information, like 'Sydney' and 'Melbourne' can be viewed as data items d_i from service requests q_i in the weather service process. Because these data items are potential private information for cloud customers, it is clear that in this strategy, this map is an injective map as well as a surjective map. Thus, it is a bijective map. We can get:

$$\forall i \in [1, n], d_i = f(q_i) \qquad \textbf{Formula 4-2}$$

And the original noise set is:

$$D = \{ d_1, d_2, ..., d_i, ..., d_n \} \qquad \textbf{Formula 4-3}$$

Besides, the final noise set is a part of the original noise set:

$$D_N = \{ d_1, d_2, ..., d_i, ..., d_m \}$$ **Formula 4-4**

where $n \geq m \geq 0$.

In a noise set, there may be one or more items as real private data that should be protected as customer privacy. In other words, the data item(s) should be concealed by other noise data items in the noise set with similar occurrence probabilities. Hence, we have:

$$m = x + y$$ **Formula 4-5**

The number of the private data item(s) is x, and the number of other noise data item(s) is y. Hence, we can evaluate the privacy leakage risk under noise obfuscation with x and y. And the set of these private data items is D_x, the set of other noise data items is D_y. Accordingly, we have the join of these two sets:

$$D_N = D_x \cup D_y$$ **Formula 4-6**

In D_N, malicious service providers could find out the real private data items with a probability, and we can set this probability as the privacy leakage risk. In this probability, 'unethical' service providers cannot get extra information from other sources to increase the probability to guess the real private ones. Hence, we can list all possible conditions of real private data items' leakage and combine them to obtain the Real Privacy Leakage Risk under noise obfuscation (PLR_R) which is:

$$PLR_R = plr(x,y) = \frac{1}{x} * \frac{1}{C_{x+y}^1} * \frac{1}{C_x^1} + \frac{2}{x} * \frac{1}{C_{x+y}^2} * \frac{1}{C_x^2} + ... + \frac{x}{x} * \frac{1}{C_{x+y}^x} * \frac{1}{C_x^x} = \sum_{i=1}^{x} (\frac{i}{x} * \frac{1}{C_x^i * C_{x+y}^i})$$ **Formula 4-7**

In formula 4-7, we should consider that malicious service providers may guess parts of real private ones under noise obfuscation, instead of all real ones. Hence, the corresponding risk can be reduced. In this formula, the first item in the polynomial formula means the probability of that one real private data item can be revealed by service providers; the second one means the probability of that two real private data items can be revealed; and so on. In each item, the exhaustive method can be utilised to list all possible cases under one specific number of real private data item. In other words, we consider that 'unethical' service providers may guess parts of real private

ones based on noise obfuscation. Hence, we can utilise formula 4-7 to evaluate the privacy leakage risk.

4.3.2 *PTNCA*: Privacy-leakage-Tolerance based Noise Set Creation Algorithm

From formula 4-7, we get the real privacy leakage risk or the probability of privacy leakage in terms of noise pre-processing. Hence, in this subsection, we introduce the creation process of the noise set by privacy-leakage-tolerance.

As introduced before, for cloud customers, the requirement for the privacy leakage probability is necessary to guide service processes by cloud customers. This is the Customer-defined Privacy-Leakage-Tolerance (PLT_C), and it is obvious that:

$$PLT_C \geq PLR_R \qquad \textbf{Formula 4-8}$$

From formulas 4-7 and 4-8, we can get:

$$\sum_{i=1}^{x}\left(\frac{i}{x}\times\frac{1}{C_x^i\times C_{x+y}^i}\right)\leq PLT_C \qquad \textbf{Formula 4-9}$$

Based on that the domain of y is $[0,+\infty]$, it is clear that the more items we have in the noise set, the lower privacy risk we can obtain. So:

$$plr(x,y) > plr(x,y+1) \qquad \textbf{Formula 4-10}$$

Besides, formula 4-7 can be monotonically decreased with y increasing. Hence, to satisfy the privacy requirement introduced before, we can get y:

$$y \geq plr^{-1}(PLT_C) \qquad \textbf{Formula 4-11}$$

In formula 4-11, we can use the inverse function to get y. According to this formula, this inverse function is monotonic too. So, we can use stepwise refinement to implement it. Besides, x is fixed under a specific privacy protection condition, and we can omit it in this formula. Hence, the minimum of y is:

$$y_{min} = plr^{-1}(PLT_C) \qquad \textbf{Formula 4-12}$$

Based on y_{min}, we can consider how to obtain D_N.

As discussed before, D_N can express the effectiveness of privacy protection, and the cost of privacy protection on noise obfuscation can be decided by D_N, too. To get a low cost on noise obfuscation with the same effectiveness of privacy protection, we have to consider how to create D_N based on y_{min}.

For each data item d_i from D, if it is chosen as one in D_N, the specific noise cost on this data item should be decided by specific noise generation strategies. The total cost is:

$$Cost(Strategy, D_N) = \sum_{d_i \in D_N} Cost(Strategy, q_i) \qquad \textbf{Formula 4-13}$$

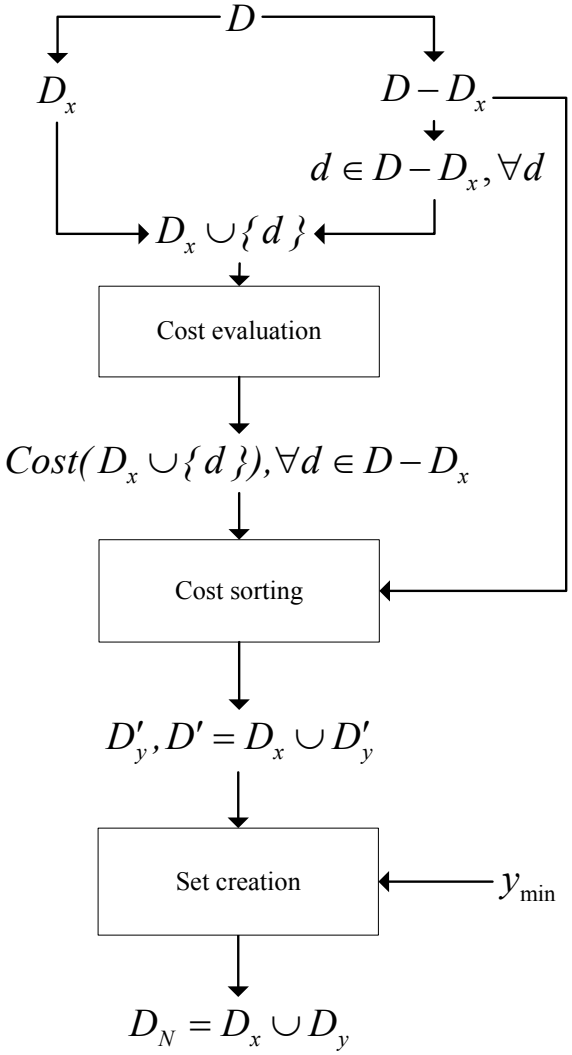
With a fixed $y = y_{min}$, to get the lower $Cost(Strategy, D_N)$, we investigate the original noise set: $D = \{d_1, d_2, ..., d_i, ..., d_n\}$, and its sorted set $D' = D_x + D'_y$. D'_y is from $D - D_x$ with being sorted by cost evaluation. Hence, D_y can be created by the lowest y_{min} data item(s) in D'_y. That is the main idea of *PTNCA* described in Algorithm 4-1.

In Algorithm 4-1, 'Cost evaluation' can evaluate every possible data item in D_y from the perspective of noise cost. In formula 4-13, the independence among data items in the noise set is an important issue to evaluate the cost on every data item in the noise set.

'Set creation' creates D_y based on sorted D'_y and y_{min}. The first y_{min} data items of D'_y make up D_y and the noise set $D_N = D_x \cup D_y$. Hence, a noise request set can be mapped from D_N:

$$Q_N = f^{-1}(D_N) \qquad \textbf{Formula 4-14}$$

In general, based on the privacy leakage risk evaluation and *PTNCA*, *PTNCM* can be built to support *PTNES* for privacy protection in cloud computing by controlling D_N and Q_N.



Algorithm 4-1 *PTNCA*: Privacy-leakage-Tolerance based Noise set Creation Algorithm

4.4 *PTNES*: Privacy-Leakage-Tolerance based Noise Enhancing Strategy

Based on the previous sections, we now present our novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) for privacy protection in cloud computing in Algorithm 4-2.

In Algorithm 4-2, we present our novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*). Based on *PTNCA* and *PTNCM*, *PTNES* investigates the privacy leakage risk evaluation and the privacy-leakage-tolerance under noise obfuscation in the noise pre-processing component.

In this algorithm, Step 1 is the beginning step to collect all request queues and sets as past data to support the subsequent steps. In Step 2, we get the real privacy leakage risk evaluation. In Step 3, we obtain the key issue of the strategy: the noise set D_N. In this step, the size of D_N can be decided by the privacy-leakage-tolerance firstly. Then, based on the specific noise generation strategy, we can evaluate each cost on each possible other noise data item to obtain the lower cost with a fixed effectiveness of privacy protection on noise obfuscation. Briefly, we use $D_N = PTNCA(Strategy, D, y_{min})$ to describe the function of this algorithm. In Step 4, noise service requests can be generated and utilised under *PTNES* enhancing. After this, Steps 2, 3 and 4 would be executed again as a run-time privacy protection mechanism until the whole noise obfuscation terminates.

Title: Privacy-leakage-tolerance based noise enhancing strategy
Input: the queue of real service requests is Q_R
the customer-defined privacy-leakage-tolerance is PLT_C
the size of real private data items in the noise set is x
Output: the queue of final service requests is Q_S

Step 1: Collect the original noise set
Collect and record the service request queue and service request set in past time: Q_R and Q ;
Get the original noise set: $D = f(Q)$;

Step 2: Evaluate the privacy leakage risk in this specific noise pre-process
Compute the privacy leakage risk based on the sizes of real private data items and other noise data items x and y by formula 4-7: $PLR_R = plr(x, y) = \sum_{i=1}^{x}(\frac{i}{x} \times \frac{1}{C_x^i \times C_{x+y}^i};)$

Step 3: Create the noise set based on the privacy-leakage-tolerance
Get the size of other noise data items in the noise set by formula 4-12: $y_{\min} = plr^{-1}(PLT_C)$;

Generate the noise generation set by Algorithm 4-1: $D_N = PTNCA(Strategy, D, y_{\min})$;

Step 4: Execute one specific noise generation strategy based on the noise set
Under one noise generation strategy,
generate a noise N by the noise request set $Q_N = f^{-1}(D_N)$;
Inject N into Q_R to get Q_S for the service process;
Update the service request queue.

Goto **Step 2**.

Algorithm 4-2 *PTNES*: Privacy-leakage-Tolerance based Noise Enhancing Strategy

In general, the key part of *PTNES* is to build and update the noise set D_N. We present *PTNCM* and *PTNCA* to summarise the major part of this noise pre-

processing strategy. With D_N, noise obfuscation considers the customer-defined privacy-leakage-tolerance to obtain a better privacy protection on noise obfuscation. In the next section, we will illustrate that *PTNES* can improve the efficiency of noise obfuscation on privacy protection with similar effectiveness by simulation.

4.5 Simulation and Evaluation

In this section, we evaluate *PTNES* by simulation. *PTNES* is a noise pre-processing strategy to enhance noise obfuscation, different from other specific noise generation and utilisation strategies. In other words, it can influence typical noise generation processes by creating the noise set as a necessary noise pre-processing. Hence, in the simulation process, the evaluation of this strategy focuses on its impact on other typical noise generation strategies.

As introduced in Chapter 3, we use SwinCloud as the cloud computing simulation environment [93]. Specifically, in this SwinCloud environment, we use some nodes as customers to send service requests with specific noise obfuscation processes. It also evaluates the cost of noise data to evaluate the efficiency of privacy protection on noise obfuscation. Other nodes are utilised as service providers to receive service requests and evaluate the effectiveness of privacy protection on noise obfuscation.

About typical noise generation strategies: the random strategy (*RNGS*) [77], the historical probability strategy (*HPNGS*) [42] and the time-series pattern strategy (*TPNGS*) [43] can be enhanced by *PTNES* in terms of noise pre-processing. We will describe these positive improvements on these strategies by *PTNES*.

In this section, we use *Noise Cost* to denote the cost of noise service requests and express the efficiency of privacy protection on noise obfuscation in this strategy. It is the percentage of noise service requests in final service request queues. In other words, it is noise injection intensity ε. It is clear that if ε is bigger, customers have to spend higher cost on noise service requests.

From the preceding sections, the setting of PLT_C as the *privacy-leakage-tolerance* is very important to the simulation process of *PTNES*. In the simulation process, we discuss it in the range of [0.05, 0.5] which means representative privacy

leakage probabilities that customers can tolerate. If it is too high, it is meaningless for privacy protection, and if it is too low, unnecessary huge cost has to be paid by customers. Besides, x is a key issue of the privacy risk evaluation and we set it in the range of [1, 5] for that n is 40. If it is too high, n has to be increased to keep noise obfuscation being functional.

As introduced before, for *RNGS*, *HPNGS* and *TPNGS*, the noise cost can be compared in two situations: with and without *PTNES* enhancing. In Figure 4-2, when *RNGS* operates, our novel *PTNES* can reduce *Noise Cost* from about 0.95 to 0.6. It is a significant improvement on the efficiency of privacy protection on noise obfuscation.

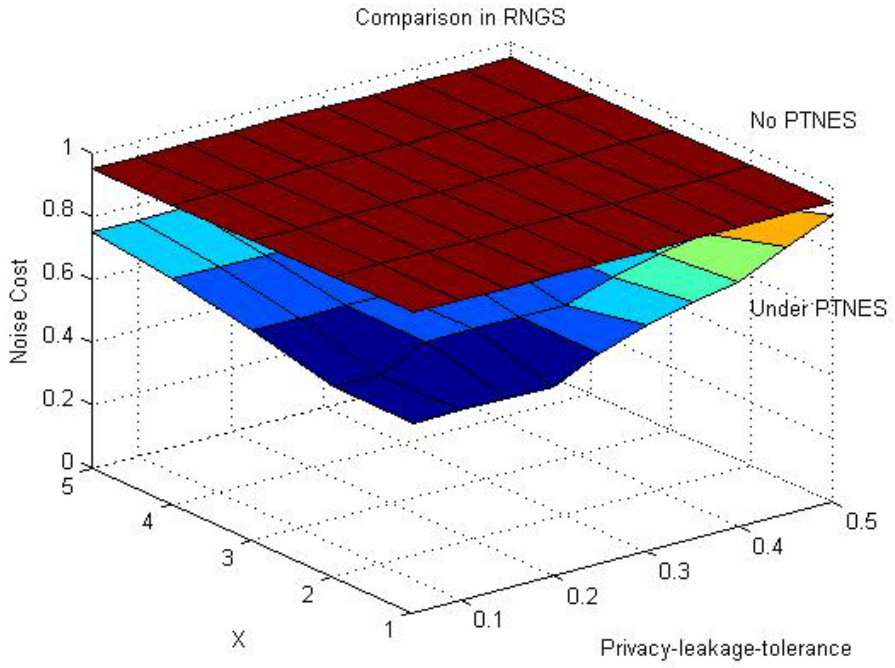


Figure 4-2 Comparison in *RNGS*

In Figure 4-3, when *HPNGS* operates, our novel *PTNES* can reduce *Noise Cost* from about 0.28 to 0.02. It is also a significant improvement on the efficiency of privacy protection on noise obfuscation.

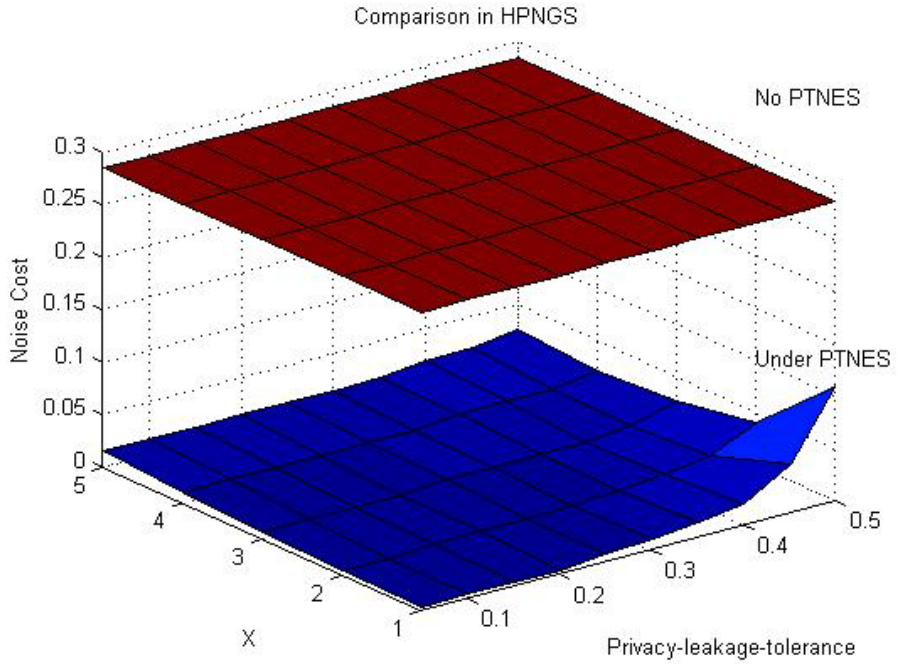


Figure 4-3 Comparison in *HPNGS*

In Figure 4-4, when *TPNGS* operates, our novel *PTNES* can reduce *Noise Cost* from about 0.029 to 0.002. It is again a significant improvement on the efficiency of privacy protection on noise obfuscation.

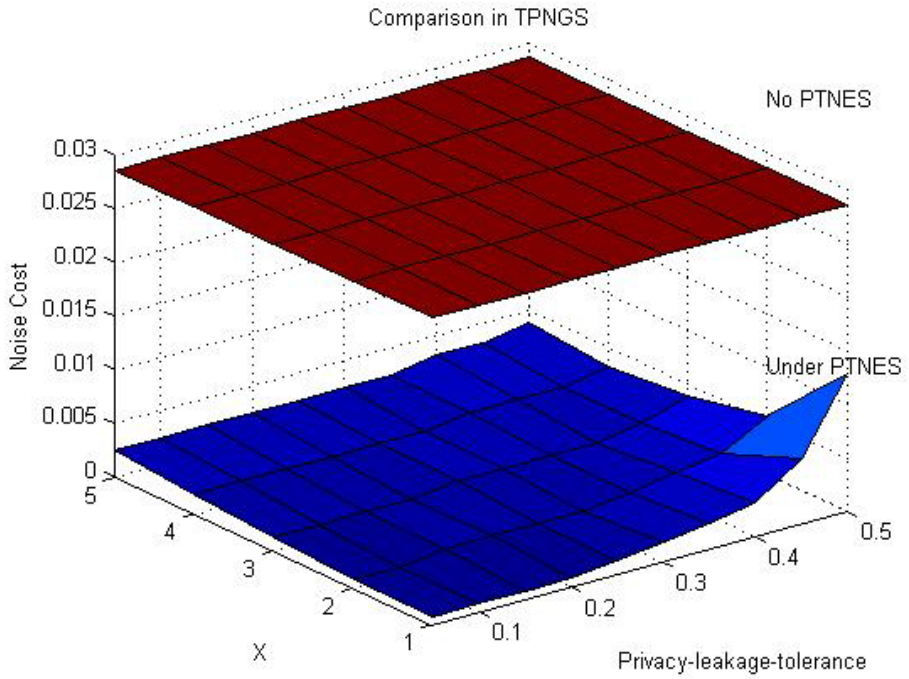


Figure 4-4 Comparison in *TPNGS*

Hence, as a noise pre-processing strategy, *PTNES* can improve the efficiency of

privacy protection significantly by reducing *Noise Cost*. In Figure 4-2, Figure 4-3 and Figure 4-4, with the increasing of *privacy-leakage-tolerance*, *Noise Cost* decreases. It is obvious that if a customer has a low privacy-leakage-tolerance setting, the cost on noise obfuscation can be much more than a high one. Besides, it is the same with another axis *x* in our figures: if customers plan to protect more data items for privacy, the cost should be more.

In summary, the simulation evaluation demonstrated that our novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) could reduce noise cost under existing noise generation strategies significantly for improving the efficiency of privacy protection on noise obfuscation in terms of noise pre-processing.

4.6 Summary

Noise pre-processing is the first main component of our novel noise obfuscation model for privacy protection in cloud computing. It focuses on customer's privacy requirements on noise obfuscation in terms of privacy protection. In this regard, the privacy-leakage-tolerance can be utilised to link noise obfuscation functions and cloud customers' privacy requirements by the noise set' creation. Therefore, in the noise pre-processing component of our novel noise obfuscation model, the noise pre-processing strategy in this chapter can be presented to enhance noise obfuscation for cloud privacy protection in terms of noise pre-processing.

In this chapter, as the main issue of the noise pre-processing component, we presented the novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) as a noise pre-processing strategy to enhance noise obfuscation. Specifically, based on the boundary probability of privacy leakage which cloud customers can accept or tolerate, the noise set can be created to guide noise generation processes and pursue better efficiency of privacy protection on noise obfuscation. Hence, we proposed this novel noise pre-processing strategy to create the noise set based on the privacy-leakage-tolerance and guide noise generation processes in this regard. The simulation experiments conducted in the SwinCloud environment demonstrated that our novel strategy (*PTNES*) is capable of improving

the efficiency of privacy protection on noise obfuscation in terms of noise pre-processing. Therefore, overall noise cost for noise obfuscation can be reduced. In brief, the noise pre-processing strategy can receive privacy requirements from cloud customers, and transfer these requirements to noise generation in terms of noise pre-processing, in one single noise obfuscation process. As a result, noise obfuscation's cost can be reduced and controlled in accordance with customers' requirements.

In future, the privacy-leakage-tolerance can be further investigated to improve the adaptation of this noise pre-processing strategy for other types of customers' requirements, such as the ratio between the privacy-leakage-tolerance and the cost on noise obfuscation. That is a trade-off for cloud customers to manage noise obfuscation by considering the effectiveness of privacy protection and the cost on noise obfuscation.

Chapter 5

Noise Generation by Time-series Pattern

As a core component of our novel noise obfuscation model for privacy protection in cloud computing, the noise generation component is the second step of this model, and is actually a critical part of the single noise obfuscation process and the whole noise obfuscation function. Briefly, it receives the noise set from the noise pre-processing component, and generates noise data effectively and efficiently to operate the single noise obfuscation process and the whole noise obfuscation function. Hence, we can propose some noise generation strategies to execute noise generation functions in this component. Besides, as the kernel function of noise obfuscation, the effectiveness of privacy protection is the crucial standard to be considered in this component. As introduced in Chapter 1 and Section 3.4, there are two different novel noise generation strategies to make sure the effectiveness of privacy protection under two different serious privacy risks: probability fluctuation and association analysis, respectively. Besides, in the pay-as-you-go cloud environments, the customers' requirements on cost-saving need noise obfuscation to consider the efficiency of privacy protection. Hence, in noise generation, our novel noise generation strategies focus on keeping privacy safe to withstand these types of privacy risks, while keeping a reasonable noise data cost in noise obfuscation.

Briefly, this chapter presents one novel noise generation strategy to withstand this privacy risk about probability fluctuation. As introduced in Section 3.4, this privacy risk expresses one main way to break noise generation for privacy attacker: using external features of service data, compared to using internal features of service

data in Chapter 6. In this strategy, we abstract fluctuations of probabilities from past service data by modelling time-series patterns, and utilise these patterns to forecast future fluctuations. Hence, noise generation processes can utilise these forecasted results to conceal these fluctuations and address this privacy risk introduced before. Based on that, our novel Time-series Pattern based Noise Generation Strategy (*TPNGS*) for privacy protection in cloud computing can be presented to operate this function.

In this chapter, Section 5.1 introduces the background of this strategy. Section 5.2 presents the time-series pattern based noise injection model to support *TPNGS*. Section 5.3 discusses the novel Time-series Pattern based Forecasting algorithm (*TPF*) for noise generation. Section 5.4 investigates the time-series pattern based noise generation. Section 5.5 proposes our novel *TPNGS*. Section 5.6 provides the simulation evaluation to illustrate *TPNGS*. Finally, Section 5.7 summaries this chapter.

This chapter is mainly based on our work presented in [43].

5.1 Background of the Strategy

In this chapter, we propose a novel Time-series Pattern based Noise Generation Strategy (*TPNGS*). In brief, this strategy focuses on fluctuations of occurrence probabilities which could jeopardise the existing noise generation strategies, such as *HPNGS* [42]. In this strategy, time-series patterns can be utilised to forecast these fluctuations and guide noise generation processes. Hence, under this strategy, noise generation can improve the effectiveness of privacy protection on noise obfuscation in cloud computing to withstand these fluctuations.

Currently, a Historical Probability based Noise Generation Strategy (*HPNGS*) has been proposed to reduce the cost of noise obfuscation in a pay-as-you-go cloud environment [42]. Compared to the conventional random noise generation [77], *HPNGS* generates noise requests based on their previous probabilities: if one request has a high occurrence probability of real service requests, it will be generated as noise requests with a low probability. Hence, all requests including noise ones and real ones could still reach about the same occurrence probabilities, but with fewer

noise requests. For the pay-as-you-go style of cloud computing, few noise requests generated mean less cost.

In reality, due to the dynamic of cloud environment, occurrence probabilities of real service requests may have some fluctuations at some time intervals of the entire time period. For the purpose of privacy protection, we need to make all these occurrence probabilities be similar at any time intervals to conceal these fluctuations. It means that customers' privacy can be protected under these fluctuations. However, the existing strategy (*HPNGS*) has not taken these fluctuations into account because it utilises past probabilities as a whole to generate noise requests without considering time intervals. In other words, *HPNGS* can reach about the same probabilities in the entire time period, but may be not the case at every time interval, which can form the entire time period, due to these probability fluctuations. As a result, final service requests, including real ones and noise ones, may have some significant probability fluctuations. Then, service providers could still be able to deduce customer private information from these fluctuations at those time intervals. This is a serious privacy risk. Besides, random noise generation [77] does not consider this privacy risk either.

To address this problem, we develop our novel Time-series Pattern based Noise Generation Strategy (*TPNGS*) for privacy protection in cloud computing. In this strategy, we analyse all past probabilities, and deduce time-series patterns by time-series segmentation. Based on these past time-series patterns, we analyse current probabilities of real requests and forecast "future" probabilities of real requests with pattern matching. At last, this strategy can generate time-series pattern based noise requests to protect customer privacy. These noise requests can make final requests to reach the goal that all occurrence probabilities of final requests are kept about the same, even at time intervals with probabilities' fluctuations.

Let us take the cloud weather report as the motivating example, again. As introduced in Section 1.2.2, customer privacy about location information in service requests can be protected in general by existing noise obfuscations. But in reality, given the privacy risk in this strategy, the customer could go to 'Sydney' in this month and 'Perth' in the next month. Hence, these probabilities of requests may have some fluctuations: 'Sydney' is high in this month and low in the next month; 'Perth' is low in this month and high in the next month. In the view of the entire service period, both occurrence probabilities may be about the same already. But the

itinerary of this customer still can be discovered by some unethical services: the person will go to 'Sydney' in this month and 'Perth' in the next month. As a result, these fluctuations are quite hard to be concealed by existing noise obfuscations. To address this, the updated goal of privacy protection in this strategy is to keep occurrence probabilities of final requests to be about the same at every time interval, instead of only in the entire time period. We can forecast these fluctuations by time-series patterns and generate noise service requests to achieve this goal. That is the main motivation of our novel *TPNGS* in this chapter.

Considering the privacy risk in this chapter, time-series pattern is an effective tool to forecast 'future' occurrence probabilities based on past probabilities in the situation with probability fluctuations. We can analyse and deduce several time-series patterns from all past probabilities. Then, jointly with current occurrence probabilities, we can forecast persuasive 'future' real request probabilities to guide noise generation. And the probability fluctuations can be foreseen and addressed.

In the following sections, firstly, we detail the noise injection model to support *TPNGS*. Secondly, we present the novel Time-series Pattern based Forecasting algorithm (*TPF*) for noise obfuscation to support *TPNGS* in terms of dealing with occurrence probabilities' fluctuations. Thirdly, we present the novel *TPNGS*. Finally, some simulation evaluations can illustrate this strategy from the perspective of privacy protection.

5.2 Time-series Pattern based Noise Injection Model

In this section, we introduce the time-series pattern based noise injection model to support *TPNGS*. Our time-series pattern based noise injection model is modified from [42] to fulfil our time-series pattern idea as shown in Figure 5-1.

Denotations in this section are listed in Figure 4-1: Q_R, Q_N, Q_S and ε. And Q is a set of all service requests in this chapter: $Q = \{q_1, q_2, ..., q_i, ..., q_n\}$.

The overall working process of the model is to inject Q_N into Q_R based on ε so that we can get Q_S. The model can be described as follows. Suppose q_i is an item of Q and $P(Q_R = q_i)(t)$, $P(Q_N = q_i)(t)$ and $P(Q_S = q_i)(t)$ are probabilities of q_i in

Q_R, Q_N and Q_S at time *t* respectively.

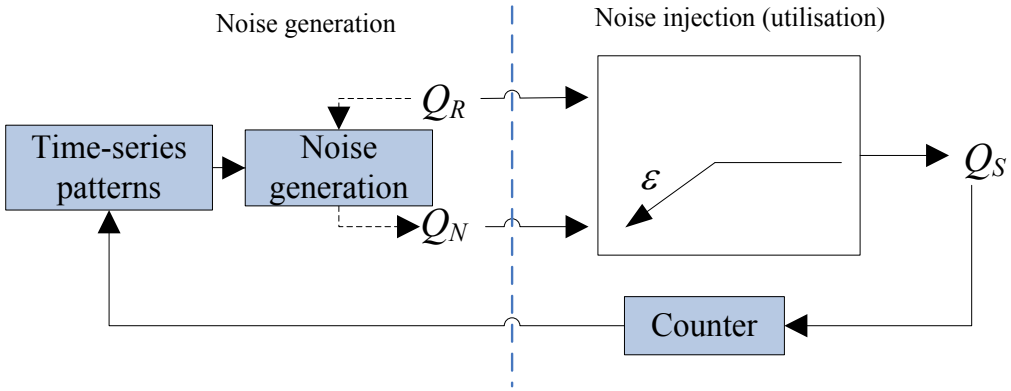


Figure 5-1 Time-series pattern based noise injection model

As introduced before, to protect customers' privacy, we need to achieve the state that for $\forall i$, all $P(Q_S = q_i)(t+1)$ are about the same. Therefore, if we forecast that $P(Q_R = q_i)(t+1)$ has a high value by this strategy, then q_i will not be taken as noise at time *t+1*, so that $P(Q_N = q_i)(t+1)$ will have a smaller value, and vice versa. This is the general process of generating noise requests based on time-series patterns of real requests.

In the next section, based on this time-series pattern based noise injection model, we can present the novel time-series pattern based forecasting algorithm for noise obfuscation to support *TPNGS*.

5.3 Time-series Pattern based Forecasting Algorithm for Noise Obfuscation

In this section, we present the novel Time-series Pattern based Forecasting algorithm (*TPF*) for noise generation which is the key part of our novel *TPNGS*. Firstly, we introduce an algorithm for Time-series Segmenting and Pattern Generation (*TSPG*). Then, based on these patterns, we introduce an algorithm for Pattern Matching and Forecasting (*PMF*). At last, to support our *TPNGS* strategy, the *TPF* algorithm is presented.

Similar to other data in time-series pattern based forecasting algorithms [30, 91, 32], occurrence probabilities have the characteristic of changing with time. That is

the common precondition of time-series pattern based algorithms. In this chapter, past occurrence probabilities are composed of various occurrence probabilities of various service requests, and each of them can be treated as an independent time-series pattern based forecasting process. Therefore, in one time-series pattern based forecasting process, we execute both *TSPG* and *PMF* algorithms to derive several forecasting results. Then, we combine these processes together and integrate these forecasting results. This is the main procedure of the novel *TPF* algorithm for noise generation.

5.3.1 *TSPG*: Time-series Segmenting and Pattern Generation Algorithm

Here we introduce the first part of time-series pattern based forecasting—the *TSPG* algorithm based on [91].

At first, *TSPG* divides past occurrence probabilities and gets some time segments. Then, it checks the validation of them and generates time-series patterns. We utilise the bottom-up and top-down approaches to move windows in time-series to make sure that the variance of one segment is close to, but not more than a pre-set parameter as a maximum boundary of variance. After that, we split the time-series queue into several time segments. Lastly, we validate and set them as patterns by a pre-set parameter: *Min_pattern_length* which means the minimum boundary of a validated pattern's length. The input of *TSPG* is the past occurrence probabilities of real requests: $P(Q_R = q_k)(t), k \in [1, n], t \in [0, T]$, and the output is a group of time segments— $Patterns[j], j \in [0, m]$. The function of the *TSPG* algorithm can be viewed as $Patterns[j], j \in [0, m] = TSPG(P(Q_R = q_k)(t), k \in [1, n], t \in [0, T])$. Besides, each validated pattern has an attribute—*nextvalue* which is the first value of the next pattern or time segment after this pattern in the whole queue. It is a key attribute for forecasting in *PMF* described next.

5.3.2 *PMF*: Pattern Matching and Forecasting Algorithm

Here, we introduce the second part of time-series pattern based forecasting—the *PMF* algorithm based on [91], too.

In brief, *PMF* utilises time-series patterns, resulted from *TSPG*, to match current

probabilities. If we find a matched pattern, its forecasting attribute—*nextvalue* can play a key role to forecast 'future' probabilities. *Min(abs(Patterns.mean-CP.mean))* denotes a function which returns one pattern with a minimum absolute difference of means between it and *CP* which denotes the current probabilities queue. This is the main part of *PMF* to find out the suitable pattern to match the current probabilities queue, and we utilise the means of probabilities at patterns to evaluate this. Hence, the function of this algorithm is $MP, FR = PMF(Patterns[j], j \in [0, m], CP)$: i.e., one input is the patterns we have got: $Patterns[j], j \in [0, m]$, another input is the current probabilities queue: *CP*; one output is the matched pattern: *MP*, another output is the forecasting result: *FR*. Our forecasting result *FR* is a probability which denotes the future occurrence possibility of one real service request, and it is decided by the matched pattern *MP*.

In the real process of pattern matching, the *PMF* algorithm takes the mean of current probabilities queue *CP* as the default value. If we cannot find out a suitable pattern, the mean is used as the forecasting result *FR* to guide noise generation.

5.3.3 *TPF*: Time-series Pattern based Forecasting Algorithm

Here, we present the novel *TPF* algorithm for noise generation. In Algorithm 5-1, we detail the novel Time-series Pattern based Forecasting algorithm (*TPF*) for noise generation based on the *TSPG* algorithm which can be applied as a function named *TSPG()*, and the *PMF* algorithm which can be applied as a function named *PMF()*. We operate them for various probabilities of service requests and derive various forecasting results. After that, we need to normalise these forecasting results. It is apparent that for a certain time interval, the sum of probabilities of all service requests is 1. Besides, we denote *L* for the length of current probabilities queue. In this chapter, we set it to be equal to *Min_pattern_length* for the balance between effectiveness and efficiency of forecasting.

Title: Time-series pattern based forecasting algorithm for noise generation
Input: All past occurrence probabilities $P(Q_R = q_k)(t), t \in [0,T], k \in [1,n]$
The length of current probabilities queue L
Output: One group of future probabilities $P(Q_R = q_k)(T+1), k \in [1,n]$

For (i = 1; i ≤ n ; i++) //Execute forecasting process
    Patterns[i] = *TSPG*($P(Q_R = q_i)(t), t \in [0,T]$);
    FR[i][t] = *PMF*(*Patterns[i]*, $P(Q_R = q_i)(t), t \in [T-L,T]$);
End

For (i = 1; i ≤ n; i++) //Sum forecasted results
    FRS = *FRS* + *FR[i]*;
End

For (i = 1; i ≤ n; i++) //Normalise forecasted results
    $P(Q_R = q_i)(T+1)$ = *FR[i]* / *FRS*;
End

Algorithm 5-1 *TPF*: Time-series Pattern based Forecasting algorithm

In the *TPF* algorithm, we first utilise the Time-series Segmenting and Pattern Generation algorithm (*TSPG*) and the Pattern Matching and Forecasting algorithm (*PMF*) to execute time-series patterns based forecasting processes which are divided from an entire probabilities forecasting process. In each single process for each service request, we deduce time-series patterns by segmenting on past probabilities, and utilise these patterns to match the current probabilities to forecast probabilities at the next time interval. Then, we combine these results from these processes. At last, we normalise them to integrate one group of 'future' probabilities of real requests for noise generation. Compared to [91], the *TPF* algorithm puts a lot of efforts into the utilisation of forecasting results and their normalisation for noise generation which are novel.

5.4 Time-series Pattern based Noise Generation

In this section, we introduce the two key issues of the noise generation strategy—noise generation probabilities and noise injection intensity. In the process of noise generation, noise generation probabilities determine which kinds of noise requests should be generated and the noise injection intensity decides how many noise requests should be generated.

5.4.1 Noise Generation Probabilities

Based on [42], we present the noise generation probabilities in this strategy. We add parameter time *t* to denote the time attribute in noise generation processes. Hence, we have noise generation probabilities in *TPNGS*:

$$\forall i, P(Q_N = q_i)(t) = \frac{M(t) - P(Q_R = q_i)(t)}{n \times M(t) - 1} \quad \textbf{Formula 5-1}$$

In formula 5-1, $M(t)$ is that for every *i*, the largest $P(Q_R = q_i)(t)$ at time *t*.

$$M(t) = MAX\{\forall i, P(Q_R = q_i)(t)\} \quad \textbf{Formula 5-2}$$

Based on formulas 5-1 and 5-2, we can get $P(Q_R = q_i)(t)$ which is an important part of noise generation probabilities.

$$\forall i, P(Q_R = q_i)(t + \Delta t) = TPFA(P(Q_R = q_i)(t'), t' \in [1, t]) \quad \textbf{Formula 5-3}$$

In formula 5-3, $TPFA()$ denotes the function of the *TPF* algorithm in Algorithm 5-1. Hence, formula 5-3 is a key part of *TPNGS*: we use past requests' probabilities to forecast future requests' probabilities to aid noise generation by time-series patterns. We set $\Delta t = 1$. Then, we have formula 5-4 below:

$$\forall i, P(Q_R = q_i)(t) = TPFA(P(Q_R = q_i)(t'), t' \in [1, t-1]) \quad \textbf{Formula 5-4}$$

Combining formulas 5-1, 5-2 and 5-4, we can get the final noise generation probabilities in *TPNGS* by formula 5-5:

$$\forall i, P(Q_N = q_i)(t) = \frac{MAX\{\forall j, TPFA(P(Q_R = q_j)(t'), t' \in [1, t-1])\} - TPFA(P(Q_R = q_i)(t'), t' \in [1, t-1])}{n \times MAX\{\forall j, TPFA(P(Q_R = q_j)(t'), t' \in [1, t-1])\} - 1}$$
$$= \frac{MAX\{\forall j, TPFA(j, t-1)\} - TPFA(i, t-1)}{n \times MAX\{\forall j, TPFA(j, t-1)\} - 1} \quad \textbf{Formula 5-5}$$

5.4.2 Noise Injection Intensity

To reach the goal of privacy protection discussed before, we try to get final "indistinguishable" probabilities:

$$\forall i, \forall t, P(Q_S = q_i)(t) = \frac{1}{n} \quad \textbf{Formula 5-6}$$

Based on the noise injection model in Section 5.3, we have the following probabilities:

$$\forall i, P(Q_S = q_i)(t) = (1-\varepsilon)P(Q_R = q_i)(t) + \varepsilon P(Q_N = q_i)(t) \quad \textbf{Formula 5-7}$$

Combining formulas 5-6 and 5-7, we can derive noise injection intensity ε to reach the goal for privacy protection:

$$\varepsilon(t) = 1 - \frac{1}{n \times M(t)} \quad \textbf{Formula 5-8}$$

To realise formula 5-8, we have formula 5-9.

$$\varepsilon(t) = 1 - \frac{1}{n \times MAX\{TPFA(\forall i, P(Q_R = q_i)(t'), t' \in [1, t-1])\}} \quad \textbf{Formula 5-9}$$

Formulas 5-5 and 5-9 enables the whole strategy to reach its goal, i.e. formula 5-6.

Compared to existing noise generation strategies, like *HPNGS* or random generation, *TPNGS* enhances the outcome of privacy protection from $\forall i, P(Q_S = q_i) = 1/n$ to formula 5-6. As a result, it can address the serious privacy risk identified before. Besides, it is clear that the goal of *TPNGS*, i.e. realisation of formula 5-6, is a sufficient condition of the goal of existing strategies: $\forall i, P(Q_S = q_i) = 1/n$. Hence, if the occurrence probabilities are about the same at every time interval, these probabilities will be about the same in the overall time period.

5.5 Time-series Pattern based Noise Generation Strategy

In this section, based on the former sections, we present our novel Time-series Pattern based Noise Generation Strategy—*TPNGS*.

In Algorithm 5-2, we can find out that the major improvement of *TPNGS* is to use

$\forall i, TPFA(P(Q_R = q_i)(t'), t' \in [1, t-1])$ as forecasting results in Step 1. As stated earlier, the *TPF* algorithm is the time-series pattern forecasting algorithm for noise generation which utilises time-series patterns to summarise past probabilities and forecast 'future' probabilities. In this strategy, we use the *TPF* algorithm in the first step to forecast, and utilise the results of the *TPF* algorithm in later steps (Step 2 and Step 3)—computing noise generation probabilities and noise injection intensity. It is obvious that this strategy performs better in the privacy protection situation with fluctuations of probabilities than existing strategies, like *HPNGS* or random generation. In Step 4, noise injection processes have been executed. Besides, under an extreme condition without fluctuations, it is clear that *TPNGS* and *HPNGS* could perform similarly in noise generation, for there is no need to forecast.

Title: Time-series pattern based noise generation strategy
Input: Q_R is the queue of real service requests
Output: Q_S is the queue of final service requests

Step 1: Collect past probabilities and utilise *TPF* algorithm

Collect and record all occurrence probabilities in past time:

$$\forall i, P(Q_R = q_i)(t'), t' \in [1, t-1]$$

Utilise *TPF* algorithm to forecast by formula 5-10:

$$\forall i, P(Q_R = q_i)(t) = TPFA(P(Q_R = q_i)(t'), t' \in [1, t-1])$$

Step 2: Compute noise generation probabilities

Generate noise generation probabilities by formula 5-5:

$$\forall i, P(Q_N = q_i)(t) = \frac{MAX\{\forall j, TPFA(j, t-1)\} - TPFA(i, t-1)}{n \times MAX\{\forall j, TPFA(j, t-1)\} - 1}$$

Step 3: Compute noise injection intensity

Obtain noise injection intensity by formula 5-9:

$$\varepsilon(t) = 1 - \frac{1}{n \times MAX\{\forall i, TPFA(P(Q_R = q_i)(t'), t' \in [1, t-1])\}}$$

So, we have the queue of noise requests:

$$Q_N[P(Q_N = q_i), \varepsilon]$$

Step 4: Execute noise injection process

Generate a noise N by $Q_N[P(Q_N = q_i), \varepsilon]$;
Inject N into Q_R on the probability of ε to get Q_S ;
Update past probabilities $\forall i, P(Q_R = q_i)(t'), t' \in [1, t]$.

Algorithm 5-2 *TPNGS*: Time-series Pattern based Noise Generation Strategy

In the next section, we will evaluate *TPNGS* with other strategies.

5.6 Simulation and Evaluation

In this section, we evaluate *TPNGS* by simulation. Generally speaking, *TPNGS* focuses on noise generation functions to deal with fluctuations of occurrence probability by time-series patterns and improve the effectiveness of privacy protection on noise obfuscation.

Just like other strategies in this thesis, we use SwinCloud as the cloud computing simulation environment [93]. And the aim of this simulation is to simulate *TPNGS* in order to demonstrate that *TPNGS* can improve the effectiveness of privacy protection significantly, compared to other existing representative noise generation strategies, like *HPNGS* and random generation.

Besides, how to deal with distributed denial-of-service (*DDoS*) attacks has become a very serious issue concerned by servers. In this chapter even in this whole thesis, we omit the possibility of our noise being viewed as *DDoS* attacks. In fact, the number of our noise is much less than a common *DDoS* attack which normally has millions of requests [26].

The simulation process is to compute and compare the privacy protection effectiveness of *TPNGS* with that of *HPNGS*. In this process, we choose *HPNGS* to compare with *TPNGS*. In the end of this section, we demonstrate the comparison between *HPNGS*, *TPNGS* and random generation. Before the simulation, we generate a service queue as the real service queue from a set with a size of 10 randomly.

We set a function: $EPP(Strategy,t) = VAR(Strategy,t)$ to express the effectiveness of privacy protection to compare two strategies. As discussed before, the variance of these probabilities of final requests is a suitable tool to evaluate the effectiveness of privacy protection. $VAR(Strategy,t)$ means that the variance of all occurrence probabilities of requests in Q_S is under *Strategy* at time *t*. A low variance of all probabilities denotes that all occurrence probabilities of final requests are about the same, and malicious service providers cannot find out real ones as addressed before. Therefore, the less $EPP(Strategy,t)$, the better effectiveness of privacy protection that can be achieved.

In the worst case for *TPNGS* executing, i.e. one request is one pattern, pattern generation is pre-computing. Hence, in noise generation processes, *TPNGS* only

needs to traverse all patterns or requests to get the matched one, and this time cost would not influence noise generation processes significantly, compared to other existing typical strategies.

In this section, we derive $EPP(HPNGS,t)$ and $EPP(TPNGS,t)$ which denote the effectiveness of privacy protection with these two noise generation strategies at time t, respectively. They are depicted in Figure 5-2. They change by t from 0 to 5000. In Figure 5-2, the horizontal coordinate is time t. The vertical coordinate is *EPP* reflecting the privacy protection effectiveness. If *EPP* is lower, the privacy protection effectiveness is better. We can find out that with time t passing, both $EPP(HPNGS,t)$ and $EPP(TPNGS,t)$ keep a similar pattern of fluctuating in specific zones. $EPP(HPNGS,t)$ fluctuates mainly between 0.00004 and 0.0001 while $EPP(TPNGS,t)$ fluctuates between 0.00001 and 0.00004. Therefore, in general, $EPP(TPNGS,t)$ is about 1/3 of $EPP(HPNGS,t)$ from the figure. Therefore, we can conclude that our novel *TPNGS* significantly improves the effectiveness of privacy protection than existing *HPNGS*.

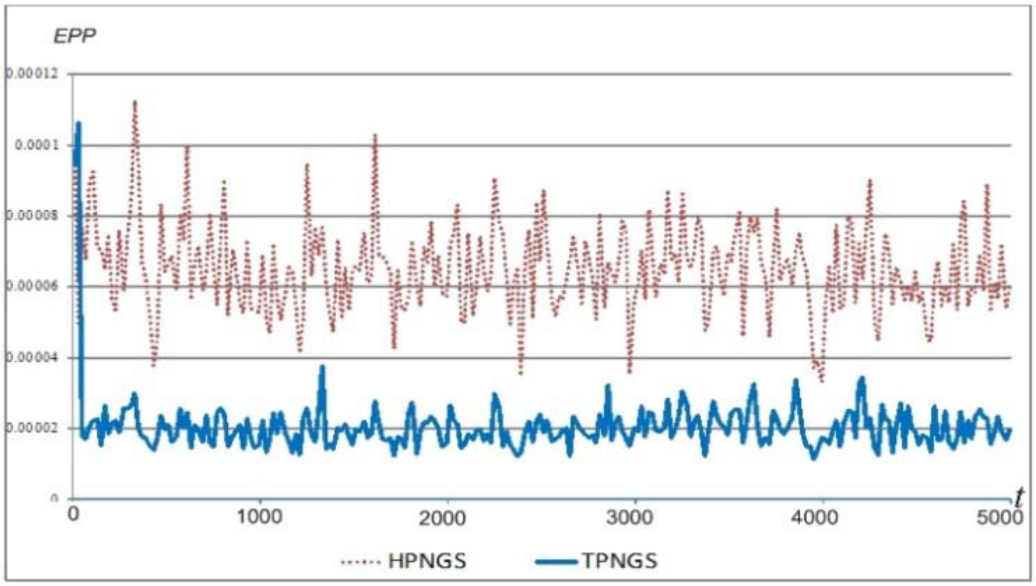


Figure 5-2 Comparison between *HPNGS* and *TPNGS*

Besides, in Figure 5-3, we can find out that in the whole simulation process, noise injection intensities of *TPNGS* are smaller than those of *HPNGS*. They fluctuate in the levels of 0.45 and 0.7, respectively. As introduced before, noise injection intensity is the probability of noise requests in final requests. Due to the pay-as-you-go style in cloud computing, the number of noise requests means the

cost of noise requests. Hence, *TPNGS* can save the cost of noise generation than that of existing *HPNGS* and improve the efficiency of privacy protection on noise obfuscation. Besides, *TPNGS* uses small sliding windows to analyse time-series data, not like *HPNGS* which uses whole queues to be a sliding window. For one specific time interval, noise obfuscation only considers and obfuscates a piece of data, not the entire one. That is why *TPNGS* could get lower cost on noise generation than *HPNGS*.

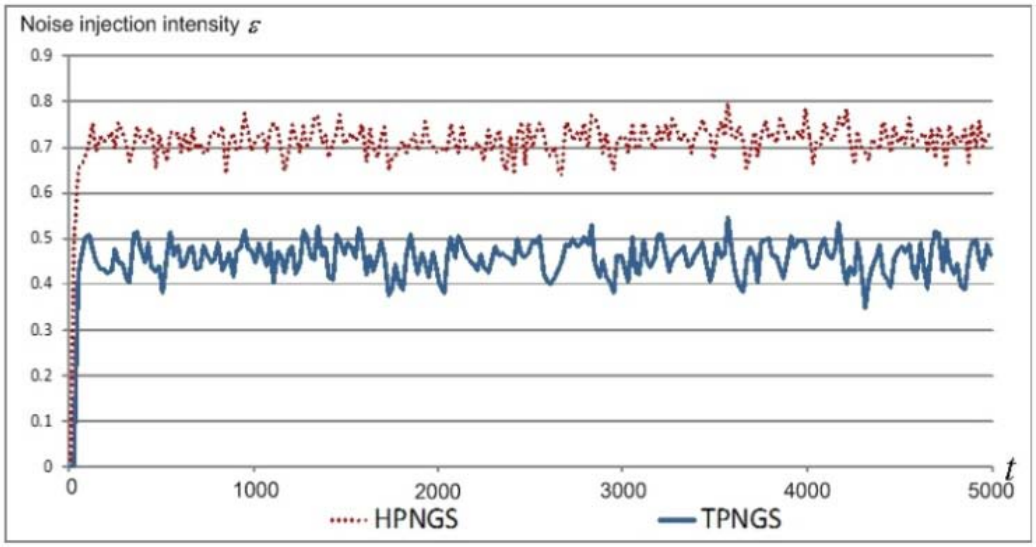


Figure 5-3 Comparison on noise injection intensity

In summary, our novel Time-series Pattern based Noise Generation Strategy (*TPNGS*) can make a significant improvement on the effectiveness of privacy protection with a decreased noise service cost in comparison to the Historical Probability based Noise Generation Strategy (*HPNGS*), in terms of the probability fluctuation privacy risk.

About the random noise generation [77], the effectiveness and efficiency of privacy protection have been discussed in [42]. Hence, it is obvious that *TPNGS* can improve both effectiveness and efficiency of privacy protection from *HPNGS* which mainly improve the efficiency of privacy protection from the random noise generation. Therefore, *TPNGS* improves the effectiveness and efficiency of privacy protection from exiting representative noise generation strategies, in terms of the probability fluctuation privacy risk.

5.7 Summary

In this chapter, we investigated the noise generation component in terms of the probability fluctuation privacy risk. As the core function of noise obfuscation, noise generation can be promoted from the perspective of various privacy risks. In this chapter, we focused on one type of them— the probability fluctuation privacy risk. In this regard, we presented our novel Time-series Pattern based Noise Generation Strategy (*TPNGS*) to withstand this privacy risk and improve the effectiveness of privacy protection on noise obfuscation in cloud computing. In this strategy, we can utilise time-series patterns in past occurrence probabilities to forecast occurrence probabilities and fluctuations. Hence, these fluctuations of occurrence probabilities can be concealed by these novel noise service requests generated by *TPNGS*. Briefly, *TPNGS* can improve the effectiveness of privacy protection on noise obfuscation in cloud computing under the probability fluctuation privacy risk, and keep the cost-saving feature in pay-as-you-go cloud environments. And based on the simulation experiments, we demonstrated that our novel strategy could significantly improve the effectiveness of privacy protection on noise obfuscation to withstand the probability fluctuation privacy risk.

This chapter presents one aspect of the thesis about noise generation which focuses on how to improve the effectiveness of privacy protection to withstand the probability fluctuation privacy risk, compared to the next chapter—how to improve the effectiveness of privacy protection to withstand the association analysis privacy risk. In other words, as discussed in Section 3.4, these privacy risks represent two main ways to break noise generation for privacy attackers respectively: using external features of service data, such as time-series in this chapter and using internal features of service data, such as association probability in the next chapter. Besides, the noise pre-processing component introduced in Chapter 4 can provide a noise set to the noise generation strategy in this chapter as a pre-process discussed in Chapter 3.

In the next chapter, we will discuss noise generation in another part—to address the association analysis privacy risk. Hence, the noise generation component in our novel noise obfuscation model can be completed and executed to fulfil this key function of the single noise obfuscation process and the whole noise obfuscation

function for cloud privacy protection.

In future, based on the exiting work, we will investigate how to improve the strategy for time-series patterns based forecasting to conceal private information, such as dynamic time-series patterns to replace the static ones used in this chapter and pursue more efficient noise generation.

Chapter 6

Noise Generation by Association Probability

After Chapter 5, i.e. noise generation for single service by time-series pattern, another kind of noise generation is investigated by this chapter. That is the noise generation for single service by association probability. Specifically, as a key step in our novel noise obfuscation model, noise generation is to generate noise data to conceal real private information. Some noise generation strategies can abstract these functions and be utilised for cloud privacy protection to withstand privacy risks. As introduced in Chapter 3, this chapter focuses on one kind of privacy risk—association analysis, and presents one novel noise generation strategy to deal with it. As discussed in Section 3.4, this privacy risk expresses another main way to break noise generation for privacy attacker: using internal features of service data, compared to using external features of service data in Chapter 5. Association probabilities generally express internal mutual relations among service data, such as dependency among service requests, and may attract some privacy attackers' interests. For instance, the dependency among service requests can express unique behaviour patterns or identities as customer privacy. Hence, some customers' privacy can be revealed by malicious service providers through these association probabilities. That is the association analysis privacy risk we have to face in this chapter. As introduced in Chapter 3, this chapter works on how to improve the effectiveness of privacy protection on noise obfuscation to withstand the association analysis risk.

In this chapter, we present a novel noise generation strategy based on past

association probabilities to pursue about the same association probabilities of various service data, such as service requests. That is our novel Association Probability based Noise Generation Strategy (*APNGS*) for privacy protection in cloud computing. To conceal these association probabilities of service requests, *APNGS* generates noise requests to pursue about the same association probabilities of final service requests which are made up of customers' real service requests and these noise service requests. Besides, to analyse these association probabilities, a novel association probability model is necessary to be presented as the core of this chapter, which will be introduced in the following sections.

This chapter is organised as follows. Section 6.1 discusses the background of this strategy. Section 6.2 presents the association probability based noise injection model to support our novel *APNGS*. Section 6.3 presents the novel association probability model for noise generation as a key part of our novel *APNGS*. Section 6.4 discusses the association probability based noise generation processes. Based on these previous sections, Section 6.5 proposes our novel Association Probability based Noise Generation Strategy (*APNGS*) for privacy protection in cloud computing. Section 6.6 addresses the simulation and evaluation about *APNGS* to demonstrate that this strategy can improve the effectiveness of privacy protection significantly to withstand the association probability risk. Finally, Section 6.7 summarises this chapter.

This chapter is mainly based on our work presented in [39].

6.1 Background of the Strategy

Currently, noise obfuscation primarily utilises noise service data to conceal occurrence probabilities of real service data. For example, when occurrence probabilities of final requests are about the same, service providers cannot distinguish which ones are real ones with high confidence, except having other extra information sources. In other words, the goal of existing noise obfuscation on privacy protection is that the variance of all occurrence probabilities is as small as possible.

But in reality, privacy is of different variety. In cloud environments, there could

be various kinds of sensitive information which can be deduced from service data as customer privacy except occurrence probabilities, such as association rules among 'real' service requests. If two requests are associated by association rules: after one request sent by one customer, then the other has a high probability to be sent sequentially. It could be a distinctive behaviour pattern of this customer. When malicious service providers find these rules, customers' behaviour patterns or their identities can be revealed as privacy leakage. Hence, it is a serious privacy risk.

Let us take the cloud weather service as the motivating example again. One customer, who often travels to some cities in Australia, like 'Sydney', 'Melbourne' and 'Brisbane', and checks the weather forecasting report regularly from a weather service in cloud environments before departure. The frequent appearance of service requests about the weather report for 'Sydney' can reveal the privacy that the customer usually goes to 'Sydney'. But if a system aids the customer to inject other requests like 'Perth' or 'Darwin' into the 'Sydney' queue, the service provider cannot distinguish which ones are real and which ones are 'noise'. Hence, privacy can be protected by noise obfuscation in general. Considering the privacy risk in this chapter, the customer may go to 'Sydney' and 'Melbourne' sequentially as a travel routine which could be viewed as customer privacy. Consequently, these association probabilities of requests may express this routine as private information: 'Sydney' and 'Melbourne' could be sent together with a high association probability in real service requests. And from the perspective of occurrence probabilities, their occurrence probabilities may be about the same, and privacy has been protected already by existing noise generation strategies. But the travel routine of this customer can still be discovered by some unethical service providers: the person has a behaviour pattern that he will go to 'Sydney' and 'Melbourne' sequentially. This information may reveal the customer's identity. However, existing representative noise obfuscations do not consider this issue. To address this, the updated goal of noise obfuscation in this chapter is to keep association probabilities of final service requests about the same, instead of occurrence probabilities only. That is the main motivation of this chapter.

To some extents, it is a challenge for noise generation to enrich its intension in privacy protection. For 'malicious' service providers, to get this private information in this serious risk addressed above, association probabilities among service requests

are necessary for association rules mining. In other words, concealing association probabilities is another kind of noise obfuscation goal we considered in this chapter. Similar to the occurrence probabilities goal mentioned before, the main goal of noise obfuscation in this chapter is that the variance of all association probabilities among service requests is as small as possible. Hence, to address this privacy risk introduced before, we need to analyse association probabilities of past real service requests, and generate noise service requests which can conceal association rules by making association probabilities be about the same. These novel noise service requests can protect customers' privacy in this chapter under the association probability privacy risk.

In this chapter, we present our novel Association Probability based Noise Generation Strategy (*APNGS*) for privacy protection in cloud computing. In this strategy, we analyse all historical real service requests, and induce association probabilities from these service data. Based on these association probabilities, we analyse current association probabilities of real requests and generate association probability based noise requests to protect customers' privacy. These noise requests can reach the goal that association probabilities of final service requests can be kept about the same. Therefore, the effectiveness of privacy protection on noise obfuscation can be improved from the perspective of association probability. In the following sections, we will introduce this strategy in detail step by step.

6.2 Association Probability based Noise Injection Model

In this section, we introduce the association probability based noise injection model to support *APNGS*. This noise injection model is based on other existing representative noise injection models [77, 42, 43] with modifications to support *APNGS* as depicted in Figure 6-1.

Denotations in this section were listed in Figure 4-1: Q_R, Q_N, Q_S and ε. And Q is a set of all service requests in this chapter: $Q = \{q_1, q_2, ..., q_i, ..., q_n\}$.

'Association probabilities': they are the basis of our noise generation strategy and guide noise generation processes. That is the main improvement of this model.

'Noise generation': its function is to generate Q_N. We use 'association

probabilities' and 'counter' to compute noise generation probabilities in *APNGS*.

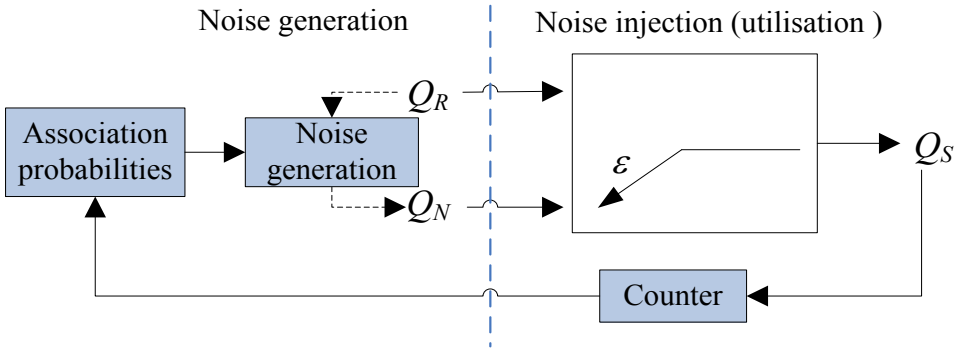


Figure 6-1 Association probability based noise injection model

The overall working process of the model is to inject Q_N into Q_R based on ε, and Q_S is the result. The model can be described as follows: the customer generates a real service request queue Q_R to be sent. The noise service request queue Q_N is generated by *APNGS*. To obtain Q_S, a switch function is: the next service request in Q_S comes from Q_N on the probability of ε, and from Q_R on the probability of $1-\varepsilon$. Suppose q_i is an item of Q and $P(Q_R = q_i)(t)$, $P(Q_N = q_i)(t)$ and $P(Q_S = q_i)(t)$ are occurrence probabilities of q_i in Q_R, Q_N and Q_S at time t, respectively. That is the basis of association probabilities which will be discussed in the next section.

6.3 Association Probability Model for Noise Generation

In this section, we introduce the novel association probability model for noise generation. In brief, we plan to investigate how to obtain association probabilities from service request queues in this model.

From the association probability based noise injection model in the preceding section, the queue Q_R can decide association probabilities which are private information. Hence, to define association probabilities based on request queues, we have formula 6-1:

$$AP = f_{AP}(Q_R) \qquad \textbf{Formula 6-1}$$

In formula 6-1, *AP* denotes association probabilities, and it is an $n \times n$ matrix. Each item in this matrix *AP[i,j]* is the association probability between q_i and q_j.

Based on request queues, sliding window is the key and widely used approach to analyse information in a data stream or queue [16]. In this strategy, considering that sliding windows are not the central area for noise generation, we use a minimised and fixed sliding window to generate association probabilities. As a basic form of sliding windows, the minimised and fixed sliding window can aid to analyse data streams in terms of basic features. To obtain a suitable and common analysis in terms of noise obfuscation, we utilise the minimised and fixed sliding window in this model. Accordingly, we obtain formula 6-2 as the association probability model for noise generation:

$$AP[i,j] = \frac{Con^{i,j}(Q_R)}{t-1} \qquad \textbf{Formula 6-2}$$

In formula 6-2, $Con^{i,j}(Q_R)$ is the number of events that q_i and q_j are sent together in Q_R. Under minimised and fixed sliding windows, this event is that q_j is immediately next to q_i as a consequential relation in Q_R. And we use time length *t-1* as the denominator to normalise the formula.

In this model, sliding windows are fixed and minimised to obtain a common discussion. In some specific privacy protection situations, sliding windows can be dynamic to withstand some specific privacy risks, such as side channel knowledge on it, or particular timestamps in request queues. In other words, sliding window is the changing key in noise generation to protect privacy in cloud environments. And as a basic form, fixed and minimised sliding windows can be easily modified to match those specific noise obfuscations.

Hence, we can define association probabilities by consequential relations among service requests in request queues. For instance, according to these sliding windows, if request ‘A’ is always the next one of request ‘B’ in request queues, we can say that the consequential relation frequency between these two requests is high, and that is a high association probability which could reveal an association rule as customer private information. Besides, to concealing these association probabilities

at service side, we should consider these association probabilities at client side in advance. Accordingly, we can accumulate all past service requests and their consequential relation frequencies for deciding these association probabilities to aid noise obfuscation.

In brief, this novel association probability model for noise obfuscation is the key for *APNGS* and supports the following sections by defining association probabilities as customer privacy. Hence, based on this model, we analyse the association probability based noise generation processes to support *APNGS*.

6.4 Association Probability based Noise Generation

In this section, we discuss the two key issues in association probability based noise generation—noise generation probabilities and noise injection intensity, similar to other noise generation strategies [42, 43]. Supposed noise generation probabilities are $P(Q_N = q_i)(t), \forall i \in [1,n]$ which means that for $\forall q_i \in Q$, probabilities of Q_N being q_i at time *t*, respectively, we can generate noise requests according to these probabilities. Noise injection intensity is ε as introduced earlier.

6.4.1 Noise Generation Probabilities

In *HPNGS* [42], noise generation probabilities are:

$$\forall i, P(Q_N = q_i) = \frac{M - P(Q_R = q_i)}{n \times M - 1} \qquad \textbf{Formula 6-3}$$

From formula 6-3, in $\forall i, P(Q_R = q_i)$, the highest one is $M = MAX\{P(Q_R = q_i), \forall i\}$, which is historical and accumulative data from past Q_R in practice as depicted in Figure 6-1, just like $P(Q_R = q_i)$, and n is the number of q_i.

Besides, from Section 6.1 and [42], existing representative strategies have the same noise generation goal:

$$\forall i, P(Q_S = q_i) = 1/n \qquad \textbf{Formula 6-4}$$

In noise generation processes, we use noise generation goal to express privacy protection goal for its consistency, as discussed in Section 6.1. Hence, based on the novel association probability model defined in formula 6-2, we get the noise generation goal in *APNGS*:

$$\forall i, j, t, P(Q_S, t+1, i, j) = P[(Q_S = q_i)(t+1) \mid (Q_S = q_j)(t)] = 1/n \quad \textbf{Formula 6-5}$$

In formula 6-5, the noise generation goal is a family of conditional probabilities to express the probability of Q_S being q_i at time *t+1*, on the precondition of Q_S being q_j at previous time *t*. Besides, we have $i, j \in [1, n]$ and $t \in [1, T]$, and T is the time length of the overall process.

To realise formula 6-5, we can utilise 'new' noise generation probabilities in formula 6-6 based on formulas 6-5 and 6-3:

$$P[(Q_N = q_i)(t+1) \mid (Q_S = q_j)(t)] = \frac{M(t, j) - P(Q_R, t, i, j)}{n \times M(t, j) - 1} \quad \textbf{Formula 6-6}$$

In formula 6-6, we have two components: $P(Q_R, t, i, j)$ and $M(t, j)$. And we have formulas 6-7 and 6-8 to illustrate them:

$$P(Q_R, t+1, i, j) = P[(Q_R = q_i)(t+1) \mid (Q_S = q_j)(t)] \quad \textbf{Formula 6-7}$$

$$M(t+1, j) = MAX\{P[(Q_R = q_i)(t+1) \mid (Q_S = q_j)(t)], \forall i\} \quad \textbf{Formula 6-8}$$

In formula 6-7, $P(Q_R, t+1, i, j)$ is a family of conditional probabilities to express the probability of Q_R being q_i at time *t+1*, on the precondition of Q_S being q_j at time *t*.

In formula 6-8, $M(t+1, j)$ is the highest value, for every i, in a family of conditional probabilities to express the probability of Q_R being q_i at time *t+1*, on the precondition of Q_S being q_j at previous time *t*. It is clear that formula 6-7 is the basis of formula 6-8. Hence, we only need to focus on formula 6-7 for association probabilities among service requests introduced before.

In *APNGS*, to obtain formula 6-7, we design a process based on formula 6-2: this

is an accumulative process in noise obfuscation processes. A 3-dimension matrix $Matrix(i, j, t)$ has three parameters: t is time parameter, i is from $(Q_R = q_i)(t+1)$ which means an event that the *ith* request in the set Q will appear in Q_R at time *t+1*, and j is from $(Q_S = q_j)(t)$ which means an event that the *jth* request in the set Q appears in Q_S at time *t*. Hence, the matrix $Matrix(i, j, t)$ means all past association relations among service requests q_i and q_j at time *t*. At a specific time, a 2-dimension array *C[i][j]* can replace $Matrix(i, j, t)$. We should collect all requests from time *1* to time *t*, and obtain accumulative *C[i][j]*. Hence, $P(Q_R, t, i, j) = Matrix(i, j, t) / SUM$. *SUM* is the number of all association relations among past requests, and *SUM = n-1*. This is the implementation of association probability model in this strategy.

We will utilise this process in the next subsection to compute the noise injection intensity, too.

6.4.2 Noise Injection Intensity

According to the association probability based noise injection model defined in Section 6.2, to operate noise obfuscation, ε is a necessary parameter.

Hence, based on Section 6.2 and [42], we can get the relation among Q_S, Q_N and Q_R:

$$P(Q_S, t, i, j) = (1-\varepsilon) \times P(Q_R, t, i, j) + \varepsilon \times P(Q_N, t, i, j) \qquad \textbf{Formula 6-9}$$

There are three components in formula 6-9: $P(Q_N, t, i, j)$, $P(Q_R, t, i, j)$ and $P(Q_S, t, i, j)$. And we have formulas 6-10, 6-7 and 6-5 introduced before to illustrate them:

$$\forall i, j, t, P(Q_N, t+1, i, j) = P[(Q_N = q_i)(t+1) \mid (Q_S = q_j)(t)] \quad \textbf{Formula 6-10}$$

In formula 6-9, it is clear that the conditional probability of Q_S being q_i at time *t+1* on the precondition of Q_S being q_j at previous time *t*, is decided by the

conditional probability of Q_R being q_i at time *t+1* on the precondition of Q_S being q_j at previous time *t* , the conditional probability of Q_N being q_i at time *t+1* on the precondition of Q_S being q_j at previous time *t*, and ε .

Hence, we can get ε based on formula 6-9:

$$\varepsilon = \frac{P(Q_S,t,i,j) - P(Q_R,t,i,j)}{P(Q_N,t,i,j) - P(Q_R,t,i,j)} \qquad \textbf{Formula 6-11}$$

In formula 6-11, we have to consider time parameter *t*, and update ε to $\varepsilon(t)$. Besides, we use formulas 6-5 and 6-4 to simplify formula 6-11:

$$\varepsilon(t+1) = \frac{\frac{1}{n} - P(Q_R,t+1,i,j)}{\frac{M(t+1,j) - P(Q_R,t+1,i,j)}{n \times M(t+1,j) - 1} - P(Q_R,t+1,i,j)} \qquad \textbf{Formula 6-12}$$

At last, we get the noise injection intensity by formula 6-13:

$$\varepsilon(t+1) = 1 - \frac{1}{n \times M(t+1,j)} \qquad \textbf{Formula 6-13}$$

Based on noise generation probabilities—formula 6-6 and noise injection intensity—formula 6-13, we can investigate *APNGS* to reach formula 6-5.

Compared to existing representative strategies, *APNGS* updates the goal of noise generation from formula 6-4 to formula 6-5. It can address the serious risk identified before. In the next section, we will present *APNGS* in detail.

6.5 Association Probability based Noise Generation Strategy

In this section, we present our novel *APNGS* based on the previous sections. In Algorithm 6-1, we utilise $n \times n+1$ counters to record the matrix and the sum of association relations among requests in Step 1. From formula 6-6, we can generate noise generation probabilities by the former matrix in Step 2. About noise injection intensity, formula 6-13 can obtain it in Step 3. At last, we can get noise requests from noise generation probabilities, and utilise them by the noise injection intensity

in Step 4.

Title: **Association probability based noise generation strategy**
Input: Q_R is the queue of real service requests
Output: Q_S is the queue of final service requests

Step 1: Collect data and compute association probabilities

Initialise n*n+1 counters: $C[1][1],\ldots\ldots,C[1][n]\ldots..C[n][1],\ldots\ldots C[n][n]$, S to record numbers of associations among each service data item and the sum;

Receive service data from all past Q_R and update the correspondent counter $C[i][j]$++ and the sum S++;

Compute previous association probabilities :

$$P(Q_R,t+1,i,j)=P[(Q_R=q_i)(t+1)\,|\,(Q_S=q_j)(t)]=\frac{C[i][j]}{S}$$

Step 2: Compute noise generation probabilities

Generate noise generation probabilities from formula 6-6:

$$P[(Q_N=q_i)(t+1)\,|\,(Q_S=q_j)(t)]=\frac{M(t,j)-P(Q_R,t,i,j)}{n\times M(t,j)-1}$$

by $P(Q_R,t,i,j)$ and $M(t,j)=MAX\{P(Q_R,t,i,j),\forall i\}$;

Step 3: Compute noise injection intensity

Compute formula 6-13: $\varepsilon(t+1)=1-\frac{1}{n\times M(t,j)}$ to get the noise injection intensity;

We have noise requests queue

$$Q_N\{P[(Q_N=q_i)(t+1)\,|\,(Q_S=q_j)(t)],\varepsilon(t+1)\}\ ;$$

Step 4: Noise injection process

Generate a noise N by:

$$Q_N\{P[(Q_N=q_i)(t+1)\,|\,(Q_S=q_j)(t)],\varepsilon(t+1)\}$$

Inject N into Q_R on the probability of ε to get Q_S ;

Update $P(Q_R,t,i,j)$ with counters.

Goto **Step 2**.

Algorithm 6-1 *APNGS*: Association Probability based Noise Generation Strategy

In summary, we can find out that the major improvement between *APNGS* and existing representative noise generation strategies is the noise generation goal updating: replacing $\forall i,P(Q_S=q_i)=1/n$ by $\forall i,j,t,P(Q_S,t,i,j)=1/n$. Based on that, *APNGS* can perform better in privacy protection situations considering association probabilities than existing representative noise generation strategies, as demonstrated in the next section.

6.6 Simulation and Evaluation

In this section, we perform an experimental simulation in our cloud simulation system—SwinCloud [93] introduced in Section 3.6. The aim is to simulate *APNGS* in order to demonstrate that *APNGS* can improve the effectiveness of privacy protection significantly than existing representative noise generation strategies in terms of association probabilities, like *HPNGS* [42], *TPNGS* [43] and random generation [77]. In this section, we compare *APNGS* and *HPNGS* in the simulation process, and make further discussions on other representative strategies in result analysis, because these other strategies can be represented by *HPNGS* for their similar privacy protection goals—concealing occurrence probabilities.

Before the simulation, we generate a service queue as the real service queue from a set with a size of 10 randomly to operate two strategies and it is impossible to design a special experimental sample to facilitate *APNGS* on purpose. We use a function: $Var\_Ass(Strategy,t)$ to express the main effectiveness of privacy protection on noise obfuscation. As discussed before, the variance of association probabilities among service requests is a suitable standard to evaluate the effectiveness of privacy protection on noise obfuscation from the perspective of association probability, and $Var\_Ass(Strategy,t)$ means that the variance of association probabilities in Q_S under *Strategy* protected at time *t*. Hence, it is obvious that the less $Var\_Ass(Strategy,t)$, the better effectiveness of privacy protection on noise obfuscation in terms of association probability with *Strategy* operating at time *t*. Besides, we also use $Var(Strategy,t)$ to denote the variance of occurrence probabilities in Q_S with *Strategy* operating at time *t*. It denotes the effectiveness of privacy protection on noise obfuscation in terms of occurrence probability as another aspect of the effectiveness of privacy protection.

Based on the simulation process, we derive $Var\_Ass(HPNGS,t)$ and $Var\_Ass(APNGS,t)$ which denote the main effectiveness of privacy protection under two strategies in this chapter, respectively. They are depicted in Figure 6-2.

In Figure 6-2, the horizontal coordinate is time *t*, and *t* has a range of [1, 6001]; the vertical coordinate is the main effectiveness of privacy protection Var_Ass in

this chapter. As introduced before, if Var_Ass is lower, the effectiveness of privacy protection is better, hence customers privacy is better kept. We can find out the overall trend being: with time passing, $Var\_Ass(HPNGS,t)$ and $Var\_Ass(APNGS,t)$ both keep on decreasing. Obviously, $Var\_Ass(HPNGS,t)$ is always higher than $Var\_Ass(APNGS,t)$. In Figure 6-2, $Var\_Ass(APNGS,t)$ is about ¼ to ½ of $Var\_Ass(HPNGS,t)$. Hence, we have the conclusion: *APNGS* achieves a significant improvement on the effectiveness of privacy protection on noise obfuscation over *HPNGS*, from the perspective of association probability.

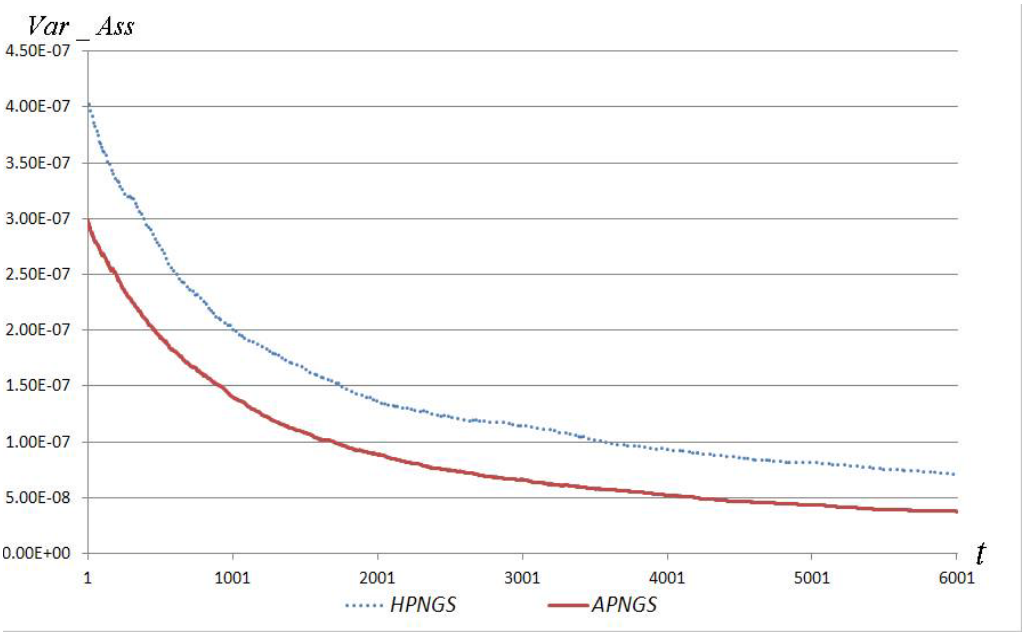


Figure 6-2 Effectiveness comparison on association probability between *HPNGS* and *APNGS*

Besides, we need the variance of occurrence probabilities: $Var(Strategy,t)$ to evaluate another aspect of privacy protection effectiveness. In Figure 6-3, we find out that $Var(APNGS,t)$ can keep within a low level of about 2.00e-05, and is lower than $Var(HPNGS,t)$ which has a level of about 3.00e-05. Therefore, we have a conclusion that *APNGS* has a better effectiveness of privacy protection on noise obfuscation in terms of occurrence probability than *HPNGS*, though *HPNGS* is already a suitable one [42]. That is because association probabilities consider more details of data queues than occurrence probabilities to keep the service requests balance.

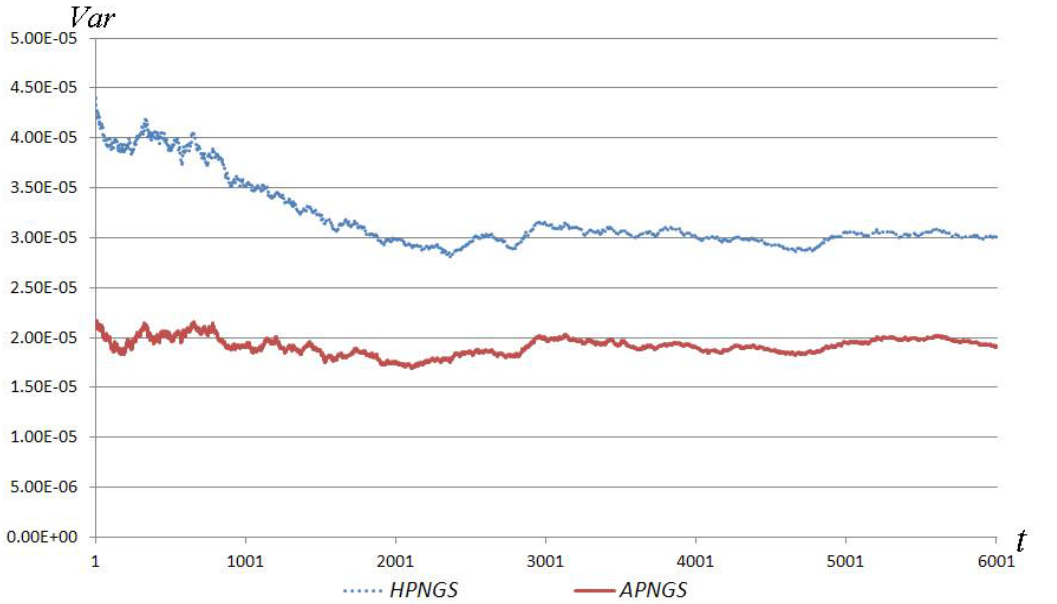


Figure 6-3 Effectiveness comparison on occurrence probability between *HPNGS* and *APNGS*

At last, the cost of privacy protection on noise obfuscation should also be considered. In cloud, noise service requests consume resources and customers need to pay for resources consumed. We can use the noise injection intensity to evaluate the cost on noise.

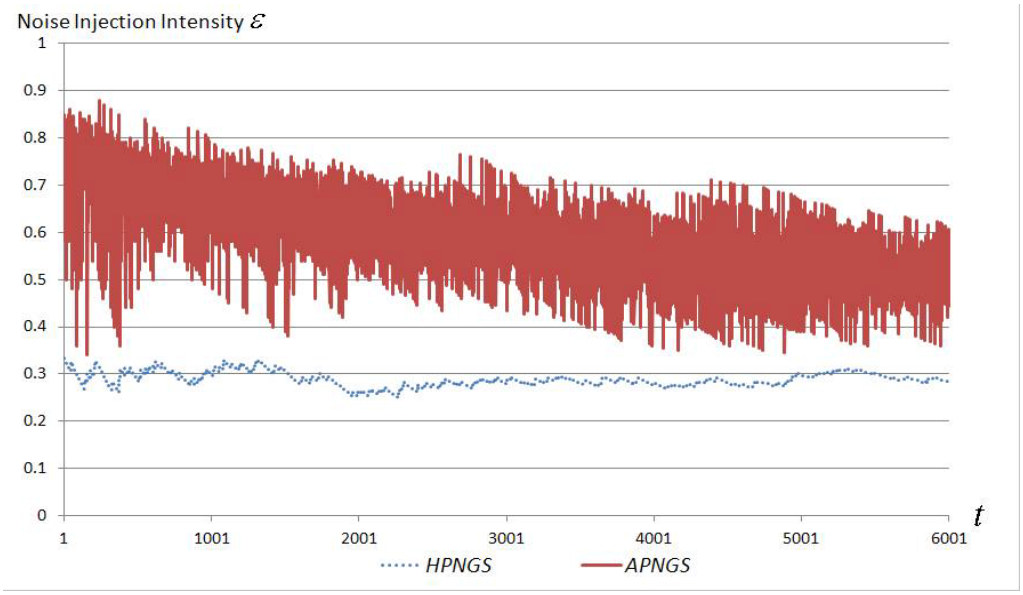


Figure 6-4 Comparison on noise injection intensity between *HPNGS* and *APNGS*

In Figure 6-4, we can find out that ε of *APNGS* is about 1.5 times more than

ε of *HPNGS* at the beginning, and then the disparity decreases to about 1 time with time passing. It means that *APNGS* costs more than *HPNGS*. In other words, to obtain a better effectiveness of privacy protection in cloud computing, customers need to pay more, in terms of association probability or occurrence probability. It is a trade-off depending on customers' demands.

About other representative noise generation strategies, such as random generation [77] and *TPNGS* [43], *APNGS* also performs well: *HPNGS* has already improved the efficiency of privacy protection on noise obfuscation from random generation with similar effectiveness. *TPNGS* focuses on fluctuations in occurrence probabilities and addresses them with time-series pattern forecasting, and association probabilities are not incorporated.

In summary, *APNGS* can significantly improve the effectiveness of privacy protection on noise obfuscation in terms of association probability over existing representative strategies, with a good effectiveness of privacy protection on noise obfuscation in terms of occurrence probability, at a reasonable extra cost.

6.7 Summary

In this chapter, we presented our novel Association Probability based Noise Generation Strategy (*APNGS*) for privacy protection in cloud computing. Based on existing typical noise generation strategies, association probabilities can be utilised in noise generation processes to conceal them as private information. For instance, these association probabilities could be the dependencies among service requests in real service request queues, which can express the unique behaviour patterns or identities as customer privacy. As a result, these association relations or rules should be concealed and protected as customer privacy by this strategy in cloud computing. In this strategy, we presented the novel association probability model for noise generation to analyse and abstract association probabilities. After that, noise generation can be executed based on this model and pursue the goal of noise obfuscation in this aspect of noise generation—similar association probabilities of final service data. Based on the simulation experiments, we demonstrated that our novel strategy could significantly improve the effectiveness of privacy protection on

noise obfuscation to withstand the association analysis privacy risk.

In Chapter 5 and this chapter, we introduced our work on the noise generation component of our novel noise obfuscation model. In general, we improved existing noise generations to withstand two serious privacy risks—probability fluctuation and association probability. As introduced in Section 3.4, these two privacy risks can represent two main ways to break noise generation for privacy attackers: external and internal features of service data. Accordingly, we presented two novel representative noise generation strategies respectively. Hence, under these noise generation strategies, the noise generation component can be executed as the key part of the single noise obfuscation process and the whole noise obfuscation function. Besides, the noise pre-processing component introduced in Chapter 4 can provide the noise set to the noise generation strategies in this chapter and Chapter 5 as a pre-process discussed in Chapter 3. Furthermore, our novel noise obfuscation model presented in Chapter 3 can be operated by the single noise obfuscation process in the single service scenario, and in the multiple services scenario, this model can be enhanced by the noise utilisation component based on the single noise obfuscation process, which will be introduced in the following chapters.

In future, some further discussions on association probabilities can be investigated to obtain more dynamic and precise privacy strategies, such as in the association model, the direct consequential relation can be replaced by some indirect relations.

Chapter 7

Noise Utilisation for Ethical Multiple Services

In this chapter, we start to discuss the noise utilisation component of our novel noise obfuscation model. As introduced before, the noise utilisation component is the last step of the whole noise obfuscation function. After noise pre-processing and noise generation, we have effective and efficient noise data in the single service scenario for privacy protection in cloud computing. Hence, in noise utilisation, we can discuss how to use these noise data and single noise obfuscation processes for the multiple services scenario for privacy protection in cloud computing. In this chapter and the next one, we will discuss these noise utilisations.

As discussed before, noise utilisation in this thesis focuses on the multiple services scenario. In the scenario of single service, noise utilisation is straightforward to be considered: noise injection models in former strategies can clearly express these noise utilisation processes. But in the multiple services scenario, we cannot utilise noise data directly according to these former noise injection models, due to complex and opaque environments among multiple services in cloud computing. That is why we have to investigate the noise utilisation component in terms of multiple services scenario. Specifically, we can consider two kinds of cases on multiple services: ethical multiple services and unethical multiple services.

In this chapter, we focus on the ethical multiple services case. In this case, some ethical multiple services can aid cloud customers to protect privacy from other malicious service providers. Briefly, ethical cloud services could facilitate noise

obfuscation to pursue a more effective privacy protection in cloud computing. Hence, under this privacy concern, we present a novel Correlation based Noise Injection Strategy (*CNIS*) to combine these ethical services together to execute the whole noise obfuscation function. The noise injection architecture is the supporting environment for the noise injection strategy. In this architecture, we can discuss the correlation model and the single service process with noise obfuscation. After that, *CNIS* can be operated based on effectively linking single service processes with single noise obfuscation processes together by correlation. The effectiveness of privacy protection on noise obfuscation can be improved by this novel *CNIS* in the ethical multiple services case in terms of noise utilisation. In this chapter, this noise injection strategy is the noise utilisation strategy to fulfil the noise utilisation component of our novel noise obfuscation model in the ethical multiple services case.

This chapter is organised as follows. Section 7.1 addresses the background of this strategy. Section 7.2 describes the noise injection architecture in cloud computing. Section 7.3 presents our novel Correlation based Noise Injection Strategy (*CNIS*) for privacy protection in cloud computing. Section 7.4 demonstrates the evaluation results to illustrate the advantage of *CNIS*. Finally, Section 7.5 summarises this chapter.

This chapter is mainly based on our work presented in [41].

7.1 Background of the Strategy

Generally speaking, in cloud computing, to protect customer privacy, noise obfuscation belongs to these measures which can aid customers to protect privacy at client side. As the main idea of the whole thesis, it can inject noise service requests into real customers' service requests so that service providers are hard to distinguish which requests are real ones.

But currently, existing noise obfuscations focus on noise utilisation for a single service process. Actually, customers' service requests may need more than one service provider to answer them. It is a cooperative service process with various service providers. For instance, in cloud environments, the style of inter-clouds with

public clouds and privacy clouds is easily accepted for its flexibility on the balance of privacy protection and usability [59]. Especially in inter-cloud environments, a cooperative service process could bring several service providers from different clouds together, and the complexity of privacy protection increases with more and more service providers' taking part in the process due to the spread of privacy. As a result, we need to protect customers' privacy during the entire cooperative service process, which can be addressed as a noise utilisation problem in noise obfuscation. By the way, we use "customers" to denote real people who have privacy to be protected in cloud computing, and "clients" to denote virtualised clients which communicate with service providers in cloud. Hence, a "client" is a network terminal which applies noise obfuscation strategies to protect the privacy of the "customer" that utilises the "client" to get services in cloud computing. It is the computer which a "customer" uses.

To address the noise injection (utilisation) problem introduced above, we can investigate a noise injection architecture for entire cooperative service processes in cloud computing. And it specialises in various single service processes with noise obfuscation in cloud computing based on correlation. They are basic supporting functions to fulfil the architecture. Based on these, we present our novel Correlation based Noise Injection Strategy (*CNIS*) for privacy protection in cloud computing, and it protects customers' privacy during the entire process of services' cooperation. In this strategy, we use noise service requests to protect customers' privacy in a cooperative service process by not only clients, but also other service providers. The correlation model is the bridge to connect clients and services together for the noise injection architecture. The noise injection architecture is used to describe cooperative service processes, and it provides a supporting environment for this novel *CNIS*. Hence, this strategy protects customers' privacy during entire cooperative service processes and addresses the noise utilisation concern in the ethical multiple services cases.

Besides, as presented in Chapter 3, the single noise obfuscation process composes of the noise pre-processing component and the noise generation component. In this chapter, this process is the 'noise obfuscation' in the single service process which is the basis of the noise injection architecture. Under our novel *CNIS* in this chapter, we can connect single service processes with single

noise obfuscation processes together, to execute noise obfuscation in the ethical multiple services case for privacy protection in cloud computing.

Based on the above, this strategy requires ethical service providers to protect customer privacy by cooperating with clients. Service providers could have the motivation to keep customers' privacy safe during the services between themselves and other service providers by noise obfuscation. For instance, service requests sent by this service could also leak customer privacy, just like customers' classification and/or distribution of information. This could be a weapon for its business competitors. To address this, service providers could protect their customers' privacy by facilitating this *CNIS*. Hence, this *CNIS* is feasible to be applied in practice.

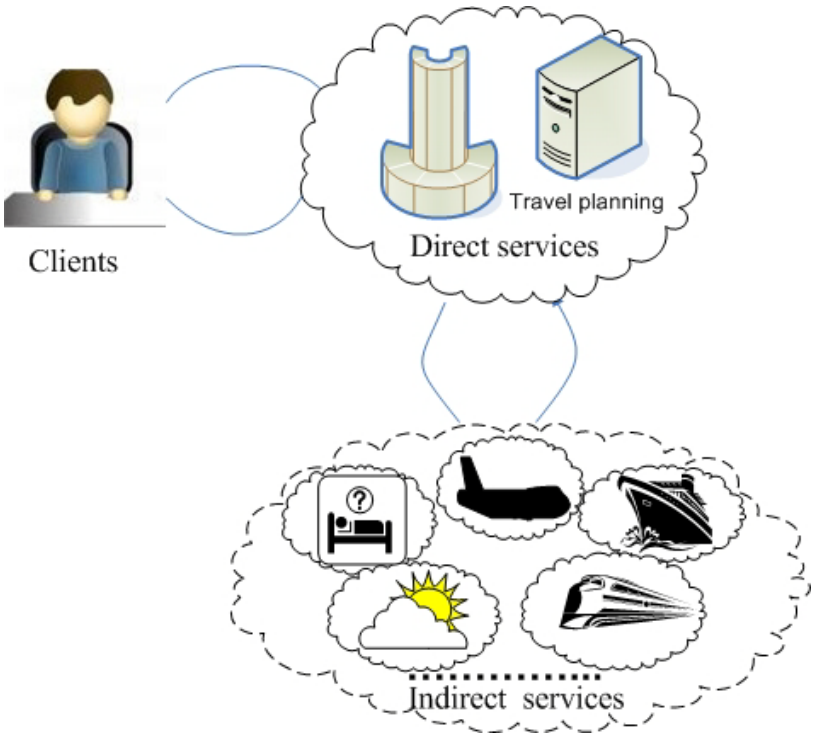


Figure 7-1 A cooperative service process on cloud

Let us take a travel planning service as the motivating example. In Figure 7-1, a client sends a service request to a travel planning service, and this service sends its service requests to other services to get travel information, such as flight service and hotel service. These services answer service requests back to the travel planning service which analyses all information from the answers to provide one or several travel plans and responds back to the client. It can be viewed as a cooperative

service process with ethical multiple services. Obviously, some customers' privacy may be leaked by some potential 'immoral' services in the travel planning process which needs several different services to respond, such as location information and travel itinerary. It can be associated with one customer's identity from other public information to break the customer's identity privacy protection. Hence, the risk of privacy is serious. This is the motivation of this chapter.

In brief, this novel *CNIS* for privacy protection in cloud computing focuses on privacy protection in cooperative service processes, and utilises our correlation model to guide noise generation and injection (utilisation) during entire cooperative service processes. Therefore, customers' privacy can be protected in this ethical multiple services case.

7.2 Noise Injection Architecture in Cloud Computing

To present our novel *CNIS*, we need to present our noise injection architecture for the strategy. To introduce the noise injection architecture, firstly, we need the correlation model between services to connect all single service processes with noise obfuscation. Moreover, in this section, we firstly introduce our correlation model between ethical multiple services. Then, we describe the single service process with noise obfuscation. At last, we present our noise injection architecture.

7.2.1 Correlation Model between Services

Correlation model is a basis in this chapter. It is based on other mature trust model work [96, 74] with modifications. We need to utilise the model to link ethical multiple service providers.

Correlation between Services: In this chapter, we use the 'correlation' between two service roles: a service request initiator and a service request respondent in the view of single service process. A service request initiator could be a client or a service, and a service request respondent is solely a service. We will describe these service roles in detail in the following sections. Now we introduce the correlation.

Correlation: as a major part of correlation model, Correlation is a 8-tuple (P,Q,T,D,t,v,p,n) which asserts the relation between entity P and entity Q, where:

- P and Q are the subset of the set (Ω) of all the entities in cloud computing. In this chapter, P and Q always have different role attributes.
- T is the set of {*direct, recommended, derived*} to denote different correlation types. In this strategy, we only consider the type of "*direct*". Because we plan to use the correlation to guide noise obfuscation, and other two types have to influence noise obfuscation by the type of "*direct*". For noise obfuscation, the type of "*direct*" is adequate.
- D is the set of domains of {<*dn, dt* >} where *dn* denotes the name of the domain; *dt* is the type of the domain and $dt \in DN$ = {*intra, inter*} denoting that the correlation is intra-domain or inter-domain, respectively. We will detail these in the following subsection.
- t is the time constraint when the relation is thought to be valid.
- v is the evaluation on this correlation.
- p is the number of positive experiences associated with this correlation.
- n is the number of negative experiences associated with this correlation.

Correlation's Domains: after defining the correlation model, we can explain domain D of correlation now. In Section 7.3, domain D will play an important role in our novel *CNIS*.

In complex service processes, there are three members: clients, direct services and indirect services. In Figure 7-2, direct services are services which receive service requests from clients, and have to accept some cooperation from other services to answer clients' service requests. Indirect services are services that are out of clients' "views". In general, clients only send service requests to direct services, and direct services may ask other direct services or indirect services to accomplish the function of service process. These indirect services are invisible to clients. There are three kinds of single service processes between them: "client-direct", "direct-indirect" and "direct-direct".

In this chapter, we have an assumption that an indirect service is the last step of a cooperative service process, and it does not send service requests to other indirect services to obtain cooperation. That is because the "indirect-indirect" cannot be enhanced by noise obfuscation functions in clients' views. Hence, we do not have "indirect-indirect" single service processes.

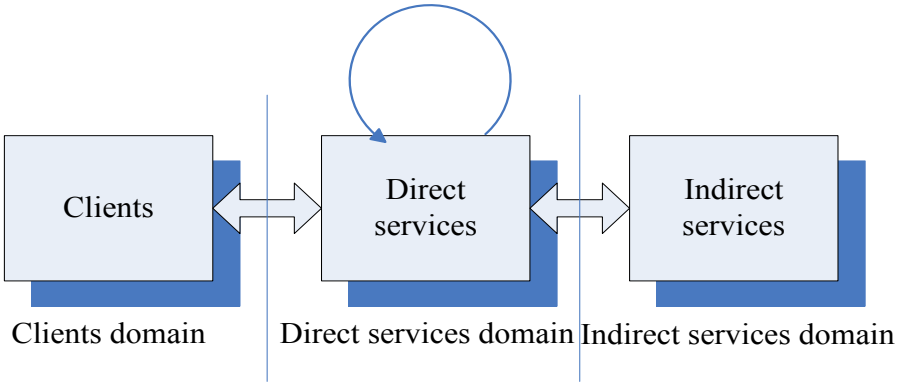


Figure 7-2 Cooperative service processes with clients, direct services and indirect services

Now, we can consider three domains of correlation. There are "clients" domain, "direct services" domain, and "indirect services" domain. They come from three members in cooperative service processes, respectively. Hence, we can discuss these correlation domains from the perspective of single service process.

The correlation between a client and a direct service: As a client, direct services are the only kind of services which can "see" and send its service requests to. The correlation from a client to a direct service is a sectional correlation from the client to the direct service excluding all other services belonging cooperative service processes in complex cloud environments. And this correlation has a domain type with an inter-domain, and the domain of this correlation is <*clients, inter*>. As a direct service, the correlation from a direct service to a client is located at service side which is beyond the scope of this strategy.

The correlation between a direct service and an indirect service: As a direct service, indirect services are all cooperative resources. The correlation from a direct service to an indirect service has a value for distributing tasks to the indirect service in complex cloud environments. And this correlation has a domain type with an inter-domain, and the domain of this correlation is <*direct, inter*>. As an indirect service, the correlation from an indirect service to a direct service is also located at service side which is beyond the scope of this strategy, too.

The correlation between one direct service and another direct service: As one direct service, other direct services are cooperative resources. When this direct service needs some help, it may send some requests to other direct services. As a request initiator, the correlation is a sectional correlation to all other services which

take part in this service process, too. And this correlation has a domain type with an intra-domain, and the domain of this correlation is <*direct, intra*>.

In summary, we presented our correlation model between services based on other mature research work and emphasised domain types of correlations which can be utilised to manage noise generation strategies and noise injections presented in Section 7.3. They are essential issues of our novel *CNIS*.

7.2.2 Single Service Process with Noise Obfuscation

In the view of single service process in cloud computing for noise obfuscation, two service roles in a cooperative service process can be classified: service request initiator and service request respondent. To introduce them, we utilise a single service process to represent one step of an entire cooperative service process with only these two members and single service process. We use this process to cover different domains and clouds in terms of correlation.

Service request initiator plays a main role in this strategy. It guides the noise service requests' generation and injection by the correlation model. Service request respondent plays another role in the process. In this strategy, it just receives and responds service requests from the perspective of a service request initiator.

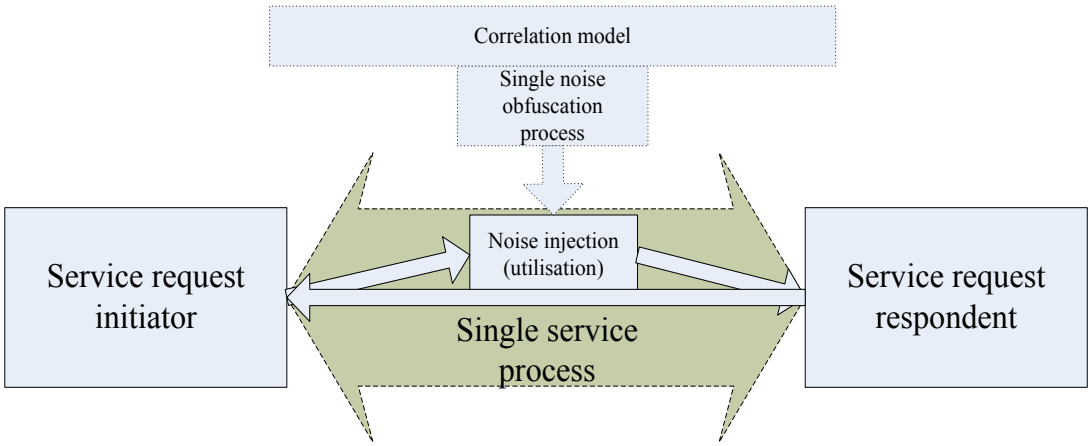


Figure 7-3 Single service process with noise obfuscation

In Figure 7-3, it depicts a single service process between a service request initiator and a service request respondent. From the perspective of a service request respondent, the process is very simple and just operates between itself and a service request initiator. The service request initiator protects privacy by noise injection

which is invisible to its partner (the service request respondent). The single service process is a combination of the dual service roles with noise injection, and a process to execute these two service roles. Hence, we utilise single service processes to model entire cooperative service processes.

About "single noise obfuscation process" and "correlation model", they are important issues to support single service processes under our novel *CNIS* in this chapter. By the way, this "single noise obfuscation process" is the "single noise obfuscation process" discussed before, composed of the noise pre-processing component and the noise generation component of our novel noise obfuscation model presented in Chapter 3.

In the next subsection, we will use single service processes with noise obfuscation to present the noise injection architecture which is the supporting environment for our novel *CNIS*.

7.2.3 Noise Injection Architecture

In Section 7.2.1 and 7.2.2 above, we introduced the correlation model between services and the single service process with noise obfuscation which are parts of the noise injection architecture. In this subsection, we present our noise injection architecture to support our novel *CNIS*. In Figure 7-4, we have three domains in this architecture: clients domain, direct domain and indirect domain. The clients and direct domains are visible to customers, and our novel *CNIS* is deployed in these two domains to protect customers' privacy. Indirect domain is invisible to customers.

From the virtualisation perspective, we have three layers in the entire architecture: the role layer, the service layer and the deployment layer which correspond to the role environment, the service environment and the cloud deploying environment, respectively.

What we focus on is the grey parts in the role layer. Based on single service processes with service request initiators and service request respondents, noise injections have been highlighted as essential parts in these single service processes with noise obfuscation. To generate noises to protect privacy, we need some single noise obfuscation processes which have been highlighted to be operated before noise injections. Besides, these processes can be executed by the correlation model as

highlighted, too. In the next section, our novel *CNIS* will be proposed accordingly.

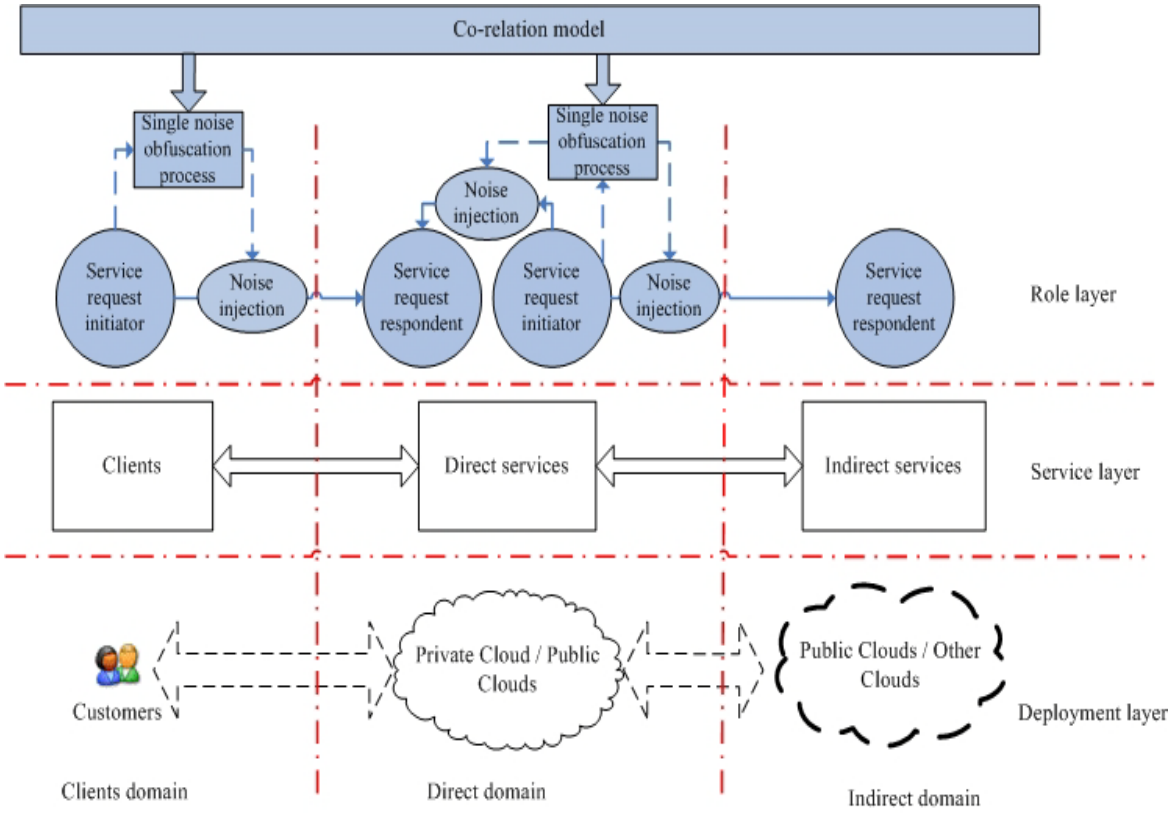


Figure 7-4 Noise injection architecture

7.3 Novel Noise Injection Strategy

In the preceding section, the noise injection architecture in cloud computing was presented as a supporting environment to our noise injection strategy. In this section, we propose our novel Correlation based Noise Injection Strategy (*CNIS*) for privacy protection in cloud computing. Our novel *CNIS* plays its role in single service processes with noise obfuscation which are the grey parts of noise injection architecture depicted in Figure 7-4. Every single service process with noise obfuscation in this architecture utilises *CNIS* to manage noise generation to protect privacy.

Correlation based Noise Injection Strategy : *CNIS*
Input: Service request queue: Q_R, Initial correlations: $CR=\{cr_1,cr_2,\ldots\ldots,cr_i,\ldots\ldots,cr_m\}$, Risk of privacy: d, Quality evaluation of this single-service process: e.

<table>
<tr><td colspan="2">Output: Final service request queue Q_S
Updated correlations: $CR=\{cr_1,cr_2,\ldots\ldots,cr_i,\ldots\ldots,cr_m\}$,</td></tr>
<tr><td rowspan="3">Step 1: Evaluating correlation</td><td>Input: Correlations between services CR;
Dual service roles in this noise injection: p and q</td></tr>
<tr><td>Output: v denotes the correlation value between p and q in the range of [0, 1]</td></tr>
<tr><td>From Section 7.2.1, we have correlations: $CR=\{cr_1,cr_2,\ldots\ldots,cr_i,\ldots\ldots,cr_m\}$, where $cr_i=(P_i,Q_i,T_i,D_i,t_i,v_i,p_i,n_i)$.
Check all c_i in cr_i, remove all cr_i with unavailable cr_i.
Check all i which can satisfy conditions: $P_i==p$ and $Q_i==q$.
If i does not exist, Dijkstra shortest path algorithm is used to find out an array of correlations, and a newly derived cr will be inserted into CR. Then this step will be re-executed.
If i exists, if i is unique, $v=v_i$.
if i is not unique, $v=v_j$ (where $j \in$ dataset of i and $T_j==$"direct")</td></tr>
<tr><td rowspan="3">Step 2: Evaluating risk of privacy in this single service process with noise obfuscation</td><td>Input: cr_i denotes the correlation in a single-service process,
$d \in \{$serious, moderate, minor$\}$ which denotes initial level of privacy risk from customer's judgements.</td></tr>
<tr><td>Output: $d' \in \{$serious, moderate, minor$\}$ which denotes final level of privacy risk.</td></tr>
<tr><td>Check D_i from cr_i.
If (D_i ==<<i>clients, inter</i>>), $d'=d$.
If (D_i ==<<i>direct, intra</i>>), $d'=d$.
If (D_i ==<<i>direct, inter</i>>), if (d ==“<i>minor</i>”), $d'=$ “<i>moderate</i>”.
if (d!=“<i>minor</i>”), $d'=d$.</td></tr>
<tr><td rowspan="4">Step 3: Settling down noise injection intensity, and generating noise for injecting into service request queue Q_R</td><td>Input: Q_R, d', v</td></tr>
<tr><td>Output: Q_S, with Q_N denoted by $P(Q_N=q_i),\forall i$ and ε denotes noise injection intensity</td></tr>
<tr><td>3.1 Settle down the noise generation strategy.
If (d' == “minor”), go to Step 3.2, “minor” noise generation strategy to be applied.
If (d' == “moderate”), go to Step 3.3, “moderate” noise generation strategy to be applied.
If (d' == “serious”), go to Step 3.4, “serious” noise generation strategy to be applied.
These three strategies will be introduced next, respectively.</td></tr>
<tr><td>3.2 The “minor” noise generation strategy
We generate noise Q_N by noise generation probabilities, such as :
$P(Q_N=q_i)=\frac{1}{n}$,
and noise injection intensity, such as:
$\varepsilon=1-v$
Go to Step 3.5.</td></tr>
<tr><td>3.3 The “moderate” noise generation strategy
We generate noise Q_N by noise generation probabilities, such as :
$P(Q_N=q_i)=\frac{Max\{P(Q_R=q_i)\}-P(Q_R=q_i)}{n*Max\{P(Q_R=q_i)\}-\sum_i P(Q_R=q_i)}$,
and noise injection intensity, such as :
$\varepsilon'=2(1-v)\frac{n*Max\{P(Q_R=q_i)\}-\sum_i P(Q_R=q_i)}{n*Max\{P(Q_R=q_i)\}}$ and the final $\varepsilon=Max\{\varepsilon',1-v\}$.
Go to Step 3.5.</td></tr>
</table>

	3.4 The "*serious*" noise generation strategy We generate noise Q_N by noise generation probabilities, such as: $P(Q_N = q_i) = \frac{Max\{P(Q_S = q_i)\} - P(Q_S = q_i)}{n * Max\{P(Q_S = q_i)\} - \sum_i P(Q_S = q_i)}$, and noise injection intensity, such as: $\varepsilon(t) = 2(1-v)\frac{n * Max\{P[Q_S(t) = q_i]\} - \sum_i P[Q_S(t) = q_i]}{n * Max\{P[Q_S(t) = q_i]\}}$ and the final $\varepsilon = Max\{\varepsilon(t), 1-v\}$, which changes with time *t*. Go to Step 3.5.
	3.5 Noise injection We get the noise *N* from Q_N, and inject it into Q_S on the probability of ε, hence we can get Q_S. In this step, we execute this noise injection process.
Step 4: Evaluating quality of this single service process, and updating correlation in correlation model about this single service process	Input: *e* denotes quality evaluation of this single service process.
	Output: Updated cr_i
	4.1 Get a feedback *e* to denote the quality of the service In this step, we collect feedback *e* from a service request initiator.

	4.2 Update p_i and n_i in cr_i
	If($e \geq v_i$), $p_i = p_i + 1$
	If($e < v_i$), $n_i = n_i + 1$

	4.3 Update v_i in cr_i $v_i = v_i + e \times (p_i - n_i)$

In this noise injection strategy, we choose three noise generation strategies to execute noise generation. The three noise generation strategies have different customers' privacy. Independent to noise generation strategies we presented in the previous chapters, these three noise generation strategies only express some common noise generation processes to support noise utilisation processes in this strategy. There are three types of noise generation strategies in terms of correlation and private information. In other words, noise generation strategies presented in Chapter 5 and Chapter 6 can be modified in these types as needed.

We set three kinds of customers' privacy: $pl \in \{$*direct privacy, probability distribution, interaction frequency*$\}$. The first one denotes that original service requests without any analysis and induction. The second one denotes that this kind of privacy can be induced from the occurrence probabilities of original service requests, such as in *HPNGS* [42], *TPNGS* (Chapter 5) and *APNGS* (Chapter 6). The last one denotes that the interaction frequency of original service requests can induce this kind of privacy.

Hence, it is easy to understand that each strategy sets one kind of privacy as its

goal of noise privacy protection: the goal of the "*minor*" noise generation strategy is to conceal the *pl*= "*direct privacy*"; the goal of the "*moderate*" noise generation strategy is to conceal the *pl*= "*probability distribution*"; and the goal of the "*serious*" noise generation strategy is to conceal the *pl*= "*interaction frequency*".

About ε's generation in these three strategies, when the risk of privacy rises, it changes from $\varepsilon = 1 - v$, $\varepsilon = Max\{\varepsilon', 1 - v\}$ to $\varepsilon = Max\{\varepsilon(t), 1 - v\}$ by increasing step by step. It means that if the risk of privacy increases, the noise injection intensity increases as well with noise generation strategies changing accordingly.

In this noise injection strategy, we have an initial level of privacy risk *d* which is decided by customers, and do some amendments for the final level of privacy risk in our strategy. Similarly, we use customers' quality evaluation to operate the updating function of correlation in Step 4. Correlations can be adapted, and the updating function realises this for improving the adaptability of the correlation model.

In summary, our novel *CNIS* for privacy protection in cloud computing is established on the correlation model and the single service process with noise obfuscation. It operates on the noise injection architecture with several single service processes and dual service roles, and protects customers' privacy during entire cooperative service processes in the cloud ethical multiple services case. In the following section, we will illustrate that our novel *CNIS* can protect privacy better than the existing Single Noise Injection Strategy (*SNIS*).

7.4 Simulation and Evaluation

In this section, we introduce an experimental simulation in our SwinCloud, like previous strategies in this thesis. And the simulation executes an instance about a cooperative service process. The aim is to simulate and demonstrate that our novel strategy improves the effectiveness of privacy protection on noise obfuscation significantly. Hence, we set several nodes to represent two service roles—service request initiator and service request respondent, respectively. The former one firstly generates or forwards real service request queues where every request is from a request set with 50 items. Then, they generate noise service request queues to inject into real request queues. The latter one receives the final service request queues and

analyses the effectiveness of privacy protection on noise obfuscation. Some nodes represent dual roles as the intermediate steps of cooperative service processes. That is the main idea of the simulation process. In this process, we set the cooperative service process with all *Services* as a linear structure depicted in Figure 7-5.

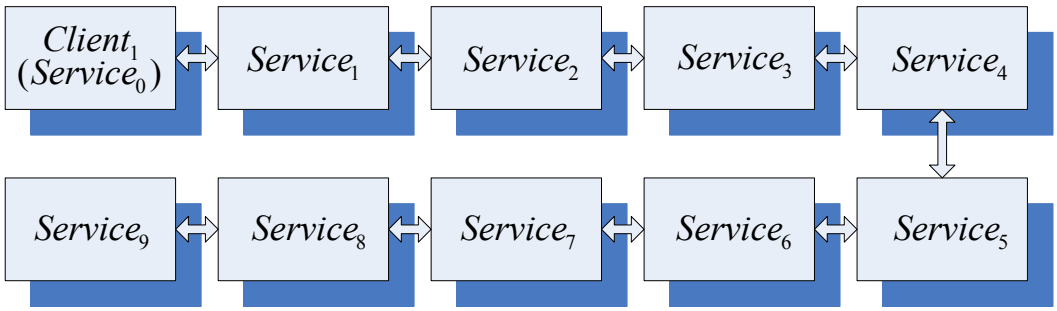


Figure 7-5 Linear service structure

In Figure 7-5, $Client_1$ (it can also be viewed as $Service_0$) sends service requests to $Service_1$ with noise obfuscation, then $Service_1$ operates this information and sends to the next service—$Service_2$ with noise protection too, and the same steps are repeated until $Service_9$ which is the last service to accomplish the client's service requests.

Except the last service, all services and the client can inject noise to protect privacy. In the simulation process, we make a comparison between our novel Correlation based Noise Injection Strategy (*CNIS*) and the existing Single Noise Injection Strategy (*SNIS*). To evaluate the effectiveness of privacy protection in this chapter, we set $r_i(CNIS)$ and $r_i(SNIS)$ from $r_i = \prod_i \frac{\varepsilon_i}{v_i}$ which denotes the risk of privacy under the protection of one noise injection strategy at step i. ε_i is the noise injection intensity at the service step between $Service_{i-1}$ and $Service_i$. v_i is the correlation value between $Service_{i-1}$ and $Service_i$. It is obvious that a lower r_i means a better effectiveness of privacy protection on noise obfuscation at step i.

About *SNIS*, we utilise it to represent various existing noise obfuscations in the view of noise injection. They all focus on noise generation and one single service process with single noise obfuscation process, regardless noise utilisation in the ethical multiple services case. In other words, it can be described by a range of independent single noise obfuscation processes without noise utilisation functions for multiple services scenarios.

Based on the simulation process described before, we have $r_i(CNIS)$ and $r_i(SNIS)$. They are depicted in Figure 7-6. And they change by the step of service process i.

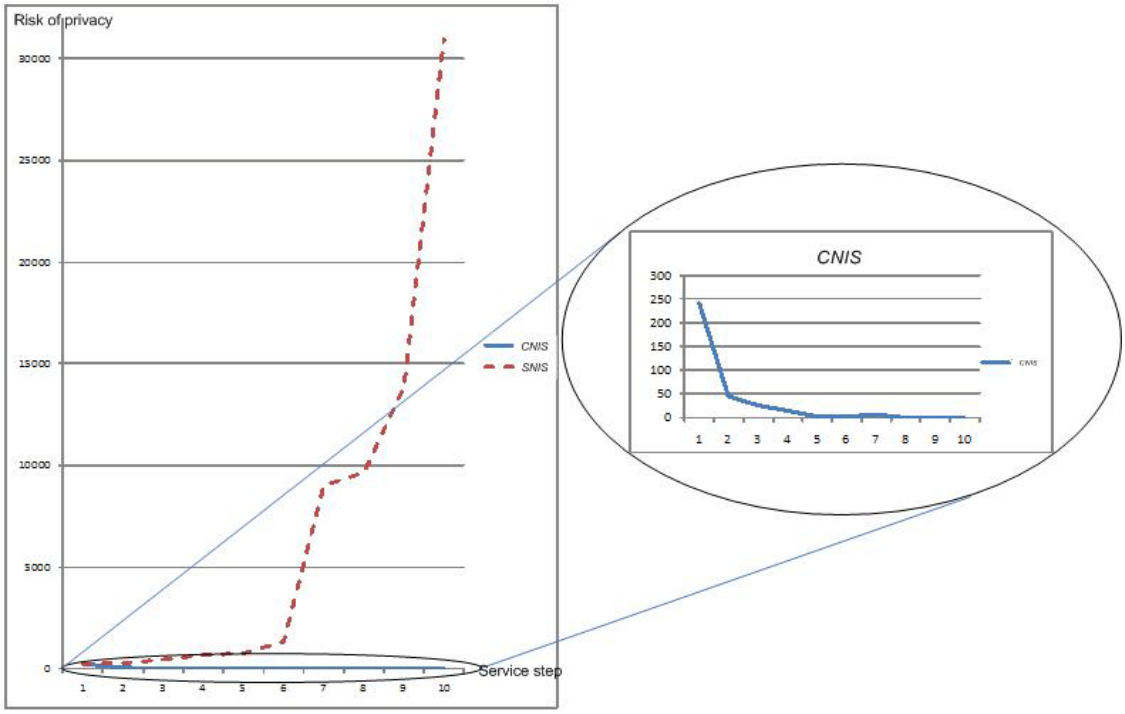


Figure 7-6 Comparison between $r_i(CNIS)$ and $r_i(SNIS)$

In Figure 7-6, the horizontal coordinate is the step (i) in the service process. The vertical coordinate is the risk of privacy ($r_i = \prod_i \frac{\varepsilon_i}{v_i}$). If the risk of privacy r_i is lower, the effectiveness of privacy protection is better, and vice versa.

Obviously, $r_i(CNIS)$ and $r_i(SNIS)$ have a same start, but later on they are extremely different. The former keeps the risk of privacy in a zone of moderate fluctuation and the latter increases rapidly. With the increasing of i, the disparity between $r_i(CNIS)$ and $r_i(SNIS)$ becomes more and more, and our strategy— *CNIS* is more and more effective. At $Service_1$, these two privacy risks are the same. At $Service_5$, the disparity of two privacy risks is 614.301 times. At $Service_9$, the disparity of two privacy risks is as high as 156128.55 times. Therefore, our strategy can improve the effectiveness of privacy protection on noise obfuscation significantly and keep the risk of privacy low, especially in complex cooperative service processes.

As a client, it could know the single service process with noise obfuscation with $Service_1$ instead of this entire service process as discussed in Section 7.1. That is why *CNIS* and *SNIS* start at a same level of privacy protection. However, with other services joining the service process, *SNIS* could expose the disadvantage and have a much worse effectiveness of privacy protection, due to neglecting the entire cooperative service process.

In summary, we can conclude that our novel *CNIS* could decrease the risk of privacy significantly than existing noise obfuscations in cooperative service processes in cloud computing. In other words, in noise utilisation processes, we can improve the effectiveness of privacy protection on noise obfuscation significantly by this strategy for the ethical multiple services case.

7.5 Summary

In this chapter, we presented our novel Correlation based Noise Injection Strategy (*CNIS*) for privacy protection in cloud computing. In this strategy, we introduced the noise injection architecture in cloud computing which was based on the single service process with noise obfuscation and the correlation model. Hence, customers' privacy can be considered and protected at every step of entire cooperative service processes by this strategy. In one word, the key work in this chapter is to extend single noise obfuscation processes to cooperative noise obfuscation processes in the ethical multiple services case for privacy protection in cloud computing.

As introduced before, the noise injection strategy described in this chapter considers the noise utilisation function for the ethical multiple services case. As a noise utilisation strategy, this strategy focuses on how to utilise the single noise obfuscation process effectively including noise generation and noise pre-processing. Besides, the ethical multiple service case can promote noise utilisation by correlation. In this case, it is obvious that the effectiveness of privacy protection on noise obfuscation can be improved in terms of noise utilisation, which has been demonstrated by the simulation evaluation. In the next chapter, we will discuss another multiple services case—the unethical multiple services case.

In future, we will improve our work on the correlation model. For example, the

correlation model can be modified by considering quantified correlations to replace the qualified correlations in this chapter now. In other words, it would not have three discrete levels of noise utilisation, but a continuous range of noise utilisation.

Chapter 8

Noise Utilisation for Unethical Multiple Services

As discussed before, we mainly consider the noise utilisation in the multiple services scenario in Chapter 7 and this chapter. In Chapter 7, we presented our novel correlation based noise injection strategy to discuss how to utilise noise data and single noise obfuscation processes for the ethical multiple services case. In this chapter, we discuss the noise utilisation function for the unethical multiple services case. Briefly, in this chapter, noise obfuscation and noise utilisation have to consider these potential cooperation relations among unethical or malicious services for noise obfuscation in cloud computing. Under this privacy risk, the noise utilisation function and strategy should focus on managing single noise obfuscation processes to deal with the malicious cooperation of unethical multiple services. In other words, when these unethical services cooperate and share information about customer privacy, noise obfuscation has to improve itself to withstand this serious privacy risk in terms of noise utilisation.

To address this privacy risk, we present a novel Common Set based Noise Cooperation Strategy (*CSNCS*) as the noise utilisation strategy to withstand the unethical multiple services case in the noise utilisation component of our novel noise obfuscation model. In this strategy, the common set is the key basis to guide and utilise single noise obfuscation processes together to be effective by creation of the noise set(s). Based on the common set, different single noise obfuscation processes can be cooperated and utilised together to withstand this unethical multiple services case in terms of noise utilisation. That is the noise cooperation process which is

executed by this novel noise cooperation strategy. In this chapter, we can complete the noise utilisation component of our novel noise obfuscation model by this strategy.

This chapter is organised as follows. Section 8.1 introduces the background of this strategy. Section 8.2 presents the noise cooperation model to discuss the unethical multiple services case. Section 8.3 describes the common set creation model for noise obfuscation to be the key issue of the novel noise utilisation strategy. Section 8.4 proposes our novel Common Set based Noise Cooperation Strategy (*CSNCS*) for privacy protection in cloud computing. Section 8.5 demonstrates this noise cooperation strategy by simulation and evaluation. Finally, Section 8.6 summaries this chapter.

8.1 Background of the Strategy

For noise obfuscation, multiple services scenarios are necessary to be considered in the noise utilisation component. After we discussed the ethical multiple services case in Chapter 7, we investigate the unethical multiple services case as a privacy challenge now, which means malicious services could combine together to break existing noise obfuscations and obtain customer privacy.

In cloud computing, it is possible that one customer utilises different service processes at the same time. For each service process, there could be a single noise obfuscation process to protect privacy by concealing real requests' probabilities for each service provider. Besides, due to the openness and virtualisation features in cloud environments, it is hard for customers to find out which services may be unethical and cooperate to share their service data about the customer to deduce some more private information. Hence, this service-cooperation could give unethical services a chance to break existing noise obfuscations together.

Specifically, existing noise obfuscations utilise noise service requests to conceal real ones from a set of service data. But during data sharing in the unethical multiple services case, the overlapped part from several service data sets from different service providers could be found by unethical services. This part could aid these services to omit many 'useless' noise service data which are not in the intersection

of these sets. Hence, the effectiveness of privacy protection on noise obfuscation is considerably lower than customers expected. Besides, these 'useless' noise service data mean a 'useless' cost for customers in the pay-as-you-go style of cloud environments. Obviously, existing noise obfuscations have not considered this risk before. Hence, this privacy risk could damage the confidence of customers to cloud computing in terms of noise obfuscation eventually.

Besides, this privacy risk is quite similar to intersection attacks [15, 11, 80]. They both have to withstand one kind of private data leakage by information combination and fusion. But in this chapter, this privacy risk is based on the noise obfuscation approach, not in the field of data publishing or mining like intersection attacks. This is a difference in the area of application. Besides, the privacy risk in this chapter is considered from the perspective of cloud customers, which is totally different with intersection attacks in the view of service providers.

To address this privacy risk, we develop our novel Common Set based Noise Cooperation Strategy (*CSNCS*) for cloud privacy protection in the unethical multiple services case. We firstly analyse the overlapped part of all service request sets from single service processes with single noise obfuscation processes. Then, based on the overlapped part, we consider a common set for noise utilisation to manage and utilise each single noise obfuscation process. Hence, creating the common set is the key task in this strategy. To obtain this set, we investigate the creation procedure of the common set from the overlapped part of every request set. Besides, we investigate the updating process of this common set in the process of noise utilisation under unethical multiple services in cloud computing, as a dynamic function. Based on that, we lastly present our novel noise cooperation strategy—*CSNCS* to execute the noise utilisation function in the unethical multiple services case.

Besides, compared to the common set discussed in this chapter, the noise set discussed in Chapter 4 about the noise pre-processing component can be viewed as a similar set for noise generation. But in Chapter 4, the noise set can be discussed in terms of noise pre-processing to connect customers' requirements and noise obfuscation in a single noise obfuscation process. In this chapter, this common set is discussed from several former noise sets in several single noise obfuscation processes, and means a key issue to protect privacy by noise obfuscation in multiple

services scenarios. In other words, in the unethical multiple services case with noise pre-processing, this noise cooperation strategy will be executed on the noise sets after noise pre-processing strategies in terms of the noise set(s).

Let us take the cloud weather report as the motivating example, again. As introduced in Section 1.2.2, customer privacy about location information in service requests can be protected in general by existing noise obfuscation. But in reality, he/she may also use one traffic service with the weather service as a multiple services scenario. To protect his/her privacy, as the noise utilisation function, two single noise obfuscation processes can be executed with specific noise generation strategies: the weather forecasting service may receive "Sydney", "Melbourne" and "Perth" service requests with about the same probabilities, and the traffic service could receive "Sydney", "Melbourne" and "Brisbane" service requests with about the same probabilities, too. Each of them can not distinguish which one is the real city the customer will go and privacy can be protected. But in the unethical multiple services case, the two service providers may share these service sets out of the view of customers for the virtualisation feature. It is clear that "Sydney" and "Melbourne" are the overlapped part of the two "city" sets. And other service requests about "Perth" and "Brisbane" can be omitted in the customer privacy analysis process by some malicious service providers. These malicious service providers only have to find the real city–"Sydney" from a narrowed intersection set: "Sydney" and "Melbourne". It is a serious privacy risk for noise obfuscation. Besides, these noise service requests about "Perth" and "Brisbane" are 'useless' for privacy protection in this situation, and it is a significant waste in cloud computing. To address this privacy risk, the "narrowed" set: "Sydney" and "Melbourne" is the common set which we should focus on in terms of the effectiveness of privacy protection on noise obfuscation. By creating this set on purpose, we can manage the whole noise obfuscation function to be functional in the unethical multiple services case. That is the main motivation of this chapter.

In this chapter, firstly, we plan to introduce the noise cooperation model to support *CSNCS*. Then, the creation of the common set is described. Lastly, we present our novel Common Set based Noise Cooperation Strategy (*CSNCS*) to realise the above noise utilisation process and complete the whole noise obfuscation function for unethical multiple service case to protect privacy in cloud computing.

8.2 Noise Cooperation Model

In brief, the noise cooperation model is an abstract model to describe the noise utilisation process in the unethical multiple services case.

As introduced before, there are several services in the noise utilisation in this chapter. Based on denotations introduced before in Figure 4-1, we can update these Q_R, Q_N, Q_S, Q and ε to $Q^i{}_R$, $Q^i{}_N$, $Q^i{}_S$, Q^i and ε_i. Hence, for a customer, there are *n* service providers which the customer would use and communicate with. They are: $Service_1, Service_2, ...Service_n$. For one specific $Service_i$, there is a set of service requests from the customer: $Q^i = \{q_1^i, q_2^i, ..., q_{ni}^i\}$. It means that all service requests are from this set. The *ni* means the number of service requests in this set. Besides, we have a common privacy set: $D = \{d_1, d_2, ..., d_m\}$, and there are *m* different data items. It is the union of every set: $D^i = \{d_1^i, d_2^i, ..., d_{ni}^i\}$ which means these corresponding datasets from different sets of service requests: Q^i. Hence,

$$D = D^1 \cup D^2 \cup ..... \cup D^n \qquad \textbf{Formula 8-1}$$

We use a map to connect Q^i and D^i: $f_i : Q^i -> D^i$. It is obvious that f_i is an injective map, and a surjective map, too. Thus, it is a bijective map, and their norms are:

$$\left\| Q^i \right\| = \left\| D^i \right\| = ni \qquad \textbf{Formula 8-2}$$

To simplify the map, we can define f_i as:

$$d_j^i = f_i(q_j^i), \forall j \in [1, ni] \qquad \textbf{Formula 8-3}$$

Furthermore, the intersection of all D^i is:

$$D' = D^1 \cap D^2 \cap ..... \cap D^n \qquad \textbf{Formula 8-4}$$

As discussed before, the intersection of all D^i is an expression of the effectiveness of privacy protection under the privacy risk in this chapter. We can

increase $\|D'\|$ to get a better effectiveness of privacy protection with an increment on noise service request cost. Hence, it is a trade-off between the effectiveness and cost.

About the cost, we define the cost set about every particular service request: $S = \{s_1, s_2, ..., s_n\}$ where s_i means the unit cost of one service request with $Service_i$. Compared to real service requests, the cost of noise service requests which are generated by noise obfuscation is the issue we focused on in this chapter. Besides, corresponding $Service_i$, noise injection intensity ε_i and the number of real service requests $numr_i$ can form the overall noise request cost for every service process:

$$Cost = \sum_i s_i \times numr_i \times \frac{\varepsilon_i}{1-\varepsilon_i}, \forall i \in [1, n] \qquad \textbf{Formula 8-5}$$

Just like in other chapters in this thesis, noise injection intensity ε_i means the percentage of noise service requests in all service requests when final service requests have about the same probabilities. Hence, noise injection intensity ε_i is:

$$\varepsilon_i = NGS(Q^i{}_R, Strategy_i) \qquad \textbf{Formula 8-6}$$

In formula 8-6, $Q^i{}_R$ is the queue of real service requests to $Service_i$, and $NGS()$ expresses the abstract process of noise obfuscation under one single noise obfuscation process with one specific noise generation $Strategy_i$ to get noise injection intensity ε_i.

From $\|D'\|$ to $\|D\|$, a number *epp* is utilised to describe the effectiveness of privacy protection on noise obfuscation in this chapter:

$$\|D'\| \le epp \le \|D\| \qquad \textbf{Formula 8-7}$$

Corresponding to *epp*, a set D_e is the goal that our noise cooperation strategy wants to obtain: $\|D_e\| = epp$, and

$$D' \subseteq D_e \subseteq D \qquad \textbf{Formula 8-8}$$

Based on the above discussions, we present the noise cooperation model in

Figure 8-1 to support our novel *CSNCS*. In the left part of this figure, the customer uses *n* different services in cloud computing, and there are *n* single noise obfuscation processes to protect these service processes. But in cloud environments, the "Service cooperation and information sharing" could exist and damage noise obfuscation in terms of privacy protection. Hence, in the right part of this figure, a brief description about service requests and sets is proposed, summarised from the left part of the figure. The "Common" set D_e and "Noise generation request sets" Q_e^i describe the whole process.

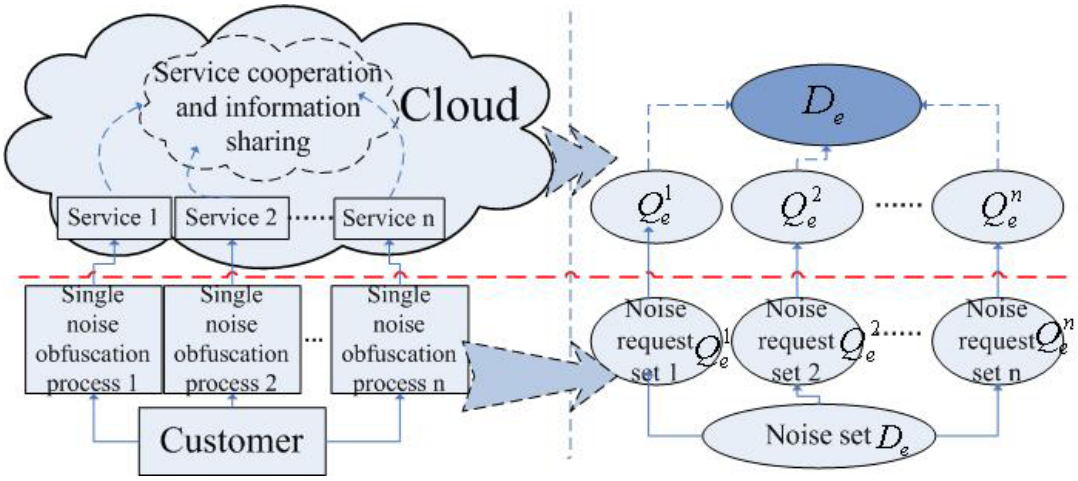


Figure 8-1 Noise cooperation model under service cooperation

Hence, it is obvious that in the service cooperation situation—the unethical multiple services case, the common set D_e is one key issue for noise utilisation to cooperate single noise obfuscation processes with noise generation strategies. And we will discuss how to create it in the next section.

8.3 Common Set Creation Model for Noise Obfuscation

In this section, we present the key part of our novel *CSNCS*—the Common Set Creation Model (*CSCM*). Firstly, some analysis on the creation process of common set is necessary to present the whole model. Then, the Common Set Creation Algorithm (*CSCA*) is proposed to describe one single creation procedure of the common set. At last, with initial creation and run-time updating situations, the Common Set Creation Model (*CSCM*) is presented to build and maintain the

common set.

8.3.1 Problem Analysis

From the preceding section, we have the target of *CSCM*: to find a suitable D_e which conforms to formula 8-5. Hence,

$$Min\{Cost, \forall D_e\} = Min\{\sum_i s_i \times numr_i \times \frac{\varepsilon_i}{1-\varepsilon_i}, \forall D_e\} \qquad \textbf{Formula 8-9}$$

Besides, We have D_e as the common set: $D_e = D' + D_{plus}$ where D_{plus} is a part of D_e. Accordingly,

$$\Phi \le D_{plus} \le D - D' \qquad \textbf{Formula 8-10}$$

In other words, D_{plus} is the target of *CSCM*. Besides, to get the lowest *Cost*, every possible D_e or D_{plus} should be computed and compared. Thus, there are $C_{\|D\|-\|D'\|}^{\|D_e\|-\|D'\|}$ kinds of possible D_e or D_{plus} that should be tested. And

$$C_{\|D\|-\|D'\|}^{\|D_e\|-\|D'\|} = C_{m-\|D'\|}^{epp-\|D'\|} = \frac{(m-m')!}{(epp-m')!(m-epp)!} = C \qquad \textbf{Formula 8-11}$$

The computational complexity of this problem is $O(C \times n \times t(\varepsilon))$ where $t(\varepsilon)$ means that the time computational complexity of the noise injection intensity computing process which depends on different noise generation strategies in single noise obfuscation processes. To get the common set with a lowest cost, we need to compare all the costs of every possible D_e or D_{plus}. Therefore, in the following subsection, we try to decrease this computational complexity.

8.3.2 *CSCA:* Common Set Creation Algorithm

How to obtain D_e? An intuitive approach is to evaluate every data item in D_{plus} and sorting them by the cost on noise service requests. Hence, under $\|D_e\| = epp$,

common set D_e is the union dataset of D' and the previous $epp - m'$ data items in sorted D_{plus}.

For single noise obfuscation processes, there are *n* different noise request sets, one of them, say Q_e^i corresponds the *ith* service process. As introduced before, D_e could be separated to D' and D_{plus}. Thus,

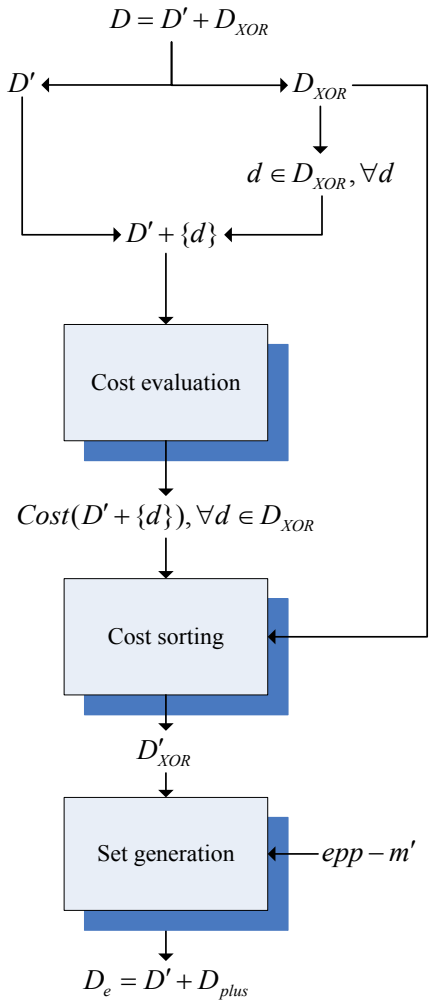
$$Q_e^i = f_i^{-1}(D') + f_i^{-1}(D_{plus}) = Q'^i + Q_{plus}^i \qquad \textbf{Formula 8-12}$$

In formula 8-12, Q'^i is the basic part of Q_e^i from the perspective of noise generation. As discussed before, the real private information must be in Q'^i, and is impossible to be in Q_{plus}^i. Based on Q'^i, all noise generation strategies in single noise obfuscation processes pursue a common noise obfuscation goal—all requests have about the same probabilities. If a new data item $d_s \in D$ is added in D_e, every possible Q_{plus}^i has new requests $f_i^{-1}(d_s)$ to be added in. And every noise generation strategy should make these new requests to be about the same probabilities in Q'^i. Consequently, it is easy to understand that the order of new requests injection in Q_{plus}^i has no influence on the final statement of service requests at service side. Hence, every single noise cost on every data item in D_{plus} is decided by D', and they have no influence on each other.

Hence, based on the above analysis, we can present *CSCA* in Algorithm 8-1. This algorithm generates D_e from D and $epp - m'$. There are three components operated step by step in the algorithm: "cost evaluation", "cost sorting" and "set generation". The former one "cost evaluation" is the key issue in this algorithm. As introduced before, to compute the noise request cost based on D_e, the specific noise generation strategy should be considered for different strategies that have different noise injection intensities from formulas 8-5 and 8-6.

In "cost sorting", a sorting algorithm can be utilised for every data item in D_{XOR}, based on the noise request cost $Cost(D' + \{d\}), \forall d \in D_{XOR}$ which is the result of "cost evaluation".

In "set generation", D_{plus} can be generated from 'cost-sorted' D'_{XOR} under $epp-m'$ which is a parameter from customers. Besides, the cost of every single service request $s_i, \forall i \in [1,n]$ is fixed. Hence, the cost should be settled down before service utilisation in the view of customers in cloud customers.



Algorithm 8-1 *CSCA*: Common Set Creation Algorithm

Based on the above analysis, the computational complexity of *CSCA* is $O((m-m') \times n \times t(\varepsilon))$. Compared to formula 8-11 and the previous time computational complexity: $O(C \times n \times t(\varepsilon))$ in Section 8.3.1, it is a significant decreasing.

In the next subsection, we present *CSCM* based on *CSCA*.

8.3.3 *CSCM:* Common Set Creation Model

In this subsection, we can present *CSCM*. Briefly, *CSCM* builds based on *CSCA*, and investigates common set D_e in the initial creation and the run-time updating.

In the initial creation process of common set D_e, from formulas 8-5 and 8-6, past service request queues and sets can be utilised to compute and obtain initial common set D_e. In Algorithm 8-1, these past requests can decide the initial D_e to support *CSNCS*. It is the beginning of *CSCM* execution.

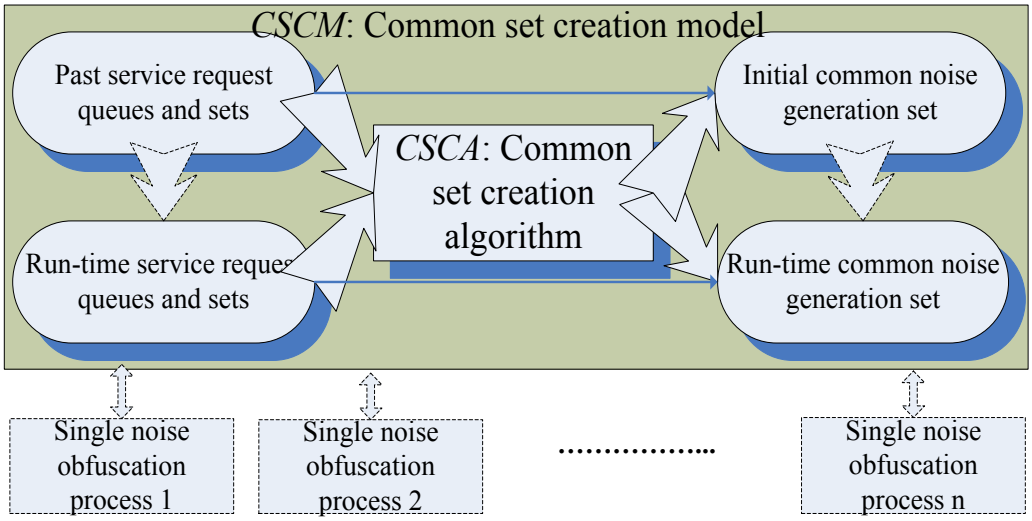


Figure 8-2 *CSCM*: Common Set Creation Model

In the run-time updating process of common set D_e, from same formulas 8-5 and 8-6, run-time service request queues and sets are necessary to maintain a reasonable D_e. In Algorithm 8-1, it is obvious that these run-time service request queues can impact all noise injection intensities in single noise obfuscation processes in the step of "Cost evaluation". Besides, these run-time service request sets may vary from past service request sets at the beginning. Hence, based on *CSCA*, the run-time updating on the common noise set is an important part of *CSCM*.

In Figure 8-2, *CSCM* builds based on *CSCA*. Both in processes of the initial creation and run-time updating, *CSCA* is the critical part to obtain common set D_e. In general, *CSCM* presents the complete view for the creation of common set D_e, and takes *CSCA* into practice to be the kernel of *CSNCS* for withstanding the unethical multiple services risk on noise obfuscation in cloud computing.

8.4 Noise Cooperation Strategy

Based on preceding sections, we present the noise utilisation strategy in the unethical multiple services case—our novel Common Set based Noise Cooperation Strategy (*CSNCS*) for privacy protection in cloud computing in Algorithm 8-2. Based on *CSCA* and *CSCM*, *CSNCS* investigates the unethical multiple services privacy risk in terms of noise utilisation.

Title: Common Set based Noise Cooperation Strategy
Input: the queues of real service requests for every service process are
$\{Q^1{}_R, Q^2{}_R, ..., Q^i{}_R, ..., Q^n{}_R\}$
Output: the queues of final service requests for every service process are
$\{Q^1{}_S, Q^2{}_S, ..., Q^i{}_S, ..., Q^n{}_S\}$

Step 1: Collect all initial service request sets from every service process
Collect all service request queues from every service process in past time:
$\{Q^1{}_R, Q^2{}_R, ..., Q^i{}_R, ..., Q^n{}_R\}$;
Get the initial service request sets from these service requests queues:
$\{Q^1, Q^2, ..., Q^i, ..., Q^n\}$;

Step 2: Compute the initial common noise set based on all noise generation strategies in single noise obfuscation processes
Compute all noise injection intensities based on noise generation strategies in single noise obfuscation processes: $\{\varepsilon_1, \varepsilon_2, ..., \varepsilon_n\}$ by formula 8-5: $\varepsilon_i = NGS(Q^i{}_R, Strategy_i)$;
Generate the initial common noise set based on the initial service request sets from these service requests queues from Algorithm 8-1: $d^i_j = f_i(q^i_j), \forall j \in [1, ni]$

Step 3: Compute all initial noise request sets for noise generation strategies in single noise obfuscation processes
Compute all initial noise request sets for noise generation strategies in single noise obfuscation processes by formula 8-12:
$$Q^i_e = f_i^{-1}(D') + f_i^{-1}(D_{plus}) = Q'^i + Q^i_{plus}$$

Step 4: Execute all noise generation strategies in single noise obfuscation processes
For every noise generation strategy, generate a noise N by the noise request set Q^i ;
Inject N into $Q^i{}_R$ on the probability of ε_i to get $Q^i{}_S$ for every service process;
Update service request queues for every service process.

Step 5: Compute the updated noise set based on the run-time service request sets
Collect and record all run-time service request queues:
$\{Q^1{}_R, Q^2{}_R, ..., Q^i{}_R, ..., Q^n{}_R\}$;
Get the run-time service request sets from these service requests queues:
$\{Q^1, Q^2, ..., Q^i, ..., Q^n\}$;
Compute all run-time noise injection intensities based on noise generation strategies:
$\{\varepsilon_1, \varepsilon_2, ..., \varepsilon_n\}$;
Generate the updated common noise set by Algorithm 8-1:
$d^i_j = f_i(q^i_j), \forall j \in [1, ni]$ and $D_e = D' + D_{plus}$

Step 6: Compute all updated noise request sets for noise generation strategies in single noise obfuscation processes
Compute all updated noise request sets by formula 8-12:
$$Q^i_e = f_i^{-1}(D') + f_i^{-1}(D_{plus}) = Q'^i + Q^i_{plus}$$;

Goto **Step 4**.

Algorithm 8-2 *CSNCS*: Common Set based Noise Cooperation Strategy

In this algorithm, Step 1 is the beginning step to collect all request queues and sets as past data to support the execution of *CSCA* in Step 2. Then, in Step 3, the result of Step 2—common set D_e supports every single noise obfuscation process. After that, all noise generation strategies in these single noise obfuscation processes are executed and final service request queues are expressed to service providers, based on the noise request sets. In Step 4, noise service requests generated by noise generation strategies are the tool to obtain final service request queues as results of noise obfuscation on privacy protection. Step 5 is the updating step to common set D_e like Step 2 and Step 3. The only difference is service request queues changed from initial data to run-time data. Based on common set D_e, all updated noise request sets are computed in Step 6. After this, Step 4 is executed again as a run-time privacy protection mechanism until the whole noise obfuscation function terminates.

As introduced in Chapter 3, the noise generation component is the central part of single noise obfuscation process. Hence, noise generation strategies are the main concern to support our novel noise utilisation strategy in this chapter. And a noise pre-processing strategy can be executed with a noise generation strategy as pre-processing. That is why we did not discuss noise pre-processing in this noise utilisation strategy, although we mentioned the single noise obfuscation process which includes both the noise pre-processing component and the noise generation component.

In summary, the key part of *CSNCS* is the building and updating of common set D_e. We present *CSCM* and *CSCA* to summarise the major part of this. With common set D_e, customers can withstand the unethical multiple services risk. In the next section, we will illustrate that *CSNCS* improves the effectiveness of privacy protection on noise obfuscation with the efficiency promotion by simulation.

8.5 Simulation and Evaluation

In this section, we investigate and evaluate *CSNCS* by simulation. *CSNCS* executes and utilises single noise obfuscation processes by common set D_e. As a result, in the

simulation process, the evaluation of this strategy focuses on its influence on these single noise obfuscation processes in all aspects, especially on noise generation strategies. Three kinds of noise generation strategies have been utilised to operate these noise obfuscation processes based on [77, 42, 43], including Chapter 5 and Chapter 6.

Similar to preceding chapters, we use SwinCloud [93] as the simulation environments to evaluate *CSNCS*. We use one node as a customer to send service requests and apply noise obfuscation. And other ten nodes in this cloud environment are utilised as ten services from $Service_1$ to $Service_{10}$ to receive service requests. These service processes between the customer and each service operate single noise obfuscation processes with noise generation strategies to protect customer privacy. Besides, the last service $Service_{10}$ plays the role of executing the unethical multiple services privacy risk in this chapter. Other nine services send their request sets to this service, and the effectiveness of *CSNCS* on privacy protection can be obtained by $Service_{10}$.

We use *Obfuscation Level* to denote the effectiveness of privacy protection on noise obfuscation in this chapter. It is the size of noise request set. In other words, it means that after noise obfuscation, how many possible requests have about the same probabilities. It is clear that if the *Obfuscation Level* is high, unethical services have to obtain the real service request from a big set. That means they have a low possibility to get the real one, and the effectiveness of privacy protection is high. From Section 8.3, the setting of $epp - m'$ is very important to the simulation process of *CSNCS*. In this process, we set it as the mean of all obfuscation levels from all single noise obfuscation processes at that time.

Firstly, we investigate the unethical multiple services privacy risk in this chapter. In Figure 8-3, ten services operate ten single noise obfuscation processes with noise generation strategies to protect privacy. These ten dash lines denote ten different obfuscation level of each single noise obfuscation process. Depended on different service processes, these ten dash lines increase gradually. The only solid line depicts the obfuscation level of the whole privacy protection under this unethical multiple services risk. Obviously, the solid line is lower than every dash line. Hence, the unethical multiple services privacy risk can decrease the effectiveness of privacy

protection on noise obfuscation significantly.

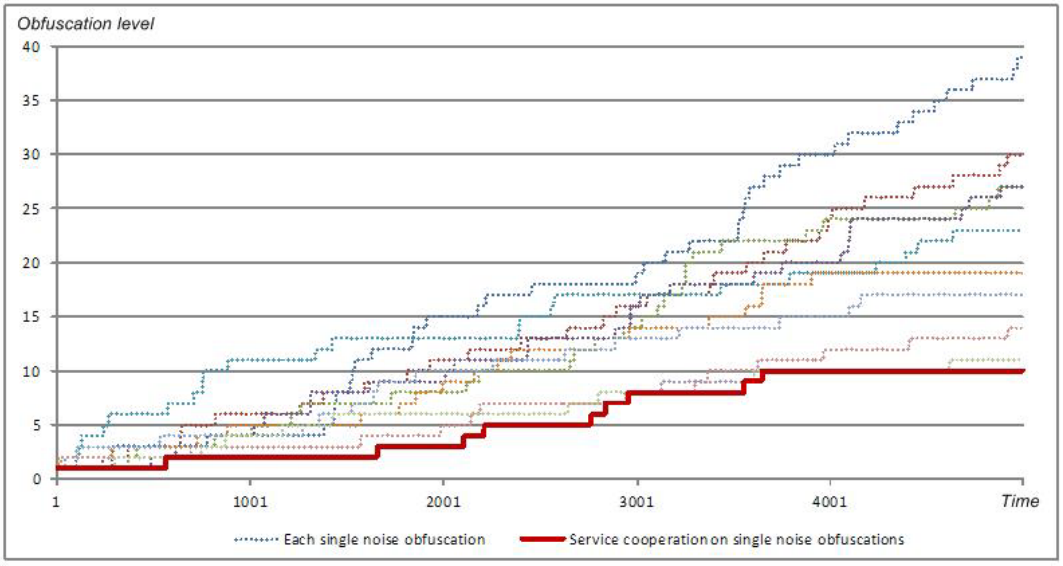


Figure 8-3 Serious privacy risk under the unethical multiple services

Secondly, the improvement on the effectiveness of noise obfuscation on privacy protection is the major evaluation part of *CSNCS*. Under this unethical multiple services risk, the effectiveness of privacy protection can decrease significantly. To address this, *CSNCS* can be operated as a noise utilisation strategy to utilise single noise obfuscation processes with noise generation strategies, and aid them to withstand this risk. Hence, to evaluate our strategy, we compare two situations: one is noise obfuscation functions' operation without *CSNCS*; another is the same noise obfuscation functions' operation with *CSNCS*. In Figures 8-4 and 8-5, we compare these two from the perspective of effectiveness of privacy protection and the cost on noise requests, respectively.

In Figure 8-4, we can find out that *CSNCS* can improve the effectiveness of privacy protection on noise obfuscation significantly, compared to non-*CSNCS*. With time passing, *CSNCS* can utilise noise obfuscation to get a larger and larger dominance in the effectiveness of privacy protection. At the beginning of the simulation, both of obfuscation levels are the same. From time 1000 to 5000, the obfuscation level of our *CSNCS* is about 3 times than noise obfuscations without *CSNCS*. Hence, the improvement on privacy protection is significant.

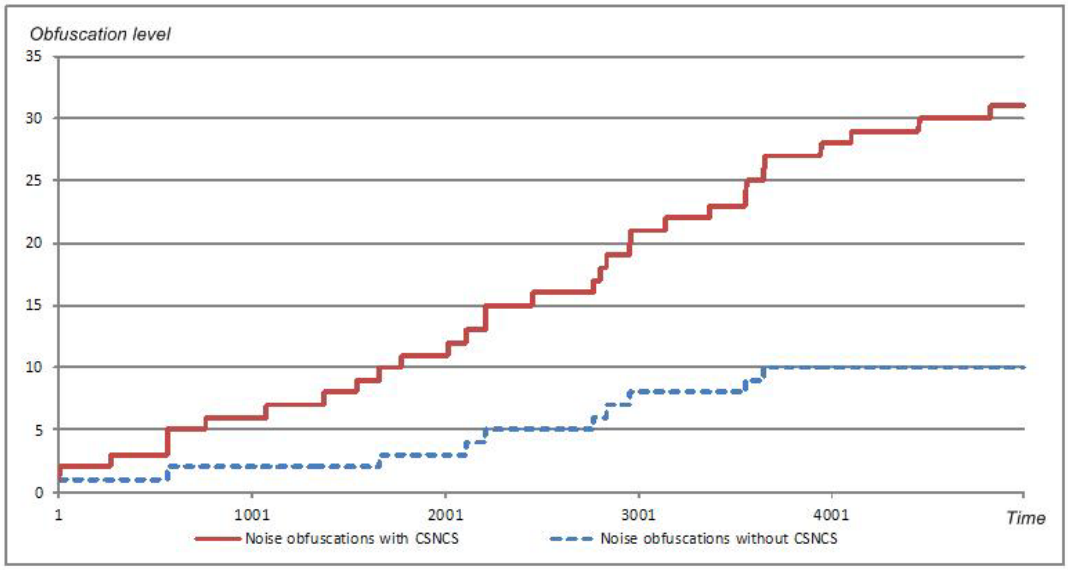


Figure 8-4 Obfuscation level comparison

Finally, we compare the cost on noise service requests between noise obfuscation processes with and without our novel *CSNCS*. The cost on real service request cannot be modified by noise obfuscation. In noise obfuscation, the only thing that can be controlled is reducing noise service requests. Thus, the volume of noise requests denotes this cost.

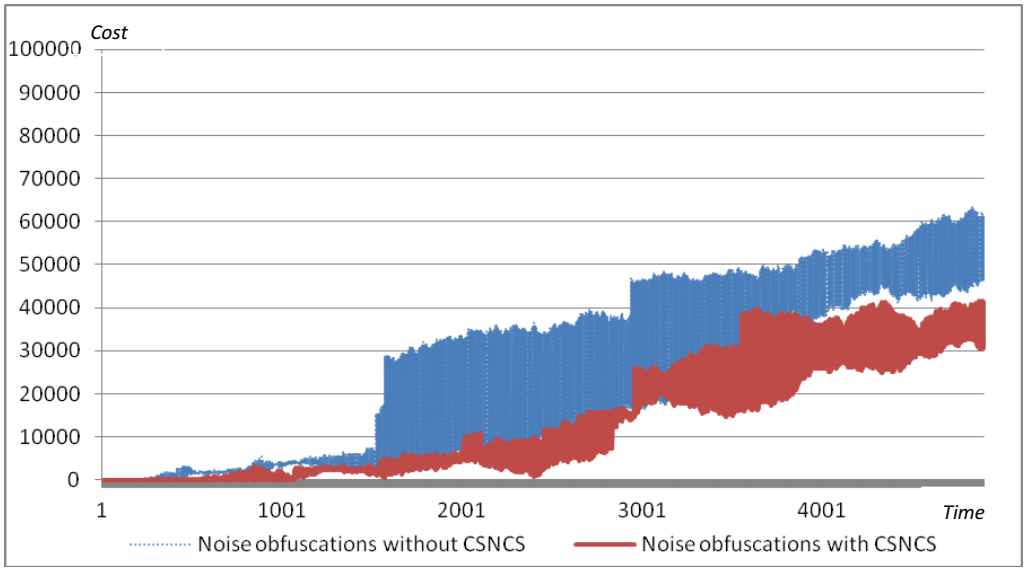


Figure 8-5 Cost comparison

In Figure 8-5, we can find out that *CSNCS* does not increase the cost on noise service requests, compared to noise obfuscations without *CSNCS*. The cost of noise obfuscations with *CSNCS* is lower than the cost of noise obfuscations without *CSNCS*, and the percentage of *CSNCS*'s cost on non-*CSNCS*'s cost fluctuates in the

range from 1/3 to 4/5. Because *CSNCS* guides all noise generation request sets of every single noise obfuscation process with noise generation strategies, and these single noise obfuscation processes become more targeted than before by omitting 'useless' noise requests.

In summary, from the simulation process above, the unethical multiple services privacy risk impacts noise obfuscation notably. And our novel *CSNCS* can withstand this risk: as a noise utilisation strategy, *CSNCS* can improve the effectiveness of privacy protection on noise obfuscation significantly in the unethical multiple services case, based on single noise obfuscation processes with noise generation strategies. Besides, *CSNCS* can decrease the cost on noise requests considerably, too.

8.6 Summary

In this chapter, we presented our novel Common Set based Noise Cooperation Strategy (*CSNCS*) for privacy protection in cloud computing as a noise utilisation strategy to be executed in the unethical multiple services case. As introduced in Chapter 3, the noise utilisation function is the last step of the whole noise obfuscation function for privacy protection in cloud computing. In this step, the multiple services scenario is focused. In the previous single service scenario, noise pre-processing and noise generation can be utilised to fulfil noise obfuscation functions, and their results—noise data can be utilised directly to execute the whole noise obfuscation function. But in multiple services scenarios, some new privacy concerns and risks may threat these noise pre-processing and noise generation functions. In other words, only single noise obfuscation processes (noise pre-processing strategies and noise generation strategies) cannot deal with these privacy concerns and risks in the multiple services scenarios. That is why we discuss noise utilisation in Chapters 7 and 8.

The noise utilisation component considers multiple services in terms of their relations, and guides single noise obfuscation processes together to address these privacy concerns and risks in multiple services scenarios. On one hand, in Chapter 7, we investigated the ethical multiple services case where noise utilisation focuses on the cooperation of ethical services. A novel correlation based noise injection strategy

was presented to utilise single noise obfuscation processes under this case. On the other hand, in this chapter, we discussed the unethical multiple services case where noise utilisation has to consider the cooperation of unethical services which may jeopardise whole noise obfuscation functions. That is the main goal of this chapter.

Briefly, in this chapter, we developed a novel Common Set based Noise Cooperation Strategy (*CSNCS*) for privacy protection in cloud computing. In this strategy, we considered noise generation requests sets from every single noise obfuscation process, and presented the common set to combine and manage them. Based on this common set's efficient creation, this strategy was presented to guide and modify every noise generation set in single noise obfuscation processes. Hence, no matter whether all service providers are unethical and deduce customer's privacy together or not, this strategy can protect customer privacy by concealing real service requests in a reasonable number of noise ones. The simulation evaluation demonstrated that our strategy could cope with this unethical multiple services privacy risk discussed before, i.e. significantly improve the effectiveness of cloud privacy protection on noise obfuscation in the unethical multiple services case. Besides, the efficiency of privacy protection can be improved by our strategy, too.

In future, based on *CSNCS*, we plan to further investigate how to protect customer privacy in the scenario where these unethical service providers may collaborate with each other to deduce customer privacy under some complex "trust" relations.

Chapter 9
Conclusions and Future Work

The technical details for our novel noise obfuscation for privacy protection in cloud computing have all been addressed in previous chapters. In this chapter, we present an overview of the whole thesis. This chapter is organised as follows. Section 9.1 summarises the content of the whole thesis. Section 9.2 outlines the main contributions of this thesis. Finally, Section 9.3 points out the future work.

9.1 Summary of This Thesis

The research objective described in this thesis is to investigate the novel noise obfuscation model for privacy protection in cloud computing, including noise pre-processing component, noise generation component and noise utilisation component. Accordingly, different noise pre-processing strategies, noise generation strategies and noise utilisation strategies have been discussed to obtain an effective and efficient privacy protection on noise obfuscation in cloud computing. The thesis was organised as follows:

- Chapter 1 introduced some key privacy challenges in cloud computing. Moreover, the noise obfuscation approach for privacy protection in cloud computing was discussed to be the main topic of this thesis in this chapter. Chapter 1 also described the structure of this thesis.

- Chapter 2 overviewed general related work on cloud privacy protection. Specifically, this chapter classified current work into service side and client side. In this thesis, we emphasised cloud privacy protection at client side due to the openness and virtualisation features of cloud computing. In this regard, the noise obfuscation approach can be investigated as a promising tool to give customers and enterprises confidence in cloud computing by protecting privacy automatically at client side. Besides, some other supporting work, including time-series analysis, association analysis and so on, was introduced to support these technical strategies presented in succeeding chapters.

- Chapter 3 presented the overview of our novel noise obfuscation for privacy protection in cloud computing—our novel noise obfuscation model. This noise obfuscation model abstracts our novel noise obfuscation approach by withstanding different types of privacy risks and concerns in different steps of noise obfuscation. Hence, in order to give noise obfuscation a comprehensive consideration for cloud computing, our model is composed of three main components including noise pre-processing component, noise generation component and noise utilisation component, which were further illustrated in separate chapters.

  The main technical details of the noise obfuscation model were presented separately in three parts, including noise pre-processing component (consisting of Chapter 4), noise generation component (consisting of Chapter 5 and Chapter 6) and noise utilisation component (consisting of Chapter 7 and Chapter 8).

- Chapter 4 presented a novel privacy-leakage-tolerance based noise enhancing strategy for privacy protection in cloud computing as a noise pre-processing strategy to execute the noise pre-processing component. Specifically, in the noise pre-processing component, the noise pre-processing strategy was proposed based on a customer-set privacy-leakage-tolerance to deal with the concern about linking customers' privacy requirements and noise obfuscations. Briefly, this strategy can improve the efficiency of cloud privacy protection on noise

obfuscation in terms of noise pre-processing, which was demonstrated by the simulation evaluation.

- Chapter 5 presented a novel time-series pattern based noise generation strategy to execute the noise generation component. As a serious privacy risk, these fluctuations of occurrence probabilities can threat existing noise generations to break noise obfuscation in terms of privacy protection. Hence, this strategy abstracts time-series patterns from past service data to model these fluctuations. Then, based on these time-series patterns, future occurrence probabilities can be generated by current occurrence probabilities, and these fluctuations can be forecasted. After this, the noise generation strategy executes the noise generation process to conceal these fluctuations. Hence, malicious service providers are hard to break noise obfuscation by these fluctuations. Besides, we demonstrated that this novel strategy improved the effectiveness of cloud privacy protection on noise obfuscation in terms of probability fluctuation by the simulation evaluation.

- Chapter 6 presented a novel association probability based noise generation strategy to execute the noise generation component, too. For noise generation, association probabilities of service data can be viewed as the private information, and malicious service providers try to break noise obfuscation to obtain it. To withstand this privacy risk, association probabilities among past service data should be analysed and concealed in noise generation processes. Accordingly, to conceal these association probabilities, this novel noise generation function can generate noise data effectively by the association probability model. Briefly, this novel association probability based noise generation strategy can improve the effectiveness of cloud privacy protection on noise obfuscation in terms of association probability, which was demonstrated by the simulation evaluation.

  Besides, in Chapters 5 and 6, these two serious privacy risks can represent two main ways to break noise generation for privacy attackers: using external and internal features of service data. Noise obfuscation has to analyse these risks as needed to protect privacy under specific conditions.

- Chapter 7 presented a novel correlation based noise injection strategy as the noise utilisation strategy for the ethical multiple services case. As a privacy concern, ethical multiple services can cooperate together with noise obfuscation to enhance privacy protection on cloud. Based on the correlation model and noise injection architecture in cloud computing, the novel correlation based noise injection strategy can be operated to link single noise generation processes together, including noise pre-processing and noise generation, to give customer privacy a comprehensive privacy protection. The simulation experiments demonstrated that this strategy improved the effectiveness of cloud privacy protection on noise obfuscation for the ethical multiple services case in terms of noise utilisation.

- Chapter 8 presented a novel common set based noise cooperation strategy as the noise utilisation strategy for the unethical multiple services case. Considering that unethical multiple services may cooperate together to break existing noise obfuscations, we discussed the noise cooperation model to analyse this privacy risk for noise obfuscation in the unethical multiple services case. Then, we presented the common set creation model for noise obfuscation as the key issue of this noise utilisation strategy. Furthermore, the novel common set based noise cooperation strategy was proposed to address this unethical multiple services risk in terms of noise utilisation. The experimental results demonstrated that this strategy improved the effectiveness of cloud privacy protection on noise obfuscation for the unethical multiple services case in terms of noise utilisation.

In summary, based on all chapters, we can conclude that with the research results in this thesis, our novel noise obfuscation, as one promising privacy protection approach, can improve effectiveness of privacy protection in cloud computing with reasonable efficiency in pay-as-you-go cloud computing.

9.2 Contributions of This Thesis

The significance of this research is that we have designed a novel and comprehensive noise obfuscation model which provides effective and efficient

privacy protection in cloud computing. Specifically, as a promising privacy protection approach, our model addresses some limitations in existing noise obfuscation strategies and/or methods. The detailed analysis is conducted for each component of our noise obfuscation model, including noise pre-processing component, noise generation component and noise utilisation component. Based on the analysis, a series of novel strategies in these components' functions, such as noise pre-processing strategies, noise generation strategies and noise utilisation strategies, have been proposed and developed. Corresponding comparisons and quantitative evaluations have shown that these innovative strategies obtain great performances in the effectiveness of privacy protection on noise obfuscation in cloud computing. That means the novel noise obfuscation model can be utilised as an effective tool to protect privacy in cloud computing. Besides, in the context of cloud economy, any resources consumed must be paid. As a privacy protection approach in cloud computing, the cost of noise data is a key issue to be considered. All of these strategies have taken the efficiency of privacy protection on noise obfuscation as their internal parts into consideration. Therefore, the research in this thesis will eventually improve overall effectiveness and efficiency of privacy protection on noise obfuscation in cloud computing. In other words, by deploying our innovative model and strategies, noise obfuscation for privacy protection will be able to better support open and virtualised cloud environments, from the perspective of both effectiveness and efficiency.

In particular, the major contributions of this thesis are:

1) A novel noise obfuscation model for privacy protection in cloud computing

In cloud environments, a comprehensive consideration on a privacy protection approach is necessary for noise obfuscation executed. This thesis has proposed a novel noise obfuscation model with the whole procedure for noise obfuscation. At each step of the procedure in the model, this thesis has discussed the basic functions, and analysed the privacy risks and concerns in existing noise obfuscations. At the first step—noise pre-processing component, the thesis firstly gives cloud customers, who are the users of noise obfuscation for privacy protection, the capability to propose privacy-leakage-tolerance as the privacy requirement to guide noise obfuscation functions. These customers propose privacy protection standards, but do

not control specific noise obfuscation processes directly. At the second step—noise generation component, as a core function of the whole noise obfuscation function, the thesis has significantly improved noise obfuscation to address some serious privacy risks on existing noise obfuscations in terms of noise generation, such as the probability fluctuation privacy risk and the association analysis privacy risk. Besides, the previous two components can combine together to be a single noise obfuscation process which focuses on the noise obfuscation approach in the single service scenario. At the last step—the noise utilisation component, based on managing these previous single noise obfuscation processes, the thesis has firstly analysed how to effectively utilise the noise obfuscation approach in the multiple services scenario, regardless the ethical or unethical case.

2) A novel noise pre-processing strategy by the privacy-leakage-tolerance

Based on the problem analysis of conventional noise obfuscation, a novel noise pre-processing strategy has been proposed to manage noise obfuscation by customer-set requirements on privacy protection—privacy-leakage-tolerance in the single service scenario. The theoretical investigation and experimental results have shown that this noise pre-processing strategy can build a suitable bridge between cloud customers and noise obfuscation functions, and improve the efficiency of privacy protection on noise obfuscation by decreasing useless noise data items before noise generation, with a reasonable effectiveness.

3) A novel noise generation strategy for probability fluctuations' concealing

To withstand the serious privacy risk on occurrence probabilities' fluctuations, a novel time-series pattern based noise generation strategy has been presented in this thesis. By analysing how the fluctuations can break existing noise obfuscations and obtain customer privacy, time-series patterns can be used to abstract past occurrence probabilities' changing, especially these fluctuations. Hence, these time-series patterns can help to forecast future fluctuations and guide noise generation to conceal these fluctuations. The effectiveness of privacy protection on noise obfuscation can be improved significantly in terms of noise generation with a reasonable cost under this serious privacy risk, which has been demonstrated by the simulation evaluation.

4) A novel noise generation strategy for association information's concealing

To deal with the privacy risk about the association analysis on past service data, a novel association probability based noise generation strategy has been presented in this thesis. In this strategy, the novel association probability model has been presented to describe the private information among service data, and guide noise generation to conceal these association probabilities. Afterwards, based on noise data generated by this strategy with the association probability model, the association probabilities privacy risk can be addressed. The experimental results illustrated that this strategy can improve the effectiveness of privacy protection on noise obfuscation in terms of noise generation, under this association analysis privacy risk.

5) A novel noise utilisation strategy in the ethical multiple services case

To consider how to utilise noise obfuscation effectively in the ethical multiple services case, a novel correlation based noise injection strategy has been presented to address this privacy concern. Based on the noise injection architecture in cloud computing, we linked different single noise obfuscation processes together to pursue a comprehensive consideration on privacy protection in the ethical multiple cloud services case. Therefore, the novel noise injection strategy can realise this consideration and utilise single noise obfuscation processes to be more effective in the ethical multiple services case. The experimental results expressed that our strategy can significantly improve the effectiveness of privacy protection on noise obfuscation in the ethical multiple services case in terms of noise utilisation.

6) A novel noise utilisation strategy in the unethical multiple services case

In the unethical multiple services case, to withstand the privacy risk under unethical services' sharing information to break existing noise obfuscations, a novel common set based noise cooperation strategy has been presented. Specifically, we designed a noise cooperation model to abstract the privacy risk. Based on this model, the noise utilisation strategy can be proposed to utilise single noise obfuscation processes to withstand this privacy risk by the common set. The simulation evaluation has shown that this strategy can improve the effectiveness of privacy

protection on noise obfuscation in the unethical multiple services case in terms of noise utilisation.

9.3 Future Work

This thesis has proposed a novel noise obfuscation model for privacy protection in cloud computing. The future work will focus on how to further improve the overall effectiveness and efficiency of privacy protection for this model with different types of strategies. Specifically, there are five aspects which can be further investigated in future:

1) Noise pre-processing strategy: our privacy-leakage-tolerance based noise enhancing strategy focuses on building bridges between customers' requirements and noise obfuscations by the privacy-leakage-tolerance. In future, we will investigate the adaptation of this noise pre-processing strategy for other types of customers' requirements, not only the privacy-leakage-tolerance in Chapter 4, such as the ratio between the privacy-leakage-tolerance and the cost on privacy protection by noise obfuscation.

2) Noise generation strategy for probability fluctuations' concealing: under the time-series pattern based probability forecasting algorithm, our time-series pattern based noise generation strategy can significantly improve the effectiveness of privacy protection on noise obfuscation to withstand fluctuations of occurrence probabilities. In future, based on the existing static creation of time series patterns in Chapter 5, the dynamic creation of time-series patterns will be investigated to further improve the adaptability of noise generation.

3) Noise generation strategy for association probabilities' concealing: our association probability based noise generation strategy considered the association probability model to withstand some privacy attackers who were interested in these association analysis's results. In future, the association

probability model can be improved to describe some complex association rules as customer privacy, not only the direct consequential relation in Chapter 6, but also some indirect relations. Hence, these association probabilities will denote more types of private information to be protected by noise obfuscation.

4) Noise utilisation strategy for ethical multiple services case: in the noise utilisation component, the correlation based noise injection strategy has been presented to link several single noise obfuscation processes together to pursue a better effectiveness of privacy protection. In future, the correlation model can be further modified by considering quantified correlations to replace the qualified correlations in Chapter 7.

5) Noise utilisation strategy for unethical multiple services case: in the noise utilisation component, the common set based noise cooperation strategy has been proposed to withstand unethical services which cooperate together to share information and break noise obfuscation. In future, we plan to further investigate how to protect customer privacy in the scenario where these unethical service providers may share customers' requests sets with each other partially under some complex "trust" relations, compared to the scenario that these unethical service providers share all request sets together in Chapter 8.

Bibliography

[1] Aaron M. Johnson, Paul Syverson, Roger Dingledine, and Nick Mathewson, "Trust-based Anonymous Communication: Adversary Models and Routing Algorithms," presented at the 18th ACM Conference on Computer and Communications Security (CCS'11), pp. 175-186, Chicago, Illinois, USA, October 17-21, 2011.

[2] Aaron Weiss, "Computing in The Clouds," *ACM Networker,* vol. 11, issue. 4, pp. 16-25, December 2007.

[3] Ahmed M. Azab, Peng Ning, and Xiaolan Zhang, "SICE: A Hardware-Level Strongly Isolated Computing Environment for x86 Multi-core Platforms," presented at the 18th ACM Conference on Computer and Communications Security (CCS' 11), pp. 375-388, Chicago, Illinois, USA, October 17-21, 2011.

[4] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant, "Limiting Privacy Breaches in Privacy Preserving Data Mining," presented at the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2003), pp. 211-222, San Diego, California, USA, June 9 -11, 2003.

[5] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz, "General Constructions for Information-Theoretic Private Information Retrieval," *Journal of Computer System Science,* vol. 71, issue. 2, pp. 213-247, August 2005.

[6] Andrés Rodríguez, José María Carazo, and Oswaldo Trelles, "Mining Association Rules from Biological Databases," *Journal of the American Society for Information Science and Technology,* vol. 56, issue. 5, pp. 493-504, March 2005.

[7] Andrew Chi-Chih Yao, "Protocols for Secure Computations," presented at the 23rd Annual Symposium on Foundations of Computer Science (SFCS' 82), pp. 160-164, Chicago, Illinois, USA, Novenmber 3-5, 1982.

[8] Andrew Chi-Chih Yao, "How to Generate and Exchange Secrets," presented at the 27th Annual Symposium on Foundations of Computer Science (SFCS'86), pp. 162-167, Toronto, Canada, October 27-29, 1986.

[9] Arvind Narayanan and Vitaly Shmatikov, "De-anonymizing Social Networks," presented at the 30th IEEE Symposium on Security and Privacy (SP' 09), pp. 173-187, Oakland, California, USA, May 17-20, 2009.

[10] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim, "Practical Privacy: The SuLQ Framework," presented at the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2005), pp. 128-138, Baltimore, Maryland, USA, June 13-16, 2005.

[11] Bee-Chung Chen, Kristen LeFevre, and Raghu Ramakrishnan, "Privacy Skyline: Privacy with Multidimensional Adversarial Knowledge," presented at the 33rd International Conference on Very Large Data Bases (VLDB'07), pp. 770-781, Vienna, Austria, September 23-27, 2007.

[12] Benjamin C. M. Fung, Ke Wang, Lingyu Wang, and Patrick C. K. Hung, "Privacy-Preserving Data Publishing for Cluster Analysis," *Data & Knowledge Engineering,* vol. 68, issue. 6, pp. 552-575, June 2009.

[13] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu, "Privacy-Preserving Data Publishing: A Survey of Recent Developments," *ACM Computing Surveys,* vol. 42, issue. 4, pp. 1-53, June 2010.

[14] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan, "Private Information Retrieval," *Journal of ACM,* vol. 45, issue. 6, pp. 965-981, November 1998.

[15] Bradley Malin and Latanya Sweeny, "Inferring Genotype from Clinical Phenotype through a Knowledge based Algorithm," presented at the 2002 Pacific Symposium on Biocomputing (PSB' 02) pp. 41-52, Hawaii, USA, January 3-7, 2002.

[16] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom, "Models and Issues in Data Stream Systems," presented at the 21th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2002), pp. 1-16, Madison, Wisconsin, USA, June 3-6, 2002.

[17] Chang Liu, Xuyun Zhang, Jinjun Chen, and Chi Yang, "An Authenticated Key Exchange Scheme for Efficient Security-Aware Scheduling of Scientific Applications in Cloud Computing," presented at the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC' 11), pp. 372-379, Sydney, Australia, December 12-14, 2011.

[18] Claudio A. Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati, and Pierangela Samarati, "An Obfuscation-Based Approach for Protecting Location Privacy," *IEEE Transactions on Dependable and Secure Computing,* vol. 8, issue. 1, pp. 13-27, January 2011.

[19] *US Health Information Portability and Accountability Act,* U. S. Congress, 1996.

[20] Craig Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," presented at the 41st Annual ACM Symposium on Theory of Computing (STOC'09), pp. 169-178, Bethesda, Maryland, USA, May 31 - June 2, 2009.

[21] Craig Gentry and Shai Halevi, "Fully Homomorphic Encryption without Squashing Using Depth-3 Arithmetic Circuits," presented at the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS'2011), pp. 107-109, Palm Springs, California, USA, October 22-25, 2011.

[22] Cynthia Dwork, "A Firm Foundation for Private Data Analysis," *Communications of the ACM,* vol. 54, issue. 1, pp. 86-95, Jaunary 2011.

[23] David E.Bakken, Rupa. Rarameswaran, Douglas M. Blough, Andy A Franz, and Ty J. Palmer, "Data Obfuscation: Anonymity and Aesensitization of Usable Data Sets," *Security & Privacy, IEEE,* vol. 2, issue. 6, pp. 34-41, November/December 2004.

[24] David Goldschlag, Michael Reed, and Paul Syverson, "Onion Routing," *Communications of the ACM,* vol. 42, issue. 2, pp. 39-41, February 1999.

[25] David Kotz, "A Threat Taxonomy for mHealth Privacy," presented at the 3rd International Conference on Communication Systems and Networks (COMSNETS' 11), pp. 1-6, Bangalore, India, January 4-8, 2011.

[26] David Moore, Colleen Shannon, Douglas J. Brown, Geoffrey M. Voelker, and Stefan Savage, "Inferring Internet Denial-of-Service Activity," *ACM Transcation on Computer Systems,* vol. 24, issue. 2, pp. 115-139, May 2006.

[27] David Tancock, Siani Pearson, and Andrew Charlesworth, "A Privacy Impact Assessment Tool for Cloud Computing," presented at the IEEE 2nd International Conference on Cloud Computing Technology and Science (CloudCom 2010), pp. 667-676, Indianapolis, USA, November 30-December 3, 2010.

[28] Deepak Garg, Limin Jia, and Anupam Datta, "Policy Auditing over Incomplete Logs: Theory, Implementation and Applications," presented at the 18th ACM Conference on Computer and Communications Security (CCS'11), pp. 151-162, Chicago, Illinois, USA, October 17-21, 2011.

[29] Dingledine Rogerm, Nick Mathewson, and Paul Syverson, "Tor: The Second-Generation Onion Router," presented at the 13th USENIX Security Symposium, pp. 303-320, San Diego, California, USA, August 9-13, 2004.

[30] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani, "An Online Algorithm for Segmenting Time Series," presented at the 2001 IEEE International Conference on Data Mining (ICDM'01), pp. 289-296, San Jose, California, USA, November 29 - December 2, 2001.

[31] EC2. (2012, December 30, 2012). Available: http://aws.amazon.com/ec2/

[32] Elaine Shi, T-H. Hubert Chan, Eleanor Rieffel, Richard Chow, and Dawn Song, "Privacy-Preserving Aggregation of Time-Series Data," presented at the 18th Annual Network & Distributed System Security Symposium (NDSS 2011), p. 17, San Diego, California, USA, February 6-9, 2011.

[33] Erik-Oliver Blass, Roberto Di Pietro, Refik Molva, and Melek Onen, "PRISM -- Privacy-Preserving Search in MapReduce," *Cryptology ePrint Archive, Report 2011/244, Available at : http://eprint.iacr.org/2011/244.pdf*, 2011.

[34] Ero Balsa, Carmela Troncoso, and Claudia Diaz, "OB-PWS: Obfuscation-Based Private Web Search," presented at the 2012 IEEE Symposium on Security and Privacy (SP' 12), pp. 491-505, San Francisco, USA, May 20-23, 2012.

[35] Etienne Perron, Suhas Diggavi, and Emre Telatar, "On Cooperative Wireless Network Secrecy," presented at the 28th Conference on Computer Communications (IEEE INFOCOM 2009), pp. 1935-1943, Rio de Janeiro, Brazil, April 19-25, 2009.

[36] Eucalyptus. (2012, December 30, 2012). Available: http://www.eucalyptus.com/

[37] European Office of Economic Cooperation and Development. (1980, December 30, 2012). *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. Available:

www.oecd.org/document/18/0,2340,en_2649_34255_1815186_119820_1_1_1,00.html

[38] Florian Kerschbaum, "Automatically Optimizing Secure Computation," presented at the 18th ACM Conference on Computer and Communications Security (CCS' 11), pp. 703-714, Chicago, Illinois, USA, October 17-21, 2011.

[39] Gaofeng Zhang, Xuyun Zhang, Yun Yang, Chang Liu, and Jinjun Chen, "An Association Probability based Noise Generation Strategy for Privacy Protection in Cloud Computing," presented at the 10th International Conference on Service Oriented Computing (ICSOC 2012), pp. 639-647, Shanghai, China, November 12-15, 2012.

[40] Gaofeng Zhang, Yun Yang, Chang Liu, Xuyun Zhang, and Jinjun Chen, "Key Research Issues for Privacy Protection and Preservation in Cloud," presented at the 2012 International Conference on Cloud and Green Computing (CGC2012), pp. 47-54, Xiangtan, Hunan, China, November 1-3, 2012.

[41] Gaofeng Zhang, Yun Yang, Dong Yuan, and Jinjun Chen, "A Trust-based Noise Injection Strategy for Privacy Protection in Cloud Computing," *Software: Practice and Experience,* vol. 42, issue. 4, pp. 431-445, April 2012.

[42] Gaofeng Zhang, Yun Yang, and Jinjun Chen, "A Histrotical Probability based Noise Generation Strategy for Privacy Protection in Cloud Computing," *Journal of Computer and System Sciences,* vol. 78, issue. 5, pp. 1374-1381, September 2012.

[43] Gaofeng Zhang, Yun Yang, Xiao Liu, and Jinjun Chen, "A Time-Series Pattern based Noise Generation Strategy for Privacy Protection in Cloud Computing," presented at the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 458-465, Ottawa, Canada, May 13-16, 2012.

[44] Hadoop. (2012, Decemeber 30, 2012). Available: http://hadoop.apache.org/

[45] Hadoop Distributed File System. (2012, December 30, 2012). Available: http://hadoop.apache.org/hdfs/

[46] Ian Goldberg, "Improving the Robustness of Private Information Retrieval," presented at the 2007 IEEE Symposium on Security and Privacy (SP' 07), pp. 131-148, Oakland, California, USA, May 20-23, 2007.

[47] IBM. (2010, December 30, 2012). *Cloud*. Available: http://www.ibm.com/cloud-computing/us/en/

[48] Indrajit Roy, Srinath T.V. Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel, "Airavat: Security and Privacy for MapReduce," presented at the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2010), pp. 20-20, San Jose, USA, April 28-30, 2010.

[49] Jakub Szefer, Eric Keller, Ruby B. Lee, and Jennifer Rexford, "Eliminating the Hypervisor Attack Surface for a More Secure Cloud," presented at the 18th ACM Conference on Computer and Communications Security (CCS' 11), pp. 401-412, Chicago, Illinois, USA, October 17-21, 2011.

[50] James W. Gray, "On Introducing Noise into the Bus-Contention Channel," presented at the 1993 IEEE Symposium on Security and Privacy (SP' 93), pp. 90-98, Oakland, California, USA, May 24-26, 1993.

[51] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," presented at the 6th conference on Symposium on Opearting Systems Design & Implementation (OSDI' 04), pp. 137-150, San Francisco, California, USA, December 6-8, 2004.

[52] Jinjun Chen and Yun Yang, "A Taxonomy of Grid Workflow Verification and Validation," *Concurrency and Computation: Practice & Experience,* vol. 20, issue. 4, pp. 347-360, March 2008.

[53] Jun Gao, Jeffrey Xu Yu, Ruoming Jin, Jiashuai Zhou, Tengjiao Wang, and Dongqing Yang, "Neighborhood-Privacy Protected Shortest Distance Computing in Cloud," presented at the 2011 International Conference on Management of Data (SIGMOD '11), pp. 409-420, Athens, Greece, June 12-16, 2011.

[54] Kazuhide Fukushima, Shinsaku Kiyomoto, and Toshiaki Tanaka, "Obfuscation Mechanism in Conjunction with Tamper-Proof Module," presented at the 2009 International Conference on Computational Science and Engineering (CSE' 09), pp. 665-670, Vancouver, Canada, August 29-31, 2009.

[55] Ke Liu, Hai Jin, Jinjun Chen, Xiao Liu, Dong Yuan, and Yun Yang, "A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform," *International Journal of High Performance Computing Applications,* vol. 24, issue. 4, pp. 445-456, November 2010.

[56] Kehuan Zhang, Xiaoyong Zhou, Yangyi Chen, XiaoFeng Wang, and Yaoping Ruan, "Sedic: Privacy-Aware Data Intensive Computing on Hybrid Clouds," presented at the 18th ACM Conference on Computer and Communications Security (CCS' 11), pp. 515-526, Chicago, Illinois, USA, October 17-21, 2011.

[57] Koen Simoens, Pim Tuyls, and Bart Preneel, "Privacy Weaknesses in Biometric Sketches," presented at the 30th IEEE Symposium on Security and Privacy (SP' 09), pp. 188-203, Oakland, California, USA, May 17-20, 2009.

[58] Latanya Sweeney, "K-Anonymity: A Model for Protecting Privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, issue. 5, pp. 557-570, October 2002.

[59] Liang Yan, Chunming Rong, and Gansen Zhao, "Strengthen Cloud Computing Security with Federal Identity Management Using Hierarchical Identity-Based Cryptography," presented at the 1st International Conference on Cloud Computing (CloudCom' 09), pp. 167-177, Beijing, China, December 1-4, 2009.

[60] Lior Malka, "VMCrypt: Modular Software Architecture for Scalable Secure Computation," presented at the 18th ACM Conference on Computer and Communications Security (CCS' 11), pp. 715-724, Chicago, Illinois, USA, October 17-21, 2011.

[61] Marco A. Boschetti, Vittorio Maniezzo, and Matteo Roffilli, "A Fully Distributed Lagrangean Solution for a Peer-to-Peer Overlay Network Design Problem," *INFORMS Journal on Computing,* vol. 23, issue. 1, pp. 90-104, Winter 2011.

[62] Mark D. Ryan, "Cloud Computing Privacy Concerns on Our Doorstep," *Communications of the ACM,* vol. 54, issue. 1, pp. 36-38, January 2011.

[63] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H.Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin,

Ion Stoica, and Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," *Communications of the ACM,* vol. 53, issue. 6, pp. 50-58, April 2010.

[64] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan, "Can Homomorphic Encryption Be Practical," presented at the 3rd ACM Workshop on Cloud Computing Security Workshop (CCSW'2011), pp. 113-124, Chicago, Illinois, USA, October 17-21, 2011.

[65] Nils Gruschka and Meiko Jensen, "Attack Surfaces: A Taxonomy for Attacks on Cloud Services," presented at the IEEE 3rd International Conference on Cloud Computing (CLOUD 2010), pp. 276-279, Miami, Florida, USA, July 5-10, 2010.

[66] Openstack. (2012, December 30, 2012). Available: http://openstack.org/

[67] Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik, "Countering GATTACA: Efficient and Secure Testing of Fully-Sequenced Human Genomes," presented at the 18th ACM Conference on Computer and Communications Security (CCS' 11), pp. 691-702, Chicago, Illinois, USA, October 17-21, 2011.

[68] Qian Liu, Chuliang Weng, Minglu Li, and Yuan Luo, "An In-VM Measuring Framework for Increasing Virtual Machine Security in Clouds," *Security & Privacy, IEEE,* vol. 8, issue. 6, pp. 56-62, November 2010.

[69] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," *IEEE Transactions on Parallel and Distributed Systems,* vol. 22, issue. 5, pp. 847-859, May 2011.

[70] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as The 5th Utility," *Future Generation Computer Systems,* vol. 25, issue. 6, pp. 599-616, June 2009.

[71] Rakesh Agrawal and Ramakrishnan Srikant, "Privacy-Preserving Data Mining," *ACM SIGMOD Record,* vol. 29, issue. 2, pp. 439-450, June 2000.

[72] Ran Canetti, Ben Riva, and Guy N. Rothblum, "Practical Delegation of Computation using Multiple Servers," presented at the 18th ACM

Conference on Computer and Communications Security (CCS'11), pp. 445-454, Chicago, Illinois, USA, October 17-21, 2011.

[73] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," presented at the 13th ACM Conference on Computer and Communications Security (CCS' 06), pp. 79-88, Alexandria, Virginia, USA, October 30 - November 3, 2006.

[74] Ricaedo Neisse, Dominil Holling, and Alexander Pretschner, "Implementing Trust in Cloud Infrastructures," presented at the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2011), pp. 524-533, New Beach, California, USA, May 23-26, 2011.

[75] Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka, and Jesus Molina, "Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control," presented at the 2009 ACM Workshop on Cloud Computing Security (CCSW 2009), pp. 85-90, Chicago, Illinois, USA, November 9-13, 2009.

[76] Ryan K. L. Ko, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, Qianhui Liang, and Bu Sung Lee, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing," presented at the 2011 IEEE World Congress on Services (SERVICES), pp. 584-588, Washington, DC, USA, July 4-9, 2011.

[77] Shaozhi Ye, Felix Wu, Raju Pandey, and Hao Chen, "Noise Injection for Search Privacy Protection," presented at the 2009 International Conference on Computational Science and Engineering (CSE' 09), pp. 1-8, Vancouver, Canada, August 29-31, 2009.

[78] Shui Yu, Guofeng Zhao, Wanchun Dou, and Simon James, "Predicted Packet Padding for Anonymous Web Browsing Against Traffic Analysis Attacks," *IEEE Transactions on Information Forensics and Security,* vol. 7, issue. 4, pp. 1381-1393, August 2012.

[79] Siani Pearson, Yun Shen, and Miranda Mowbray, "A Privacy Manager for Cloud Computing," presented at the 1st International Conference on Cloud Computing (CloudCom 2009), pp. 90-106, Beijing, China, December 1-4, 2009.

[80] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith, "Composition Attacks and Auxiliary Information in Data Privacy," presented at the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD' 08), pp. 265-273, Las Vegas, Nevada, USA, August 24-27, 2008.

[81] Stefan Sackmann, Jens Strüker, and Rafael Accorsi, "Personalization in Privacy-Aware Highly Dynamic Systems," *Communication of the ACM,* vol. 49, issue. 9, pp. 32-38, September 2006.

[82] Stephen McLaughlin, Patrick McDaniel, and William Aiello, "Protecting Consumer Privacy from Electric Load Monitoring," presented at the 18th ACM Conference on Computer and Communications Security (CCS' 11), pp. 87-98, Chicago, Illinois, USA, October 17-21, 2011.

[83] Sven Bugiel, Stefan Nürnberger, Thomas Pöppelmann, Ahmad-Reza Sadeghi, and Thomas Schneider, "AmazonIA: When Elasticity Snaps Back," presented at the 18th ACM Conference on Computer and Communications Security (CCS'11), pp. 389-400, Chicago, Illinois, USA, October 17-21, 2011.

[84] SwinDeW-G Team. (2008, December 30, 2012). *System Architecture of SwinDeW-G*. Available: www.swinflow.org/docs/System_Architecture.pdf

[85] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," presented at the 16th ACM Conference on Computer and Communications Security (CCS' 09), pp. 199-212, Chicago, Illinois, USA, November 9-13, 2009.

[86] Vibhor Rastogi, Dan Suciu, and Sungho Hong, "The Boundary Between Privacy and UItility in Data Publishing," presented at the 33rd International Conference on Very Large Data Bases (VLDB 2007), pp. 531-542, Vienna, Austria, September 23-27, 2007.

[87] Vita Bortnikov, Gregory Chockler, Alexey Roytman, and Mike Spreitzer, "Bulletin Board: A Scalable and Robust Eventually Consistent Shared Memory Over A Peer-to-Peer Overlay," *SIGOPS Operating Systems Review,* vol. 44, issue. 2, pp. 64-70, April 2010.

[88] VMWare. (2012, December 30th, 2012). Available: http://www.vmware.com/

[89] Wassim Itani, Ayman Kayssi, and Ali Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures," presented at the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC '09), pp. 711-716, Chengdu, China, December 12-14, 2009.

[90] Wenjuan Li and Lingdi Ping, "Trust Model to Enhance Security and Interoperability of Cloud Environment," presented at the 1st International Conference on Cloud Computing (CLOUD 2009), pp. 69-79, Beijing, China, December 1-4, 2009.

[91] Xiao Liu, Jinjun Chen, Ke Liu, and Yun Yang, "Forecasting Duration Intervals of Scientific Workflow Activities Based on Time-Series Patterns," presented at the IEEE Fourth International Conference on e-Science (e-Science 2008), pp. 23-30, Indianapolis, Indiana, USA, December 7-12, 2008.

[92] Xiao Liu, Dong Yuan, Gaofeng Zhang, Jinjun Chen, and Yun Yang, "SwinDeW-C: A Peer-to-Peer Based Cloud Workflow System," in *Handbook of Cloud Computing*, Borko Furht and Armando Escalante, Eds., ed: Springer 2010, pp. 309-332.

[93] Xiao Liu, Dong Yuan, Gaofeng Zhang, Wenhao Li, Dahai Cao, Qiang He, Jinjun Chen, and Yun Yang, *The Design of Cloud Workflow Systems: Architecture, Functionality and Quality of Service* SpringerBriefs, 2012.

[94] Xin Huang, Yin He, Yifan Hou, Lisi Li, Lan Sun, Sina Zhang, Yang Jiang, and Tingting Zhang, "Privacy of Value-Added Context-Aware Service Cloud," presented at the 1st International Conference on Cloud Computing (CloudCom' 09), pp. 547-552, Beijing, China, December 1-4, 2009.

[95] Yan Huang, David Evans, Jonathan Katz, and Lior Malka, "Faster Secure Two-Party Computation Using Garbled Circuits," presented at the 20th USENIX Security Symposium, pp. 539-554, San Francisco, USA, August 8-12, 2011.

[96] Yan Wang, Kwei-Jay Lin, Duncan S. Wong, and Vijay Varadharajan, "Trust Management Towards Service-Oriented Applications," *Service Oriented*

Computing and Applications Journal, vol. 3, issue. 2, pp. 129-146, January 2009.

[97] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, and Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," *IEEE Transactions on Parallel and Distributed Systems,* vol. 23, issue. 12, pp. 2231-2244, December 2012.

[98] Yun Yang, Ke Liu, Jinjun Chen, Joel Lignier, and Hai Jin, "Peer-to-Peer Based Grid Workflow Runtime Environment of SwinDeW-G," presented at the Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007), pp. 51-58, Bangalore, India, December 10-13, 2007.

[99] Yun Yang, Ke Liu, Jinjun Chen, Xiao Liu, Dong Yuan, and Hai Jin, "An Algorithm in SwinDeW-C for Scheduling Transaction-Intensive Cost-Constrained Cloud Workflows," presented at the IEEE Fourth International Conference on eScience (e-Science 2008), pp. 374-375, Indianapolis, USA, December 7-12, 2008.