2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications

A Privacy-Leakage-Tolerance based Noise Enhancing Strategy for Privacy Protection in Cloud Computing

Gaofeng Zhang[1], Yun Yang[1],
[1]Faculty of Information and Communication Technologies
Swinburne University of Technology
Hawthorn, Melbourne, Australia 3122
{gzhang, yyang}@swin.edu.au

Jinjun Chen[2]
[2] School of Systems, Management and Leadership
Faculty of Engineering and Information Technology
University of Technology, Sydney
Broadway, NSW, Australia 2007
Jinjun.Chen@uts.edu.au

***Abstract*—Cloud computing promises a service-oriented environment where customers can utilise IT services in a pay-as-you-go fashion while saving huge capital investments on their own IT infrastructures. Due to the openness, malicious service providers may exist in these environments. Some of these service providers could record service data in cloud service processes about a customer and then collectively deduce the customer's private information without authorisation. Noise obfuscation is an effective approach in this regard by utilising noise data. For example, it can generate and inject noise service requests into real customer service requests so that service providers are not able to distinguish which ones are real ones. However, existing typical noise obfuscations do not consider the customer-defined privacy-leakage-tolerance in noise obfuscation processes. Specifically, cloud customers could define a boundary of privacy leakage possibility to require noise obfuscation on privacy protection in cloud computing. In other words, under this boundary—privacy-leakage-tolerance, noise obfuscation could be enhanced by the efficiency improvement on privacy protection, such as reducing noise service requests injected into real ones. So, the customer can obtain a lower cost on noise data in the pay-as-you-go fashion for cloud environments, with a reasonable effectiveness of privacy protection. Therefore, to address this privacy concern, a novel noise enhancing strategy can be presented. We firstly analyse the privacy-leakage-tolerance for cloud customers in terms of noise generation. Then, the creation of a noise generation set can be presented based on the privacy-leakage-tolerance, and the set can guide and enhance existing noise generation strategies by this boundary. Lastly, we present our novel privacy-leakage-tolerance based noise enhancing strategy for privacy protection in cloud computing. The simulation evaluation demonstrates that our strategy can significantly improve the efficiency of privacy protection on existing noise obfuscations in cloud environments.**

***Keywords*—Cloud computing, Privacy protection, Noise obfuscation, Privacy-leakage-tolerance**

I. INTRODUCTION

Generally speaking, cloud computing is a new and promising platform for delivering information infrastructures and resources as IT services in terms of virtualisation [1]. Cloud customers can access, utilise or deploy these services to execute their business jobs in a pay-as-you-go fashion while saving huge capital investments on their own IT environments [2]. However, these customers often have concerns about whether their privacy can be protected when facilitating their IT services in cloud environments since they do not have much control inside cloud [3]. In the worst cases in terms of cloud privacy protection, customers may eventually lose the confidence in and desire to deploy and utilise cloud computing in practice [4]. Therefore, privacy protection is critical as one of the most concerned issues in cloud computing.

In cloud environments, there are many ethical organisations which operate under various regulations and policies by protecting their customers' privacy. Meanwhile, a large number of unethical and unknown service providers may exist in these open and complicated cloud eco-environments. Some of these service providers may collect service data from customers, then analyse and deduce customers' privacy without their permissions.

Besides, for service providers, it is a common phenomenon to collect and analyse their customers' information, like service requests, and so on. They always use this information to analyse customers' behaviours, habits and other information which customers may view it as privacy. This is a powerful way to promote IT services in terms of business market. Most ethical organisations have adequate self-control to use the information inside them under policies and regulations, but someone else may abuse it in immoral ways, especially in open and virtualised cloud environments. Because open and virtualised features make customers hard to distinguish which service providers are unethical, it is impossible to avoid using them in cloud environments. For example, in complex cloud eco-systems, various service providers could cooperate together to pursue powerful and cost-effective services. For cloud customers, this could cause more unknown service providers in cloud service processes than ever before, which they cannot control. Hence, in cloud computing, it is inevitable that customers' information could be collected by unethical service providers to deduce customer privacy.

Therefore, certain technical actions need to be taken to protect customers' privacy automatically at client side without participations from service providers [5]. Based on data obfuscation [6], noise obfuscation is an effective approach in this regard. For instance, it can generate and inject noise service requests into real customer service requests automatically. When final requests' occurrence probabilities are about the same, service providers cannot distinguish which

978-0-7695-5022-0/13 $26.00 © 2013 IEEE
DOI 10.1109/TrustCom.2013.64

ones are real ones based on occurrence probabilities. The key advantage is that this approach does not involve service providers. Hence, it providers a promising approach to protect customer privacy in the scenario of this paper.

To fulfil different requirements on noise obfuscation, currently, different noise generation strategies have been presented [7-9]. For example, a Historical Probability based Noise Generation Strategy (*HPNGS*) has been proposed to improve the efficiency of privacy protection on noise obfuscation based on historical probabilities [8], compared to random noise generation strategy [7]. Besides, a Time-series Pattern based Noise Generation Strategy (*TPNGS*) can deal with fluctuations of occurrence probabilities by forecasting future probabilities based on past time-series patterns [9]. But in general, existing noise obfuscations do not consider and investigate the impact from a customer-defined privacy-leakage-tolerance in terms of noise generation process.

Actually, it is a natural concern that a customer considers and evaluates privacy leakage risk with some boundaries before he/she utilises services in cloud environments, no matter automatically or not. And these customer-defined boundaries or tolerances on the possibility of privacy leakage could be important issues to evaluate and manage noise obfuscation processes. In other words, these specific privacy-leakage-tolerances could give cloud customers specific choices on specific noise obfuscation processes. For instance, a service provider in cloud has 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. Noise obfuscation could use this tolerance to guide the specific noise obfuscation process by controlling the number of noise requests utilised. Hence, the customer or noise obfuscation could reduce the volume of the noise data injected into real ones in noise obfuscation 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 'high' privacy-leakage-tolerance required service providers, less (or ever no) noise data needs to be utilised by noise obfuscation; and for 'low' privacy-leakage-tolerance service providers, more noise data can be utilised by noise obfuscation.

But existing noise generation strategies do not consider this so far. They focus on the worst case scenario to design noise generation processes and protect customers' privacy on their best. Hence, 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 larger cost on noise data in cloud environments.

To address this, considering existing noise obfuscations [8, 9], a noise data set is a key issue to generate noise data and connect a customer-defined privacy-leakage-tolerance and a specific noise generation process. It includes all possible noise data items which could be utilised in noise obfuscation, based on the customer-set privacy-leakage-tolerance. At server side, 'unethical' service providers could not distinguish which requests are real ones based on this set. Hence, the size of this noise generation set can describe the intensity of noise utilisation to some extents. To make noise obfuscation to be more practical, we will discuss the privacy-leakage-tolerance to manage noise generation processes. It means cost-saving as customers' wishes. This is the main contribution in this paper.

Generally speaking, we propose a 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 real privacy leakage risk in terms of cloud customers. It is the basis to improve the efficiency of privacy protection on noise obfuscation in the pay-as-you-go cloud environments. Then, the creation of noise generation set can be presented based on the privacy-leakage-tolerance and the real privacy leakage risk. We present a privacy-leakage-tolerance based noise generation set creation model to describe these creation processes. Besides, the set can manage existing noise generation strategies and express the advantage of this enhancing strategy based on executions of noise generation strategies. Finally, we present our novel noise enhancing strategy for privacy protection in cloud computing to improve the efficiency of privacy protection based on noise obfuscation.

Let us take a weather forecast service as 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 concern in this paper, if the customer has a high privacy-leakage-tolerance about the service provider, the customer may only need a small request set, like one set with two options: 'Sydney' and 'Melbourne', to conceal 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 unnecessary cost based on the privacy-leakage-tolerance is the main motivation of this paper.

The remainder of the paper is organised as follows. In Section 2, we overview the related work. In Section 3, we present our novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) for privacy protection in cloud computing. In Section 4, we perform a simulation to demonstrate that our novel *PTNES* can improve the efficiency of privacy protection on noise obfuscation significantly. Finally in Section 5, we conclude our contributions and point out future work.

II. RELATED WORK

In this section, we overview some privacy protection approaches: such as Privacy-Preserving Data Mining (*PPDM*), Privacy-Preserving Data Publish (*PPDP*), Privacy Information

Retrieval (*PIR*), anonymous network and noise obfuscation. Besides, some existing trust or privacy risk evaluation in cloud can support our solution.

Many researchers are starting to produce and/or have produced remarkable research on privacy protection related to cloud environments. Yan *et al.* [10] use hierarchical identity-based cryptography to realise mutual authentication in inter-clouds. A privacy preserving access control mechanism with authentication [11] has been designed to protect data in open public clouds. These papers express that there are many privacy protection situations in cloud computing that should be considered and protected by many specific privacy protection strategies. In the rest of this section, we overview some typical and widely used approaches.

PPDM reveals a view of privacy leakage in the minutiae [12]. To protect privacy, a randomisation operator can be utilised to protect output privacy in stream mining [13]. Similarly, *PPDP* has a widely utilised field in service web by considering privacy-preserving data publishing on set-valued data [14].

Different from *PPDM* and *PPDP*, *PIR* utilises another approach to protect privacy [15], which mainly prevents database operators from knowing customers' sensitive records. Based on information theory, e-commerce has started to consider practical *PIR* to enhance business processes [16]. Besides, the work on differential privacy [17] is a promising approach to protect customer privacy by pursuing anonymity.

Proxies and anonymity networks to protect customer privacy have been widely discussed. The major goal is to keep anonymity or 'invisibility' in a complex or 'dangerous' network condition. Onion routing [18, 19] provides a solution to make it difficult for attackers to trace the customer via network traffic analysis. In social networks [20] and encrypted communication [21], anonymous network can protect privacy by identity anonymity.

As analysed in Section 1, 'unethical' service providers may exist in cloud environments. They could record customers' service requests and collectively deduce customers' private information. Therefore, customers' privacy needs to be protected without involving service providers. This is the basic scenario we addressed in this paper. *PPDM* is not an ideal choice to address the scenario because it is out of customers' control. *PIR* and *PPDP* are mainly working at server side, hence have the similar problem. Anonymity or proxy networks need service provider's cooperation to enable such access.

Noise obfuscation is another widely adopted method by covering the characters of information and protecting private information. Ardagna *et al.* [22] focus on the location privacy protection in a mobile environment, and present a solution based on different obfuscation operators. Ye *et al.* [7] describe noise injection in searching privacy protection by formulating noise injection problem as a mutual information minimisation problem. And the further discussion about a common model has been presented in terms of obfuscation-based private web search [23]. Zhang *et al.* [8] present a historical probability based noise generation strategy to improve the efficiency of noise obfuscation and obtain a promising cost-saving in cloud environments. In some uncertain situations, forecasting service processes to guide noise generation are necessary to ensure the effectiveness of noise obfuscation on privacy protection [9]. But just like introduced before, existing noise obfuscations do not consider the customer-defined privacy-leakage-tolerance to manage noise generation processes, and it is a weakness in terms of the efficiency of privacy protection on noise obfuscation. This is what we plan to address in this paper.

About trust or privacy risk evaluation in cloud, Neisse *et al.* [24] investigate trust in cloud environments and promote data security. Besides, interoperability in cloud could be enhanced by trust based on heterogeneous domains and trust recommendation [25]. Accountability [26] is also an important aspect considered by trust and privacy risk in cloud. In brief, trust or privacy risk evaluation in cloud can make cloud services and customers perform better in these opaque environments on different aspects. In this paper, they are valuable references for the privacy leakage risk evaluation.

III. NOVEL PRIVACY-LEAKAGE-TOLERANCE BASED NOISE ENHANCING STRATEGY

As introduced before, we can investigate the noise generation set as a common important issue in existing noise generation strategies, and design the creation process of the set by pursuing a lower cost on noise service requests. So, on the basis of this noise generation set, different kinds of noise generation strategies could be utilised to generate noise service requests and protect customers' privacy more efficiently. That is the background of our novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) for privacy protection in cloud computing.

In this section, firstly, we plan to introduce the Privacy-leakage-Tolerance based Noise Injection Model (*PTNIM*) to support *PTNES*. Then, the creation of the noise generation set is introduced, and the Privacy-leakage-Tolerance based Noise generation set Creation Model (*PTNCM*) can summarise it. Lastly, to enhance other existing noise generation strategies, our novel *PTNES* is presented.

A. PTNIM: Privacy-leakage-Tolerance based Noise Injection Model

In this part, based on Section 1, we introduce the Privacy-leakage-Tolerance based Noise Injection Model (*PTNIM*) to support our novel *PTNES*. As introduced before, our *PTNES* is a kind of enhancing strategy which builds on the basis of other noise generation strategies. Besides, the noise injection model is necessary to execute these noise generation strategies. From [7-9, 27], different noise injection models are built for different noise generation strategies, respectively. So, in this paper, we present *PTNIM* in Figure 1 based on the former work.

Denotations in Figure 1are 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. So, 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.

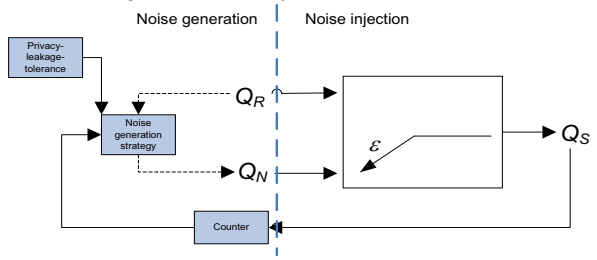


Figure 1 Privacy-leakage-tolerance based noise injection model

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 injection processes. As a novel part, 'privacy-leakage-tolerance' guides 'noise generation strategy' by the creation of noise generation set, and we would detail these processes in the following sections as the main contribution in this paper. In processes of noise injection, noise injection intensity ε is an important parameter from 'noise generation strategy'.

With *PTNIM*, we can present the privacy-leakage-tolerance based noise generation set creation model (*PTNCM*) to support *PTNES*.

B. PTNCM: Privacy-leakage-Tolerance based Noise Generation Set Creation Model

In this part, we present the key part of our *PTNES*—Privacy-leakage-Tolerance based Noise generation set Creation Model (*PTNCM*). In this model, the noise generation set can be created by the privacy-leakage-tolerance. As discussed before, this set can be utilised by noise generation strategies to generate noise requests to conceal real requests. This model can be presented in two aspects. Firstly, the privacy leakage risk evaluation under noise obfuscation can be analysed. Then, the Privacy-leakage-Tolerance based Noise generation set Creation Algorithm (*PTNCA*) can be proposed to describe the creation procedure based on the previous part.

1) Privacy Leakage Risk Evaluation

In this subsection, we investigate the evaluation problem of privacy leakage risk under noise obfuscation. As introduced before, results of the evaluation can guide the noise generation set by creating the noise generation set.

Under existing noise obfuscation work [7-9], to evaluate the privacy leakage risk, we discuss the original set of all service requests, including real ones and noise ones:

$$Q = \{q_1, q_2, ..., q_i, ..., q_n\} \tag{1}$$

Based on this set, we can present a map $f : q_i \to d_i$ to express data items in service requests. In this paper, these data items are potential private information for cloud customers, just like location information in the previous motivating example. In this paper, this map is a bijective map. Because we control data item sets in this noise enhancing strategy rather than service request sets, other complex maps would not influence the creation process of noise set, and we can extend the work in this paper to these conditions without major modifications. Hence, we can get:

$$\forall i \in [1,n], d_i = f(q_i) \tag{2}$$

And the initial noise generation set is:

$$D = \{d_1, d_2, ..., d_i, ..., d_n\} \tag{3}$$

As introduced before, the goal of this paper is to remove some 'useless' noise data for noise obfuscation. Hence, the final noise generation set is a part of the initial noise generation set:

$$D_N = \{d_1, d_2, ..., d_i, ..., d_m\} \tag{4}$$

where $n \geq m \geq 0$.

In this final noise generation set, there maybe one or more items as real private data that should be protected as customer privacy. In other words, these data items should be concealed by other noise data items in the noise generation set with similar occurrence probabilities. Hence, we have:

$$m = x + y \tag{5}$$

The number of these private data items is x, and the number of other noise data items is y. So, we can evaluate the privacy leakage risk under this noise obfuscation with x and y. The set of these private data items is D_x, and the set of other noise data items is D_y. So, we have the join of these two sets:

$$D_N = D_x \vee D_y \tag{6}$$

In D_N, 'unethical' service providers could find out the real private data items with a probability, and we can set this probability as the real privacy leakage risk. In this probability, 'unethical' service providers cannot get extra information from other sources to improve the probability to guess the real private ones. So, we can list all possible cases of real private data items' leakage and combine them to obtain the Real Privacy Leakage Risk under noise obfuscation (PLR_R) which is equation (7). And we use C_m^n to denote the number of n-combinations in a set with m distinct items.

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

In equation (7), 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 equation (7) to evaluate the real privacy leakage risk.

2) PTNCA: Privacy-Leakage-Tolerance based Noise Generation Set Creation Algorithm

From equation (7), we get the real privacy leakage risk or the probability of privacy leakage. So, in this part, we introduce the creation process for the noise generation set by the privacy-leakage-tolerance.

As introduced before, for cloud customers, the requirement for privacy leakage risk is necessary to guide service processes by cloud customers. And this requirement expresses a probability of privacy leakage which is tolerant for them. This is the Customer-defined Privacy-Leakage-Tolerance: PLT_C. And in this paper, this tolerance is in the range of [0, 1]. Besides, for cloud customers, the Real Privacy Leakage Risk (PLR_R) must be equal to or lower than this tolerance:

$$PLT_C \geq PLR_R \tag{8}$$

From equations (7) and (8), we can get:

$$\sum_{i=1}^{x}(\frac{i}{x} * \frac{1}{C_x^i * C_{x+y}^i}) \leq PLT_C \tag{9}$$

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

$$plr(x,y) > plr(x,y+1) \tag{10}$$

Hence, equation (7) is monotonically decreasing with y increasing. And to satisfy the privacy requirement introduced before, we can obtain y:

$$y \geq plr^{-1}(PLT_C) \tag{11}$$

In equation (11), we use the inverse function to get y. According to equation (7), this inverse function is monotonic too. So, we can use stepwise refinement to implement it. Besides, x is fixed under a privacy protection condition, and we omit it in this formula. So, the minimum of y is:

$$y_{\min} = plr^{-1}(PLT_C) \tag{12}$$

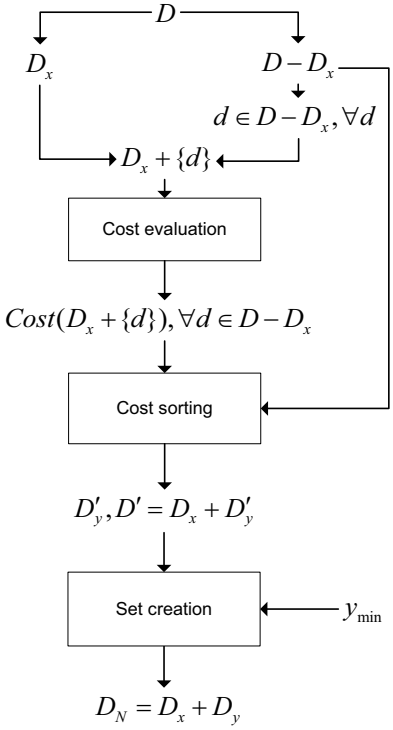
Based on y_{min}, we consider how to obtain D_N.

As discussed before, D_N could decide the intensity of noise obfuscation, and the cost of privacy protection on noise obfuscation can be managed by D_N, too. To get a low cost on noise obfuscation with the same intensity of noise obfuscation, 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 cost of noise data on this data item $Cost(Strategy, d_i)$ should be decided by specific noise generation strategies. The total noise obfuscation cost for customers is:

$$Cost(Strategy, D_N) = \sum_{d_i \in D_N} Cost(Strategy, d_i) \tag{13}$$

With fixed $y:=y_{min}$, to get the lower $Cost(Strategy, D_N)$, we investigate the initial noise generation set: $D = \{d_1, d_2, ..., d_i, ..., d_n\}$, and its sorted set $D'=D_x+D'_y$. Based on equation (6), D'_y is from D-D_x with sorting by cost evaluation. In other words, it is the sorted version of D_y. So, D_y can be created with the lowest y_{min} data item(s) in D'_y. That is *PTNCA* described in Algorithm 1.



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

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

'Set creation' create 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 generation set $D_N = D_x \vee D_y$. So, a noise generation request set can be mapped from D_N:

$$Q_N = f^{-1}(D_N) \quad (14)$$

Briefly, based on the privacy-leakage-tolerance and *PTNCA*, *PTNCM* can be built to support our novel *PTNES* for privacy protection in cloud computing by managing D_N and Q_N.

C. PTNES: Privacy-leakage-Tolerance based Noise Enhancing Strategy

Based on the previous parts, we now present the major contribution of this paper—Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) for privacy protection in cloud computing.

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 generation set is x
Output: the queue of final service requests is Q_S

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

Step 2: Evaluate the privacy leakage risk
Compute the privacy leakage risk on the basis of the sizes of real private data items and other noise data items x and y by equation (7):

$$PLR_R = plr(x, y) = \sum_{i=1}^{x} \left(\frac{i}{x} \times \frac{1}{C_x^i \times C_{x+y}^i} \right)$$

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

$$D_N = PTNCA(Strategy, D, y_{\min});$$

Step 4: Execute one noise generation strategy based on the noise generation set
For the 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 2 *PTNES*: Privacy-leakage-Tolerance based Noise Enhancing Strategy

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

In this algorithm, Step 1 is the beginning step to collect all request queues and sets as past data to support the subsequent steps. Then, in Step 2, we get the real privacy leakage risk to generate the noise generation set D_N. In Step 3, we obtain the key issue of the strategy: 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 by a specific noise generation strategy under *PTNES* enhancing. After this, Steps 1, 2, 3 and 4 would be executed again as a run-time privacy protection mechanism until the whole noise obfuscation process terminates.

In general, the key part of *PTNES* is to build and update noise generation set D_N. We present *PTNCM* and *PTNCA* to summarise the major part of this. With D_N, noise obfuscation considers the customer-defined privacy-leakage-tolerance in noise obfuscation processes by this noise enhancing strategy, just like introduced in Section 1. In the next section, we will illustrate that *PTNES* can improve the efficiency of noise obfuscation on privacy protection with similar effectiveness by simulation.

IV. EVALUATION

In this section, we evaluate *PTNES* by simulation, in terms of effectiveness and efficiency of privacy protection. Compared to existing noise obfuscation work, *PTNES* is a kind of 'enhancing' strategy, different from other specific noise generation strategies. In other words, it executes other typical noise generation strategies by creating the noise generation set. So, in the simulation process, the evaluation of this strategy focuses on its impact on other typical noise generation strategies.

A. Simulation Background and Environment

SwinCloud is a cloud computing simulation environment [28]. It is built on the computing facilities at Swinburne University of Technology. The functions of VMWare can offer unified computing and storage resources. We built several computing nodes to simulate these members-cloud customers and cloud service providers in cloud, and exchanged data as requests with each other. Each noise obfuscation process builds between these nodes and protects customers' privacy.

B. Simulation Process

In this part, *PTNES* can be evaluated step by step. In the SwinCloud environment, we use some nodes as customers to send service requests with specific noise obfuscations. It also evaluates the cost of noise data to get the efficiency of privacy protection on noise obfuscation. Other nodes are utilised as service providers to receive service requests and evaluate the effectiveness of noise obfuscation on privacy protection.

And existing typical noise generation strategies: the random strategy (*RNGS*) [7], the historical probability based strategy (*HPNGS*) [8] and the time-series pattern based strategy (*TPNGS*) [9] can be enhanced by *PTNES*. We will describe these positive improvements on these strategies by *PTNES*.

In this paper, we use *Noise Cost* to denote the cost of noise service requests and express the efficiency of privacy protection on noise obfuscation in this paper. 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 Section 3, the setting of PLT_C as *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 which 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.

C. Simulation Results and Analysis

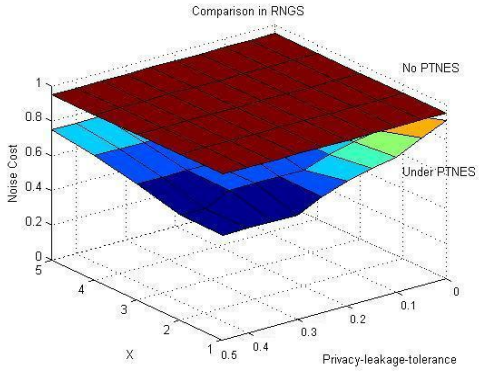


Figure 2 Comparison in *RNGS*

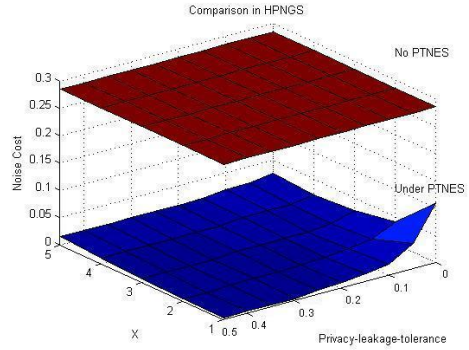


Figure 3 Comparison in *HPNGS*

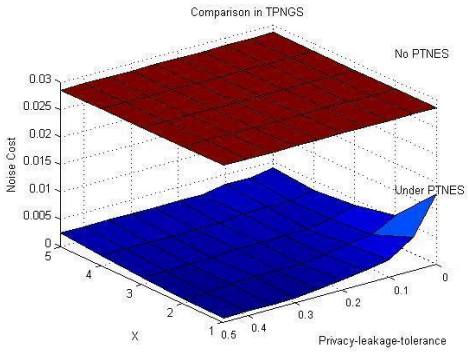


Figure 4 Comparison in *TPNGS*

As introduced before, for *RNGS*, *HPNGS* and *TPNGS*, the cost of noise generation can be compared in two situations: with and without *PTNES* enhancing.

In Figure 2, there are three coordinates: *Noise cost*, *x* and *privacy-leakage-tolerance*. When *RNGS* operates, our *PTNES* can reduce *Noise Cost* from about 0.95 to 0.6, compared to without *PTNES*. In Figure 3, when *HPNGS* operates, our *PTNES* can reduce *Noise Cost* from about 0.28 to 0.02, compared to without *PTNES*. In Figure 4, when *TPNGS* operates, our *PTNES* can reduce *Noise Cost* from about 0.029 to 0.002, compared to without *PTNES*. Hence, in typical noise generation strategies, *PTNES* can improve the efficiency of privacy protection on noise obfuscation by significantly reducing *Noise Cost*.

Besides, in these figures, we can find out that when one noise generation strategy operates without *PTNGS*, *Noise Cost* keeps in a high level with the increasing of *x* and *privacy-leakage-tolerance*. But with *PTNES* enhancing in Figure 2, Figure 3 and Figure 4, when *privacy-leakage-tolerance* increases, *Noise Cost* decreases. It is obvious that if customers have a low privacy-leakage-tolerance setting for a cloud service, a high cost on noise obfuscation has to be paid by customers. And, about another axis *x* in our figures: low *x* means low *Noise cost*. In other words, if customers plan to protect more data items as privacy, the cost should be more.

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

V. CONCLUSIONS AND FUTURE WORK

In open cloud environments, customers' privacy protection is a challenge as malicious service providers may record customer service data and then collectively deduce customers' privacy. Noise obfuscation is an effective approach in this regard. For example, it generates and injects noise service requests into real customer service requests to make sure that their occurrence probabilities are about the same. So service providers cannot distinguish which ones are real. However, existing typical noise generation strategies did not consider customers' specific privacy requirements on noise generation processes. A higher privacy-leakage-tolerance would require a lower cost on noise obfuscation processes in the pay-as-you-go style for cloud computing. Hence, to deal with this privacy concern, we have developed a novel Privacy-leakage-Tolerance based Noise Enhancing Strategy (*PTNES*) for privacy protection in cloud computing. In this strategy, we investigate the evaluation on real privacy leakage risk under existing noise obfuscations, and present the creation of noise generation set to manage noise generation processes. Based on this set's efficient creation, our strategy can guide and manage noise generation processes to decrease the cost of noise with a reasonable effectiveness of privacy protection. In a word, our novel strategy can 'enhance' noise obfuscation by considering this customer-defined boundary. The simulation evaluation has demonstrated that our strategy can decrease noise cost under other noise generation strategies significantly for improving the efficiency of privacy protection on noise obfuscation.

Based on *PTNES*, we plan to further investigate how to protect customer privacy in the scenario where these unethical or malicious service providers may collaborate together to deduce customers' privacy.

ACKNOWLEDGEMENT

The research work reported in this paper is partly supported by Australian Research Council under LP110100228.

REFERENCES

[1] 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,* 25(6): 599-616, 2009.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H.Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Z. Matei, "Above the Clouds: A Berkeley View of Cloud Computing," *Communications of the ACM,* 53(6): 50-58, 2010.

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

[4] Mark D. Ryan, "Cloud Computing Privacy Concerns on Our Doorstep," *Communications of the ACM* 54(1): 36-38, 2011.

[5] Wayne Jansen and Grance Timothy, "Guidelines on Security and Privacy in Public Cloud Computing," National Institute of Standard and Technology , Special Publication 800-144, http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf , Accessed in November 10th, 2012.December 2011.

[6] 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,* 2(6): 34-41, 2004.

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

[8] 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,* 78(5): 1374-1381, 2012.

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

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

[11] Sushmita Ruj, Milos Stojmenovic, and Amiya Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2012), pp. 556-563, Ottawa, Canada, May13-16, 2012.

[12] Rakesh Agrawal and Ramakrishnan Srikant, "Privacy-Preserving Data Mining," *ACM SIGMOD Record,* 29(2): 439-450, 2000.

[13] Ting Wang and Ling Liu, "Output Privacy in Data Mining," *ACM Transactions on Database Systems,* 36(1): 1-34, 2011.

[14] Mingqiang Xue, Panagiotis Karras, Chedy Raïssi, Jaideep Vaidya, and Kian-Lee Tan, "Anonymizing Set-Valued Data by Nonreciprocal Recoding," 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12), pp. 1050-1058, Beijing, China, August 12-16, 2012.

[15] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan, "Private Information Retrieval," *Journal of ACM,* 45(6): 965-981, 1998.

[16] Ryan Henry, Femi Olumofin, and Ian Goldberg, "Practical PIR for Electronic Commerce," 18th ACM Conference on Computer and Communications Security (CCS'11), pp. 677-690, Chicago, Illinois, USA, October 17-21, 2011.

[17] Graham Cormode, "Personal Privacy vs Population Privacy: Learning to Attack Anonymization," 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11), pp. 1253-1261, San Diego, California, USA, 2011.

[18] David Goldschlag, Michael Reed, and Paul Syverson, "Onion Routing," *Communications of ACM,* 42(2): 39-41, 1999.

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

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

[21] 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,* 7(4): 1381-1393, 2012.

[22] 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,* 8(1): 13-27, 2011.

[23] Ero Balsa, Carmela Troncoso, and Claudia Diaz, 2012, "OB-PWS: Obfuscation-Based Private Web Search," in *2012 IEEE Symposium on Security and Privacy (SP)*, San Francisco, USA, May 20-232012, pp. 491-505,

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

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

[26] 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," 2011 IEEE World Congress on Services (SERVICES), pp. 584-588, Washington, DC, USA, July 4-9 2011.

[27] 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,* 42(4): 431-445, 2012.

[28] 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, 2011