Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 384967, 12 pages
http://dx.doi.org/10.1155/2014/384967

Research Article

Trusted Measurement Model Based on Multitenant Behaviors

Zhen-Hu Ning, Chang-Xiang Shen, Yong Zhao, and Peng Liang

College of Computer Science, Beijing University of Technology, Beijing 100124, China

Correspondence should be addressed to Zhen-Hu Ning; ning_zhenhu@163.com

Received 13 January 2014; Accepted 16 February 2014; Published 30 March 2014

Academic Editors: Y. Mao, X. Meng, J. Zhou, and Z. Zhou

With a fast growing pervasive computing, especially cloud computing, the behaviour measurement is at the core and plays a vital role. A new behaviour measurement tailored for Multitenants in cloud computing is needed urgently to fundamentally establish trust relationship. Based on our previous research, we propose an improved trust relationship scheme which captures the world of cloud computing where multitenants share the same physical computing platform. Here, we first present the related work on multitenant behaviour; secondly, we give the scheme of behaviour measurement where decoupling of multitenants is taken into account; thirdly, we explicitly explain our decoupling algorithm for multitenants; fourthly, we introduce a new way of similarity calculation for deviation control, which fits the coupled multitenants under study well; lastly, we design the experiments to test our scheme.

1. Introduction

Cloud computing has recently attracted an important attention and dubbed as the "next best thing" in information and communication technologies (ICT) [1]. As the intrinsic feature of cloud computing, multitenancy brings sharing concept to almost all information technologies such as sharing computing resources, sharing storage resources, and sharing network. Coresident clients might have no preestablished trust relationship and might have no knowledge of the existence or identities of other clients. In such a setting, if one of the coresidents maybe attacks the other coresidents it will be much easier to succeed and be difficult to detect. Therefore, this risk incurred by trusted measurement of multitenant is a barrier to acceptance of cloud computing. Actually cloud computing system, such as Amazon's Elastic Compute Cloud (EC2), Microsoft's Azure, and Rackspace's Mosso, is a large scale system which is studied in cybernetics long before. Here we leverage the generalized predictive control affiliated to cybernetics to solve the problem of behavior measurement of multitenants on the same physical server brought by the new paradigm of cloud computing.

2. Background

This section consists of two parts: one is multitenancy threat; the other is the brief introduction of generalized predictive control.

2.1. Multitenancy Threat. It is important to consider the unique security risks introduced by multitenancy as intrinsic of the new paradigm of cloud computing in order to be able to derive adequate security solutions. As more and more applications become exported to third-party compute clouds, it becomes increasingly important to quantify any threats to confidentiality that exist in this setting [2, 3]. An obvious threat to these consumers of cloud computing is malicious behavior by the cloud provider, who is certainly in a position to violate customer confidentiality or integrity. However, this is a known risk with obvious analogs in virtually any industry practicing outsourcing. In this work, we consider the provider and its infrastructure to be trusted. This also means we do not consider attacks that rely upon subverting a cloud's administrative functions, via insider abuse or vulnerabilities in the cloud management systems (e.g., virtual machine monitors).

In our threat model, adversaries are non-provider-affiliated malicious parties. Victims are multitenants running confidentiality-requiring services in the cloud. A traditional threat in such a setting is direct compromise, where an attacker attempts remote exploitation of vulnerabilities in the software running on the system. Of course, this threat exists for cloud applications as well. These kinds of attacks (while important) are a known threat and the risks they present are understood.

We instead focus on where third-party cloud computing gives attackers novel abilities, implicitly expanding the attack

surface of the victim. We assume that, like any customer, a malicious party can run and control many instances in the cloud, simply by contracting for them. Further, Based on the fact the economies offered by third-party compute clouds derive from multiplexing physical infrastructure, we assume (and later validate) that attacker's instances might even run on the same physical hardware as potential victims. From this vantage, an attacker might manipulate shared physical resources (e.g., CPU caches, branch target buffers, network queues, etc.) to learn otherwise confidential information.

2.2. Generalized Predictive Control. In general sense, predictive control, regardless of various algorithms, is based on the following three basic principles [4].

(1) Predictive Model. Predictive control is also referred to as model-based control where this model is referred to as predictive model. The predictive model can predict the future output of the object based on historical information and input. And the predictive model does not emphasize its structure but emphasizes the function of the model. Therefore, the traditional model such as equation of state and the transfer function can be used as a predictive model. Similarly, nonparametric model such as step response and impulse response can also be used directly as a predictive model.

(2) Rolling Optimization. Predictive control is an optimal control algorithm, which determines the future action through an optimal performance index. However, the optimization studied in predictive control is different from optimal control in the traditional sense, and the subtle difference is that optimization in the predictive control is a rolling optimization within the limited time. At each sampling instant, the optimization performance indicators relate only to a limited time since the right moment. Until the next sampling instant, this optimization period moves forward. At different instants, the relative forms of optimization performance indicators are the same, but its absolute form, that is, containing time area, is different. Therefore, during predictive control, optimization is not offline conducted only once but repeated online, which is the core of rolling optimization, that is, the fundamental characteristics of optimal control here is different from the traditional ones.

(3) Feedback Correction. Predictive control is a closed-loop control algorithm, where a series of further control actions can be ascertained by optimization. Predictive control does not perform all these actions but perform the present action. So that the deviation from the ideal state can be avoided; this is resulted from either the model mismatch or environmental interference. Until the next sampling time, the first is to detect the actual output of the object; the second is to take advantage of this real-time information to correct the prediction based on the model; and the final is to conduct the new optimization. Therefore, the optimization of the predictive control is not only based on the model, but also the feedback information, which constitutes a closed-loop optimization.

3. Related Work

There exist several measurement models such as Tripwire [5], AEGIS [6], and trusted box [7], the trust chain model proposed by the TCG (trusted computing group). These models focus on different measurement aspects of the system or file program, but these approaches belong to static integrity measurement of the resource. As a result, they cannot consider the dynamic trustworthiness in the system.

Further the researchers put forward the following schemes to realize dynamic measurement. In [8], there is a coprocessor-based kernel integrity monitor. The monitor periodically checks system memory and detects whether malicious programs change the host system kernel. Binding instructions and data (BIND) binds with the data and the corresponding block of process in order to provide a basis for the verification side to trace data processing. However, it cannot cope with many attacks when the system is running [9]. Policy reduced integrity measurement architecture (PRIMA) focuses on the flow of information when the system is running [10], but the model trusts flow of information which comes from the trusted subjects in mandatory access control (MAC). However, it is still a role-based privilege. The measure mode is too simple and does not conform to the definition of definition of trust. Behavior based trustworthiness attestation mode (BTAM) is trusted proof model based on the behavior of the system [11]. This model firstly determines whether the system behavior is related to trustworthiness of platform state. For a large number of behaviors that cannot be determined, this model has not yet given the solution. Therefore, the dynamic trusted measure theory and technology is an urgent need for the development of cloud computing [12].

Gong [13] firstly introduces generalized prediction control theory to analyze and measure the tenants' behaviour in the information system. The novel scheme greatly increases the trustworthiness and security of information system and opens a new direction towards behaviour measurement [13]. However, the new features mentioned above brought by the cloud computing were not considered and studied. This paper is to improve that model and to adapt the new feature of multitenancy brought by cloud computing.

4. Model Design

Traditional authorization and authentication are to solve the main problem whether the user's identity is trusted, while they are ineffective to solve whether the user's behavior is trusted. The original drive to promote the change of system status is the behavior [14]. Therefore, the trusted measurement of the behavior is more precise than the trusted measurement of the identity when it comes to reflect the trustworthiness of the system. The design of our model is consistent with the trustworthiness defined by Trusted Computing Group (TCG); that is, it is defined as trusted if the behavior can be expected [13]. According to this definition, we propose a measurement model for virtual machine behavior shown in Figure 1.

The first step: the characteristics of the shared resources in cloud computing brings the advantages while leading to

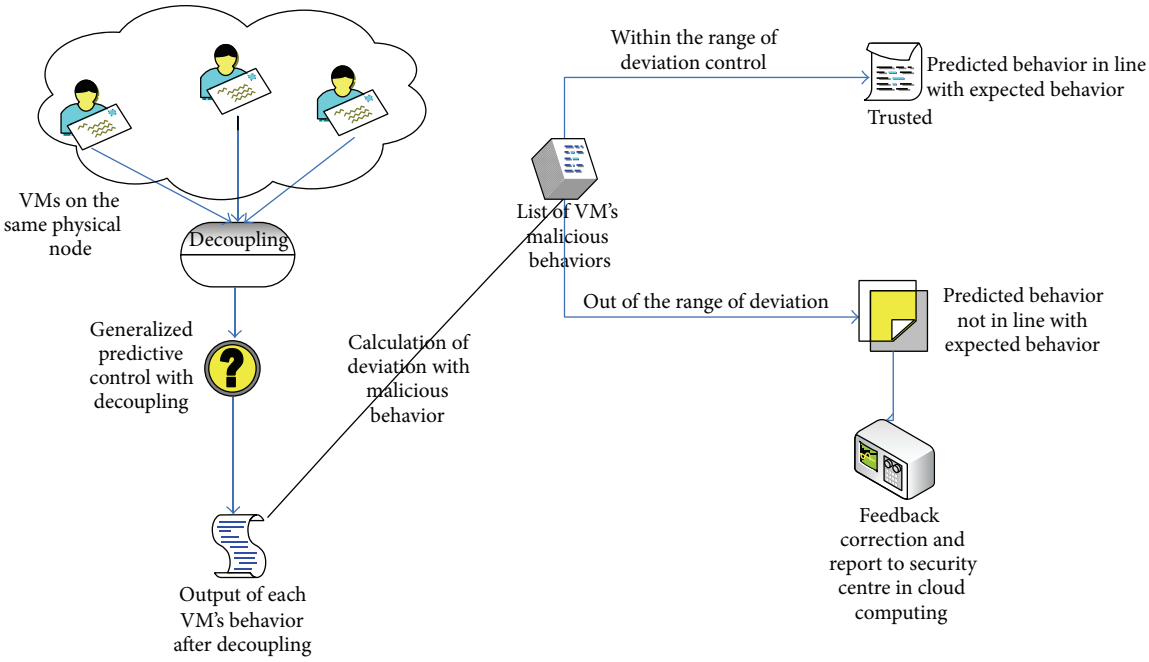

Figure 1: Behavior measurement model of virtual machine.

security problems. So it is necessary to conduct decoupling control over the behavior of the virtual machines on the same physical platform. Illustratively, the decoupling control aims to simplify the control over many virtual machines sharing resources of the same physical computing node into a lot of individual control loops for each virtual machine corresponding to individual customers.

The second step: according to the decoupling control algorithm, the inputs and outputs of several virtual machines in the same physical computer can be decoupled. The decoupled inputs and outputs of appropriate virtual machine can be controlled by the generalized predictive control algorithm here. Specifically, through the past and present behavior of the virtual machine, the further behavior can be predicted.

The third step: to match predicted behavior with characteristics list of malicious behaviors so as to obtain the similarity value/deviation value. If the deviation value is less than the threshold value predetermined by the system, then the behavior is trusted, otherwise it is an untrusted behavior.

5. Model Implementation

The multiple tenants studied here refer to the ones who share the same physical resource such as network card and bandwidth. Due to the multitenancy sharing, the cloud computing becomes much more complicated. In order to better predict the tenant's complicated behaviors, we utilize the multiple variable generalized predictive control to capture those behaviors. In this section, firstly we depict the cloud computing system in the view of generalized predictive control; secondly, we present the description of behaviors in cloud computing; thirdly, we introduce the establishment of list of malicious tenants' behaviors; fourthly, we give decoupling algorithm for multitenant behaviors both in private and public clouds using generalized predictive control without coupling; fifthly, we give the similarity calculation used in our scheme for deviation control to confirm whether the suspected behavior is trusted or not finally.

5.1. Description of Controlled Object. From the view of control theory, the physical computing nodes where several virtual machines colocate can be taken as a multi-input, multioutput information flow control system. Figure 2 shows a physical computing node colocated by four virtual machines from the perspective of the generalized predictive control theory. Eight behavioral measurement points are as input of the information system; the outputs are four virtual machines captured by eight behavioral measurement points, which are in line with the appropriate expectation, respectively. Each virtual machine is one of outputs of the entire physical computing node, while all four virtual machines are equal to total inputs of the entire physical compute node, such that the total traffic of all four virtual machines should be equal to the traffic of physical computing nodes.

5.2. Description of Tenant's Behavior. There exists monitoring components in virtualized trusted computing platform based on dual-system architecture proposed by our research team. These monitoring components can identify measurement indicators of the behavior performance of virtual machine. There are several commonly used monitoring components as follows: (1) memory and CPU monitor: to monitor memory usage and CPU call rate and report monitoring results to the behavioral data collector; (2) port monitor and message analyzer: responsible for monitoring all open TCP or UDP ports of compute nodes and capturing and

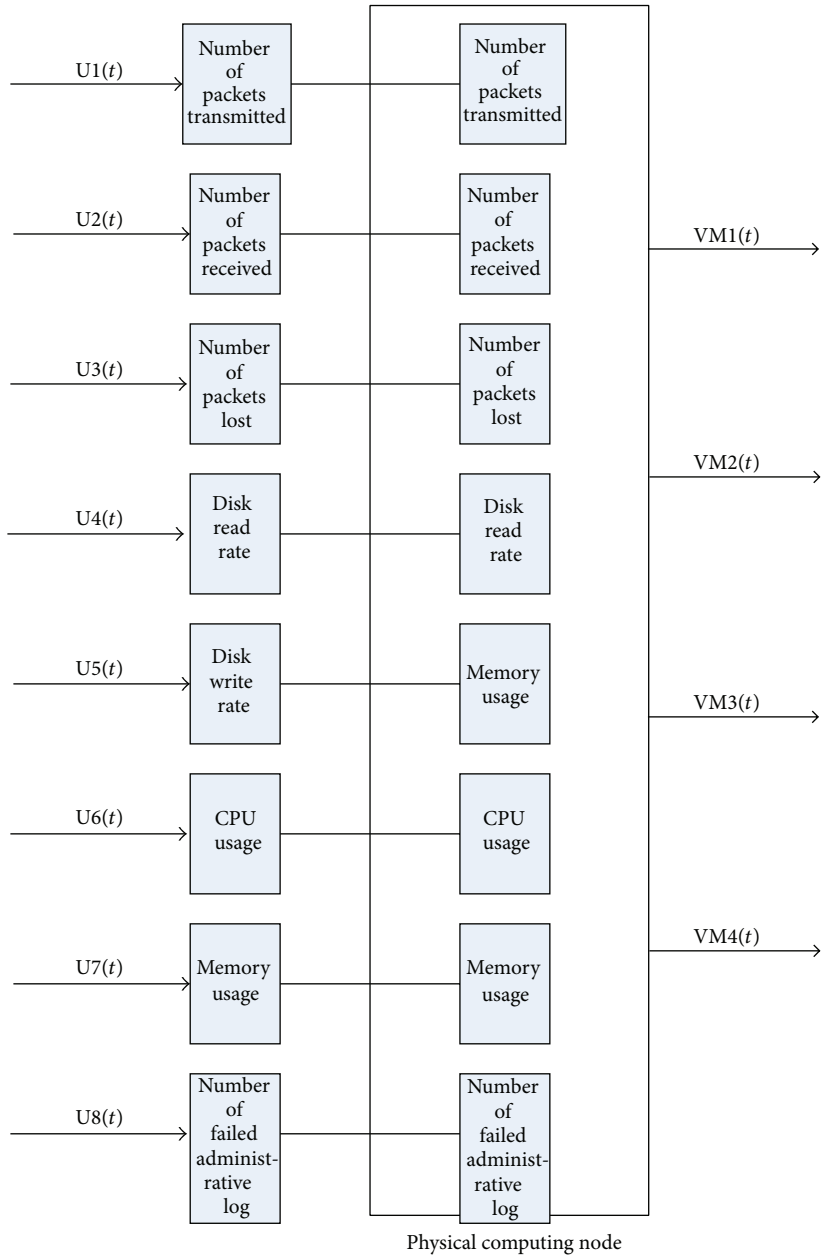

Figure 2: Description of physical computing nodes in the view of generalized predictive control.

analyzing communication message packet of the suspicious port. So that we can determine the role of the suspicious port and the corresponding process behavior of this port. If a suspicious user process is found to monitor a suspicious port and to communicate the message frequently, it is necessary to temporarily suspend the implementation of the process and to report to the Cloud Security Management Center; (3) network traffic detector: its role is to monitor the flow of network communication, in particular, the network traffic coming out of a virtual machine. Each virtual node has been deployed the monitor, so that both the denial of service attacks and the worm can be monitored and found. As a matter of fact, DoS and worm attacks will lead to a sharp rise in network traffic. If it is found that a virtual machine computing task unconventionally and frequently sends out a lot of the packages with the same content, this task needs to be suspended, that is, to prevent the execution of the virtual machine user tasks, and then to be reported to the Cloud Security Management Center.

In the cloud computing model, we studied the related results conducted by both foreign researchers such as Khorshed et al. [15] and local researchers such as Li et al. [16]; we choose the following to depict the virtual machine behavior, which is the number of transmitted packets, the number of received packets, the number of lost packets, disk read speed, disk write speed, memory usage, CPU usage, and the number

Table 1: Virtual machine behavior metric vector.

Measurement point	Measured object
MP_1	Number of packets transmitted
MP_2	Number of packets received
MP_3	Number of packets lost
MP_4	Disk read rate
MP_5	Disk write rate
MP_6	Memory usage
MP_7	CPU usage
MP_8	Number of failed administrative log on attempt

of failed login attempts. Here, these eight performance indicators are named as measurement point, abbreviated as MP. In this paper, the behavior measurement vector of running virtual machine consists of the aforementioned 8 measurement points, see Table 1.

5.3. List of Tenant's Malicious Behaviors. The researchers from University of California, San Diego, and the Massachusetts Institute of Technology, Cambridge University [17] conducted a thorough experimental study on Amazon's Elastic Compute Cloud [18]. The results show that the cloud infrastructure can be mapped out, and the position of a specific virtual machine can be located. They also point out that the aforementioned information can be exploited to make side channel attacks so as to collect the information of the target virtual machine located on the same physical machine. In a recent study, Rocha and Correia [19] investigated how malicious insiders steal confidential data and demonstrated these attacks using the video and showing insiders can easily obtain passwords, encryption keys, and documents. Chonka et al. [20] reproduced the scenario of some recent attacks happening in the cloud computing and demonstrated how the HTTP-DOS and XML DoS occur in the cloud computing.

Khorshed et al. found that there exists some common factor behind these attack models [17–19], because all the attackers use a similar attack tools and follow a certain attack process. Khorshed et al. firstly collected relevant attack tools such as Hping, socket programming, httping Unix shell script, and side channel attacks. Next they collected a variety of attack scenarios related to network security by browsing relevant website and blog, such as Danchev [21] and Grossman [22] as well as their research work [23–25], and then generated attack script using the aforementioned documents.

Based on above steps, Khorshed et al. designed the experiment to collect data in the cloud computing environment. The type of data will determine the kind of data collection tools. In the attack scenario, most common data types are as follows 8 performance indicators such as the number of transmitted and received data packets, processing time, the round-trip time, and CPU usage. Khorshed et al. adopted machine learning techniques to classify the attacks related to malicious use of resources in the cloud computing. Through a large number of experiments, they obtained 8 measurement points of behavioral performance such as the number of transmitted packets, the number of received packets, the number of lost packets, disk read speed, disk write speed, memory usage, CPU usage, and the number of failed login attempts. Further, they concluded the behavioral characteristics of the classic attack in conduction of eight measurement points [26].

5.4. Decoupling Algorithm. To maximize efficiency, multiple VMs, one VM corresponding to one tenant, may be simultaneously assigned to be executed on the same physical server, which is supported by virtualization technology. As a result, tenants share the physical resources (e.g., CPU caches, branch target buffers, network queues, etc.) to accomplish their computation tasks. From the angle of generalized predictive control (GPC), cloud computing system under study corresponds to multiple inputs and multiple outputs system in cybernetics which is different from the single input and output system that is studied in [13]. The essential difference is the coupling between tenants on the same physical server, which should be studied thoroughly. In this section, first we use GPC theory to capture the multitenant behavior and then to derive the decoupling algorithms that is shown at the end of this part.

The multitenant's behavior in cloud computing can be described by

$$A\left(z^{-1}\right) y\left(t\right) = D\left(z^{-1}\right) B\left(z^{-1}\right) u\left(t-1\right) + \frac{C\left(z^{-1}\right)\xi\left(t\right)}{\Delta}, \tag{1}$$

where $A(z^{-1}) = I + A_1 z^{-1} + \cdots + A_{n_A} z^{-n_A}$, $B(z^{-1}) = I + B_1 z^{-1} + \cdots + B_{n_B} z^{-n_B}$, $D(z^{-1}) = \mathrm{diag}(z^{-k_s})$, $\Delta = \mathrm{diag}(1 - z^{-1})$.

$\{u(t)\}$ and $\{y(t)\}$ indicate coresident tenants' inputs and outputs. $\xi(t)$ is m-dimension independent random disturbance vector, and its mean value and variance are zero and σI, respectively. Without loss of generality, suppose $A(z^{-1})$ is diagonal matrix.

$B(z^{-1})$ is divided into two parts, namely,

$$B\left(z^{-1}\right) = \overline{B}\left(z^{-1}\right) + \widetilde{B}\left(z^{-1}\right), \tag{2}$$

where $\overline{B}(z^{-1})$ is diagonal matrix polynomials and $\widetilde{B}(z^{-1})$ is a matrix whose diagonal is zero. Equation (2) indicates that $\overline{B}(z^{-1})$ is the direct relation between tenant's inputs and outputs, and $\widetilde{B}(z^{-1})$ is the mutual coupling part of communication channel.

Using (1) and (2), we have

$$\begin{aligned} A\left(z^{-1}\right)\Delta y\left(t\right) = {} & D\left(z^{-1}\right)\overline{B}\left(z^{-1}\right)\Delta u\left(t-1\right) + D\left(z^{-1}\right) \\ & \times \widetilde{B}\left(z^{-1}\right)\Delta u\left(t-1\right) + C\left(z^{-1}\right)\xi\left(t\right). \end{aligned} \tag{3}$$

Performance index function is as follows:

$$\begin{aligned} J = {} & \xi \cdot \left\{ \sum_{j=1}^{N} \left\| \phi\left(t+j\right) - r_j \omega\left(t+j\right) + \widetilde{S_j}\left(z^{-1}\right)\Delta u\left(t+j-1\right) \right\|_Q^2 \right. \\ & \left. + \sum_{j=1}^{N} \left\| \Delta u\left(t+j-i\right) \right\|_{\lambda_j}^2 \right\}, \end{aligned} \tag{4}$$

where

$$\phi(t+j) = D(z)\,\Delta y(t+j) \tag{5}$$

indicates generalized outputs, $D(z)$ indicates the inverse of $D(z^{-1})$, $\omega(t+j)$ is fixed vector, $||X||_Q^2$ indicates X^TQX, and Q is symmetric positive definite matrix. There is no such $\widetilde{S}_j(z^{-1})u(t+j-1)$, part of (4), in the performance index of common generalized prediction control. $\widetilde{S}_j(z^{-1})$ is a matrix polynomial whose diagonal is zero and $\widetilde{S}_j(z^{-1})$ can be used to eliminate the coupling effect between channels. Similarly, weighted constant matrix λ_i can be divided into two $\overline{\lambda}_j$ and $\widetilde{\lambda}_j$; $\overline{\lambda}_j$ is a diagonal matrix and $\widetilde{\lambda}_j$ is a matrix whose diagonal is zero; the function of $\widetilde{\lambda}_j$ is the same as that of $\widetilde{S}_j(z^{-1})$.

We use the methods in [27] to achieve the decoupling algorithm.

Define Diophantine equation:

$$I = F_j\left(z^{-1}\right)A\left(z^{-1}\right) + z^{-j}D\left(z^{-1}\right)G_j\left(z^{-1}\right), \tag{6}$$

where $F_j(z^{-1}) = I + F_1^j z^{-1} + \cdots + F_{n_D+j-1}^j z^{-n_D-j+1}$, $G_j(z^{-1}) = G_0^j + G_1^j z^{-1} + \cdots + G_{n_A-1}^j z^{-n_A+1}$.

Since $A(z^{-1})$ and $D(z^{-1})$ are diagonal matrix, $F_j(z^{-1})$ and $G_j(z^{-1})$ are diagonal matrix as well. Equation (6) is left multiplied with $D(z^{-1})$:

$$D(z) = D(z)F_j\left(z^{-1}\right)A\left(z^{-1}\right) + z^{-j}G_j\left(z^{-1}\right). \tag{7}$$

$D(z)F_j(z^{-1})$ left multiplies with (3), and using the above formula, we obtain the following:

$$\begin{aligned} &D(z)\,\Delta y(t+j) \\ &\quad = F_j\left(z^{-1}\right)\overline{B}\left(z^{-1}\right)\Delta u(t+j-1) \\ &\qquad + F_j\left(z^{-1}\right)\widetilde{B}\left(z^{-1}\right)\Delta u(t+j-1) + G_j\left(z^{-1}\right)\Delta y(t) \\ &\qquad + D(z)F_j\left(z^{-1}\right)C\left(z^{-1}\right)\xi(t+j). \end{aligned} \tag{8}$$

Since the term $D(z)F_j(z^{-1})C(z^{-1})\xi(t+j)$ is unrelated to other terms, optimal prediction of $\phi(t+j)$ can be represented as follows:

$$\begin{aligned} &\phi^0(t+j) \\ &\quad = F_j\left(z^{-1}\right)\overline{B}\left(z^{-1}\right)\Delta u(t+j-1) \\ &\qquad + F_j\left(z^{-1}\right)\widetilde{B}\left(z^{-1}\right)\Delta u(t+j-1) + G_j\left(z^{-1}\right)\Delta y(t). \end{aligned} \tag{9}$$

Both $F_j(z^{-1})\overline{B}(z^{-1})$ and $F_j(z^{-1})\widetilde{B}(z^{-1})$ can be divided into two parts:

$$\begin{aligned} F_j\left(z^{-1}\right)\overline{B}\left(z^{-1}\right) &= E_j\left(z^{-1}\right) + z^{-j}L_j\left(z^{-1}\right), \\ F_j\left(z^{-1}\right)\widetilde{B}\left(z^{-1}\right) &= \widetilde{E}_j\left(z^{-1}\right) + z^{-j}\widetilde{L}_j\left(z^{-1}\right), \end{aligned} \tag{10}$$

where

$$\begin{aligned} E_j\left(z^{-1}\right) &= \sum_{i=1}^{j} E_i^j z^{-i}, \qquad L_j\left(z^{-1}\right) = \sum_{i=1}^{n_D+n_B-1} L_i^j z^{-i}, \\ \widetilde{E}_j\left(z^{-1}\right) &= \sum_{i=1}^{j} \widetilde{E}_i^j z^{-i}, \qquad \widetilde{L}_j\left(z^{-1}\right) = \sum_{i=1}^{n_D+n_B-1} \widetilde{L}_i^j z^{-i}. \end{aligned} \tag{11}$$

Equation (9) can be represented as

$$\begin{aligned} \phi^0(t+j) = {} & E_j\left(z^{-1}\right)\Delta u(t+j-1) + \widetilde{E}_j\left(z^{-1}\right)\Delta u(t+j-1) \\ & + G_j\left(z^{-1}\right)\Delta y(t) + L_j\left(z^{-1}\right)\Delta u(t+j-1) \\ & + \widetilde{L}_j\left(z^{-1}\right)\Delta u(t+j-1). \end{aligned} \tag{12}$$

Substitute the above formula into (4), and choose $\widetilde{S}_j(z^{-1})$ that satisfies

$$\begin{aligned} &\widetilde{S}_j\left(z^{-1}\right)\Delta u(t+j-1) + \widetilde{E}_j\left(z^{-1}\right)\Delta u(t+j-1) \\ &\quad + \widetilde{L}_j\left(z^{-1}\right)\Delta u(t+j-1) = \widetilde{M}_j\left(z^{-1}\right)\Delta u(t-1), \end{aligned} \tag{13}$$

where $\widetilde{M}_j(z^{-1}) = \widetilde{M}_0^j + \widetilde{M}_1^j z^{-1} + \cdots + \widetilde{M}_{n_M}^j z^{-n_M}$ is a matrix polynomial whose diagonal is zero, so (4) can be represented as

$$\begin{aligned} J = {} & \sum_{j=1}^{N} \Big\| E_j\left(z^{-1}\right)\Delta u(t+j-1) + L_j\left(z^{-1}\right)\Delta u(t+j-1) \\ & + \widetilde{M}_j\left(z^{-1}\right)\Delta u(t-1) + G_j\left(z^{-1}\right)\Delta y(t) - r_j\omega(t+j)\Big\|_Q^2 \\ & + \sum_{j=1}^{N} \left\|\Delta u(t+j-i)\right\|_{\lambda_j}^2 \\ = {} & \|EU + L\Delta u(t-1) + G\Delta y(t) + \widetilde{M}\Delta u(t-1) \\ & - RW\|_I^2 + \|U\|_\lambda^2, \end{aligned} \tag{14}$$

where $R = \mathrm{diag}(r_j)$, $\lambda = \mathrm{diag}(\overline{\lambda}_j) + \mathrm{diag}(\widetilde{\lambda}_j) = \overline{\lambda}_j + \widetilde{\lambda}_j$, $j = 1, \ldots, N$,

$$\begin{aligned} E &= \begin{bmatrix} E_{10} \\ E_{21}E_{20} \\ \vdots \\ E_{NN-1}E_{NN-2}\cdots E_{N0} \end{bmatrix}, \\ U &= [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+N-1)]^T, \\ W &= [w(t), w(t+1), \ldots, w(t+N-1)]^T, \\ G &= \left[G_1\left(z^{-1}\right), G_2\left(z^{-1}\right), \ldots, G_N\left(z^{-1}\right)\right]^T, \\ L &= \left[L_1\left(z^{-1}\right), L_2\left(z^{-1}\right), \ldots, L_N\left(z^{-1}\right)\right]^T, \\ \widetilde{M} &= \left[\widetilde{M}_1\left(z^{-1}\right), \widetilde{M}_2\left(z^{-1}\right), \ldots, \widetilde{M}_N\left(z^{-1}\right)\right]^T. \end{aligned} \tag{15}$$

Calculate the minimum of J so that we obtain

$$\begin{aligned} U = {} & \left(E^T E + \overline{\lambda}\right)^{-1} E^T \\ & \times \left[RW - G\Delta y(t) - L\Delta u(t-1) - \widetilde{M}\Delta u(t-1)\right] \\ & - \left(E^T E + \overline{\lambda}\right)^{-1} \widetilde{\lambda} U, \end{aligned} \tag{16}$$

where the value of matrix $\widetilde{M}$ and $\widetilde{\lambda}$ can be determined by closed-loop system equation.

The first m rows of $(E^T E + \overline{\lambda})^{-1} E^T$ are defined as $e^T = [e_1, \ldots, e_N]$, and the first m rows of $(E^T E + \overline{\lambda})^{-1}$ are defined as $h^T = [h_1, \ldots, h_N]$.

$u(t)$ can be represented as

$$\begin{aligned} u(t) = {} & [e_1, \ldots, e_N] \\ & \times \left[RW - G(t) - L\Delta u(t-1) - \widetilde{M}\Delta u(t-1)\right] \\ & - \left[h_1\widetilde{\lambda_1} + \cdots + h_N\widetilde{\lambda_N} z^{N-1}\right] \Delta u(t). \end{aligned} \tag{17}$$

Substituting above formula into (3), we obtain the closed-loop system equation:

$$\begin{aligned} & \left\{\left[I + z^{-1}\left(e_1 L_1 + e_2 L_2 + \cdots + e_N L_N z^{N-1}\right)\right] A \right. \\ & \left. + z^{-1} D\overline{B}\left[e_1 G_1 + \cdots + e_N G_N\right]\right\} \Delta y(t) \\ & = D\overline{B}\left[e_1 r_1 + e_2 r_2 z + \cdots + e_N r_N z^{N-1}\right] \omega(t) \\ & \quad - D\widetilde{T}\Delta u(t-1) + VC\xi(t), \end{aligned} \tag{18}$$

where $\widetilde{T}$ indicates the mutual coupling part

$$\begin{aligned} \widetilde{T} = {} & \overline{B}\left[z^{-1}\left(e_1\widetilde{M_1} + \cdots + e_N\widetilde{M_N}\right)\right. \\ & \left. + \left(h_1\widetilde{\lambda_1} + \cdots + h_N\widetilde{\lambda_N} z^{N-1}\right)\right] \\ & - \left[I + z^{-1}\left(e_1 L_1 + \cdots + e_N L_N\right)\right]\widetilde{B}. \end{aligned} \tag{19}$$

According to (18), the coupling of closed-loop system is decoupled if and only if $\widetilde{T} = 0$. Because the number of variables is less than that of equations, both $\widetilde{M_j}(z^{-1})$ and $\widetilde{\lambda_j}$ of (19) can be obtained by least squares method; consequently $\widetilde{T}$ is not equal to zero exactly, and further decoupling is approximate.

Moreover, controlled object of formula (1) is CARMA model. Since there is no steady error in outputs of closed-loop system, it is necessary to determine the matrix r_j of the performance index (4). To be simplified, let $r_1 = r_2 = \cdots = r_N = r$; we can obtain r from formula (18):

$$\begin{aligned} r = {} & \left(e_1 + \cdots + e_N\right)^{-1} \\ & \times \left\{\vec{B}(1)^{-1}\left[I + e_1 L_1(1) + \cdots + e_N L_N(1)\right] A(1) \right. \\ & \left. + e_1 G_1(1) + \cdots + e_N G_N(1)\right\}. \end{aligned} \tag{20}$$

After substituting $\widetilde{\lambda_1}$, $\widetilde{M_j}$, and r_j into (16), the following law of decoupling space can be derived:

$$\begin{aligned} \Delta u(t) = {} & [I, 0, \ldots, 0]\left[E^T E + \overline{\lambda} + \widetilde{\lambda}\right]^{-1} \\ & \times E^T\left[RW - G\Delta y(t) - L\Delta u(t-1) - \widetilde{M}\Delta u(t-1)\right]. \end{aligned} \tag{21}$$

Generalized predictive control based decoupling algorithm is as follows.

Step 1. $B(z^{-1})$ can be divided into $\overline{B}(z^{-1})$ and $\widetilde{B}(z^{-1})$ using (2).

Step 2. Least square method on $\widetilde{M_j}(z^{-1})$ and $\widetilde{\lambda}$ can be computed out using (19).

Step 3. r_j can be calculated using (20).

Step 4. $\Delta u(t)$ can be derived by formula (21).

$u(t)$ is the predicted value of individual virtual machine, after decoupling, on the virtualized platform of cloud computing.

5.4.1. Decoupling Algorithm for Public Cloud. The parameters used above are known in the case the user of the virtual machine is fixed, while the aforementioned algorithm with decoupling is not applicable where the users are not fixed. For example, the users in public cloud computing are not fixed, so that the parameters related to users' behavior are unknown. In such public cloud computing, it is necessary to use parameter estimation to obtain the appropriate parameters of the corresponding controlled object and then conduct the predictive control algorithm mentioned above.

Δ left multiplies with (1); we have

$$\Delta y(t) = Q\left(z^{-1}\right) Y(t) + C\left(z^{-1}\right)\xi(t), \tag{22}$$

where

$$\begin{aligned} Q\left(z^{-1}\right) = {} & \left[A\left(z^{-1}\right) - I, G\left(z^{-1}\right)\right], \\ Y(t) = {} & \left[-\Delta y(t-1), -\Delta y(t-2), \ldots, -\Delta y\left(t-n_A\right)\right., \\ & \left.\Delta u(t-1), \Delta u(t-2), \ldots, \Delta u\left(t-n_B\right)\right]. \end{aligned} \tag{23}$$

Formula (22) is a multivariate linear equation; we solve $Q(z^{-1})$ and $G(z^{-1})$ by the least squares method.

However, $Q(z^{-1})$ may change slowly with time; a typical equation is

$$G\left(z^{-1}, t\right) = G\left(z^{-1}, t-1\right) + \frac{M(t-1)\, q(t)}{\rho + M(t-1)^T M(t-1)}, \tag{24}$$

where $0.95 \le \rho \le 1$ is the forgetting factor and $q(t) = y(t) - Q(z^{-1})Y(t)$.

Then the method to deal with (22) may become very complex.

In this paper, to solve the newest (z^{-1}), we introduce the least squares method with weighs.

That is, $Q(z^{-1})$ satisfies

$$F = \min \sum_{i=1}^{L} \lambda_i q^2 (i), \tag{25}$$

where L is the size of the sample space and $\lambda_i \geq 0$ is the weight satisfying $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_L$. Let the derivative of F be zero. We obtain $Q(z^{-1})$. To reduce the error, we can construct $\{u^*(t)\}_{1\leq t\leq L}$, $\{y^*(t)\}_{1\leq t\leq L}$ as follows.

Let k, L be integers; to obtain $G(z^{-1})$, we choose a series of $\{u(t)\}_{1\leq t\leq kL}$, $\{y(t)\}_{1\leq t\leq kL}$ and construct $\{u^*(t)\}_{1\leq t\leq L}$, $\{y^*(t)\}_{1\leq t\leq L}$ as follows:

$$\begin{aligned} \Delta u^* (t) &= \frac{1}{k} \sum_{i=1}^{k} \Delta u (Lt + i), \\ \Delta y^* (t) &= \frac{1}{k} \sum_{i=1}^{k} \Delta y (Lt + i). \end{aligned} \tag{26}$$

Since the mean value of $\xi(t)$ is 0, then we have

$$\frac{1}{k} \sum_{i=1}^{k} \xi (mt + i) \approx 0. \tag{27}$$

Then using the least squares method with weighs on $\{u^*(t)\}_{1\leq t\leq L}$, $\{y^*(t)\}_{1\leq t\leq L}$, we obtain $Q(z^{-1})$.

5.5. Deviation Control. The behavior of the virtual machine can be mapped to a point in the space that consists of eight behavioral measurement points. The model of behavioral trusted measurement can determine whether the behavior of the virtual machine is out of security border, that is, whether the behavior is a malicious one. Mathematically, the aforementioned is to obtain the distance between two points in 8-dimensional space that consists of 8 behavior measurement points. This is actually a problem to calculate the similarity between two different objects.

Similarity calculation is widely used in the intrusion detection technology and other technologies. The typical solutions are like inner product, Dice coefficient, cosine function, and Jaccard coefficient method [28].

In this paper, gray correlation analysis is adopted to calculate the deviation value. Because the predictive value of the virtual machine behavior is unknown, the historical and present behavior of the virtual machine is consistent with the information, so that this known information and corresponding location information constitute a gray system [29]. At present, the gray system theory has been extended to many fields such as the industrial, agricultural, social, economic, energy, geology, and petroleum, successfully solving a large number of practical problems in production, living, and scientific research and making remarkable achievements. The gray relational analysis is a branch of the gray system theory.

The basic idea of gray relational analysis is to determine whether they are similar to each other by the degree of similarity of curve geometry composed of the appropriate data sequence. In terms of mathematics, gray correlation degree is used here to reflect the degree of similarity. The closer the two curves are, the greater the degree of correlation of the two corresponding data sequences is, and vice versa. When it comes to specific analysis, it is desirable to replace unlimited convergence curve with approximate convergence (data array), so as to provide a great convenience in the case of dealing with a large number of practical problems.

Combined with the characteristics of a distributed computing environment based on virtual architectures, Grey Relational Analysis is adopted in this paper, and the specific calculation steps are as follows.

(1) According to the measurement point of the behavior of the virtual machine, to create the reference sequence of a virtual machine behavior, suppose n data sequences can form the following matrix:

$$\left(X_1', X_2', \ldots, X_n'\right) = \begin{pmatrix} x_1'(1) & x_2'(1) & \cdots & x_n'(1) \\ x_1'(2) & x_2'(2) & \cdots & x_n'(2) \\ \vdots & \vdots & \vdots & \vdots \\ x_1'(8) & x_2'(8) & \cdots & x_n'(8) \end{pmatrix}. \tag{28}$$

8 indicates the number of behavioral measurement points while n represents the time series:

$$X_i' = \left(x_i'(1), x_i'(2), \ldots, x_i'(8)\right)^T, \quad i = 1, 2, \ldots, n. \tag{29}$$

The data sequence is known as the reference sequence that can reflect the characteristics of the behavior of the system. The data sequence is known as comparison sequence that is composed of the factors that affect behavior of the system.

(2) The goal of the behavior of the virtual machine decides the value of the behavioral measurement point and further determines comparison sequence that has impact on the behavior of the system.

Reference data sequence should be a standard for the comparison. Here, reference data sequence comes from the list of the behavioral characteristics, seen in Table 1, written as

$$X_0' = \left(x_0'(1), x_0'(2), \ldots, x_0'(8)\right). \tag{30}$$

(3) Nondimensionalization of the reference sequence and the comparison sequence.

Due to the fact that the factors in the system have various physical meanings, the dimensions involved in the factors are different as well. As a result, it is difficult to compare the factors so as not to obtain a correct conclusion. When it comes to Grey Relational Analysis, generally it is required to carry out nondimensionalization of the appropriate data. The methodologies commonly used for nondimensionalization are as follows, for example, equalization method and the initialization method, seen in (31):

$$\begin{aligned} x_i(k) &= \frac{x_i'(k)}{(1/8)\sum_{k=1}^{8} x_i'(k)}, \qquad x_i(k) = \frac{x_i'(k)}{x_i'(1)} \\ &i = 0, 1, \ldots, n;\ k = 1, 2, \ldots, 8. \end{aligned} \tag{31}$$

After nondimensionalization, data sequence is as follows:

$$(X_0, X_1, \ldots, X_n) = \begin{pmatrix} x_0(1) & x_1(1) & \cdots & x_n(1) \\ x_0(2) & x_1(2) & \cdots & x_n(2) \\ \vdots & \vdots & \vdots & \vdots \\ x_0(8) & x_1(8) & \cdots & x_n(8) \end{pmatrix}. \quad (32)$$

Here the initialization method is adopted to conduct nondimensionalization.

(4) In our scheme, the comparison sequence refers to the behavioral measurement vector of the virtual machine to be measured. For every behavior of the virtual machine, the corresponding absolute difference between the comparison sequence and reference sequence is calculated,respectively; that is, $|x_0(k) - x_i(k)|$, where $k = 1, \ldots, 8$; $i = 1, \ldots, n$, n is defined as the number of sampling values of the object to be measured during a given period.

(5) Calculate both $\min_{i=1}^{n} \min_{k=1}^{8} |x_0(k) - x_i(k)|$ and $\max_{i=1}^{n} \max_{k=1}^{8} |x_0(k) - x_i(k)|$.

(6) Calculation of the relational coefficient through formula (33), the coefficient of the appropriate elements between every comparison sequence and reference sequence is calculated, respectively. Relational coefficient actually represents the degree of the difference between two curves in terms of geometry. Therefore, the degree of difference can reflect the degree of relationship:

$$\zeta_i(k) = \frac{T_1}{T_2}, \quad (33)$$

where

$$\begin{aligned} T_1 &= \min_i \min_k \left|x_0(k) - x_i(k)\right| \\ &\quad + \rho \cdot \max_i \max_k \left|x_0(k) - x_i(k)\right|, \\ T_2 &= \left|x_0(k) - x_i(k)\right| \\ &\quad + \rho \cdot \max_i \max_k \left|x_0(k) - x_i(k)\right|, \end{aligned} \quad (34)$$

where ρ is identification coefficient, $0 < \rho < 1$, and usually $\rho = 0.5$.

(7) Calculation of the degree of relationship.

Because the relation coefficient reflects the degree of relationship between comparison sequence and reference sequence at each moment. So, obviously there is more than one value and these values are dispersed. Therefore, it is necessary to use one value to reflect all of relation coefficient values moment. Here the average value is chosen to represent the degree of relationship between the comparison sequence and the reference sequence. The corresponding formula is as follows:

$$r_{0i} = \frac{1}{m} \sum_{k=1}^{m} \zeta_i(k). \quad (35)$$

6. Simulation and Results

In this paper, NetLogo simulation is the use of cloud computing mode virtual machines on the virtual platform to analyze our behavior-based trust measurement program. NetLogo is based modeling and integration of multiagent simulation environment, especially for the time evolution of complex systems modeling and simulation. The test environment is Intel Core Duo 2.36 g, and 4 G memory used NetLogo win7 runs to simulate the behavior of the virtual machine on a shared virtual platform, and we measure the effectiveness of the model checking virtual machine malicious behavior tested. Here the basic parameters are shown in Table 2.

Table 2: Simulation parameters.

Items	Meanings
N	350 VMs on the physical computing node
M	Initial number of infected VM when virus outbreaks
Average-node-degree	Number of interacted VMs with given VM
r_{0i}	Deviation between predicted behaviour and expected behaviour
Time	Running time of simulation of the system
β%	Percentage of malicious VMs versus the total VMs
Recovery chance	Recovery probability of the virtual computing nodes

A major function of the proposed scheme is to detect a variety of malicious behaviors of the virtual machine. To guarantee the trustworthiness of the group as much as possible, this paper uses the successful detection rate (abbreviated as MSR) of malicious behavior to reflect the detection ability of our scheme against malicious behaviors.

Within Δt, suppose there are $b(t)$ computing nodes with malicious behavior and $a(t)$ computing nodes with trusted behavior in the system, so that β% can be described as follows:

$$\beta\% = \frac{b(t)}{a(t) + b(t)}. \quad (36)$$

This paper will simulate the attack process of "worm" virus, and then to test the effectiveness of our scheme by detecting the behavior of the "worm" virus. As a matter of fact, worm virus has the following characteristics such as breaking into antivirus software, compromising security model of the system, and implantation of Trojan into downloader. The virus typical invasion action [30] is denoted by

$$\text{Attack\_Behavior}. \quad (37)$$

According to the description of the behavior in Table 1, worm virus attacks can be abstracted as a behavioral vector:

$$\begin{aligned} &\text{Attack\_Behavior} \\ &= \left(\text{MP}_1, \text{MP}_2, \text{MP}_3, \text{MP}_4, \text{MP}_5, \text{MP}_6, \text{MP}_7, \text{MP}_8\right). \end{aligned} \quad (38)$$

In order to verify the effectiveness of the trusted measurement method of the behavior of the virtual machine here, we take the scheme without the decoupling proposed in literature [13] as contrast. In our experiments, the initial ratios of infected virtual machines are set as 30%, 50%, and 70%,

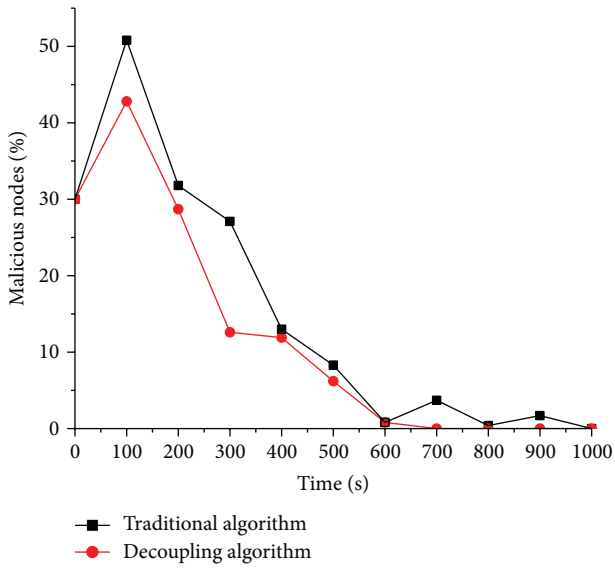

Figure 3: Percentage of the malicious suspicious virtual machine node is 30%.

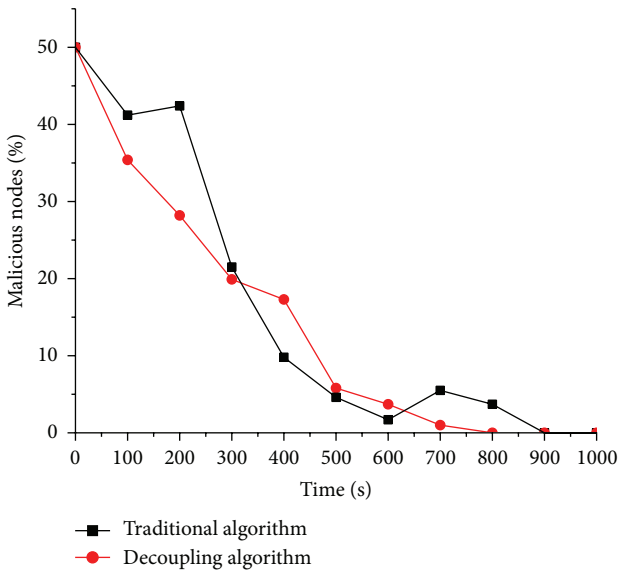

Figure 4: Percentage of the malicious suspicious virtual machine node is 50%.

respectively. For behavioral trusted measurement both with decoupling and without decoupling based on generalized predictive control, the experimental simulation are carried out three times.

When the percentages of malicious nodes are 30%, 50%, and 70%, the corresponding experimental results are shown from Figure 3 to Figure 5. After the analysis of Figures 3, 4, and 5, the following conclusions are summarized.

(1) Generally by the analysis of three figures, the simulation system for behavioral measurement model with decoupling can reach a steady state faster than the one without decoupling. The so-called steady state means such state that the number of the malicious nodes within the simulation system is 0. In our experiments, one of the parameters is the recovery chance that indicates the probability that infected node recovers as normal. In practical applications, finally the infected compute nodes recover as normal by various measurements, for example antivirus software. Faster to reach steady state means the corresponding scheme of behavioral trusted measurement is more accurate than the counterpart; that is, the user can detect and stop the spread of malicious worm virus timelier.

(2) In Figures 3, 4, and 5, the red line (decoupling algorithm) is almost below the black line (traditional algorithm), which indicates that, at any time, the scheme with decoupling proposed here can help accurately reflect the trusted state of the virtual machine and further take timely measurement so as to restrict the spread of the worm virus.

(3) In Figure 5, the distance between the red line (decoupling algorithm) and the black line (traditional algorithm) is larger than the previous two figures, which indicates, as the proportion of the malicious nodes in the system goes more, that the behavioral trusted measurement proposed here is better than the scheme in [13].

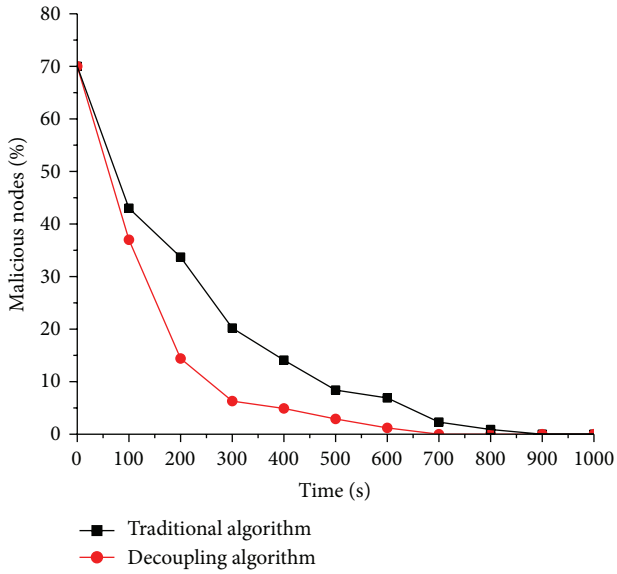

Figure 5: Percentage of the malicious suspicious virtual machine node is 70%.

In summary, the experimental simulation shows that trusted measurement scheme here can effectively predict and control the behaviors of the virtual machine. So that such attack behavior that results from the abuse of the resources in cloud computing can be found timely and well restricted; that is, the security of the entire group can be well guaranteed.

7. Conclusion

The scheme for trusted measurement over dynamic multitenant behavior in cloud computing environment put forward here addresses the problem of resource-sharing existing

in the cloud computing. By extending our previous model to the multiple tenants who share the same resource, we can further use the generalized predictive control to depict complicated behavior in cloud computing. Thanks to the advantages of generalized predictive control such as rolls optimized method and the feedback adjustment, the complicated behaviors of multitenants are well controlled. Further, the problems incurred by coupling between multitenants are solved effectively by the decoupling algorithm of generalized predictive control. As a result, the malicious behaviors between multitenants are restricted in cloud computing platform. In other words, our scheme avoids the threats introduced by multitenancy under cloud computing. In the future, we will refine our scheme and take into account the nonlinear behaviors between multiple tenants in order to deal with the behavior of tenants much more precisely.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the program Major Projects of the Wireless Mobile Communications (2012ZX03002003) and the National Science Foundation of China (61003260, 61271275).

References

[1] H. Zang, J. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, pp. 47–60, 2000.

[2] Amazon Web Services, "Creating HIPAA-Compliant Medical Data Applications with Amazon Web Services," White Paper, 2009, http://awsmedia.s3.amazonaws.com/AWS_HIPAA_Whitepaper_Final.pdf.

[3] E. Kanimozhi, "Trusted cloud—a solution for cloud cartography," *Journal of Global Research in Computer Science*, vol. 3, no. 11, pp. 44–51, 2012.

[4] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Properties of generalized predictive control," *Automatica*, vol. 25, no. 6, pp. 859–875, 1989.

[5] G. Kim and E. Spafford, "The design and implementation of tripwire:a file system integrity checker," Tech. Rep., Purdue Univiversity, West Lafayette, Ind, USA, 1993.

[6] W. Arbaugh, D. Farber, and J. Smith, "A secure andreliable bootstrap architecture," in *Proceedings of the IEEE Symposium on Security and Privacy*, IEEE Computer Society, Oakland, Calif, USA, 1997.

[7] P. Iglio, "Trustedbox: a Kernellevel integrity checker," in *Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC '99)*, IEEE Computer Society, Phoenix, AZ, USA, 1999.

[8] N. Petroni, T. Fraser, J. Molina et al., "Copilot-a coprocessor-based Kernel runtime integrity monitor," in *Proceedings of the 13th Usenix Security Symposium*, USENIX Association, San Diego, Calif, USA, 2004.

[9] E. Shi, A. Perrig, and L. Doom, "BIND: a fine-grained attestation service for secure distributed systems," in *Proceedings of the IEEE Symposium on Security and Privacy*, IEEE Computer Society, Oakland, Calif, USA, 2005.

[10] T. Jaeger, R. Sailer, and U. Shankar, "PRIMA: policy-reduced integrity measurement architecture," in *Proceedings of the 11th ACM Symposium on Access Control Models and Technologies (SACMAT '06)*, Association for Computing Machinery, Lake Tahoe, Calif, USA, 2006.

[11] X.-Y. Li, X.-D. Zuo, and C.-X. Shen, "System behavior based trustworthiness attestation for computing platform," *Chinese Journal of Electronics*, vol. 35, no. 7, pp. 1234–1239, 2007 (Chinese).

[12] C.-X. Shen, H.-G. Zhang, F. Dengguo et al., "Survey of informationsecurity," *Science in China Series:E*, vol. 37, no. 2, pp. 129–150, 2007 (Chinese).

[13] B. Gong, "The behavior measurement model based on prediction and control of trusted network," *Chinese Journal of Communication*, vol. 9, no. 5, pp. 117–128, 2012.

[14] L. Zhuang, M. Cai, and C.-X. Shen, "Hierarchical verification of behavior trustworthiness," *Journal of Beijing University of Technology*, vol. 38, no. 9, pp. 1396–1401, 2012.

[15] M. T. Khorshed, A. B. M. S. Ali, and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 833–851, 2012.

[16] X.-Y. Li, X.-L. Gui, Q. Mao, and D.-Q. Leng, "Adaptive dynamic trust measurement and prediction model based on behavior monitoring," *Chinese Journal of Computers*, vol. 32, no. 4, pp. 664–674, 2009.

[17] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 199–212, 2009.

[18] Amazon, "Amazon elastic compute cloud (AmazonEC2)," 2011, http://aws.amazon.com/ec2/.

[19] F. Rocha and M. Correia, "Lucy in the sky without diamonds: stealing confidential data in the cloud," in *Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W '11)*, pp. 129–134, 2011.

[20] A. Chonka, Y. Xiang, W. Zhou, and A. Bonti, "Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1097–1107, 2011.

[21] D. Danchev, "DanchoDanchev'sblog—mind streams of information security knowledge," 2011, http://ddanchev.blogspot.com/.

[22] J. Grossman, "Jeremiah Grossman," 2011, http://jeremiahgrossman.blogspot.com/.

[23] D. Danchev, "Coordinated Russia vs Georgia cyber attack in progress," 2008.

[24] D. Danchev, The DDoS attack against CNN.com, 2008.

[25] Grossman, Cross-site scripting worms and viruses, Whitehat-Security, 2006.

[26] M. T. Khorshed, A. B. M. S. Ali, and S. A. Wasimi, "Classifying different denial-of-service attacks in cloud computing using rule-based learning," *Security and Communication Networks*, vol. 5, no. 11, pp. 1235–1247, 2012.

[27] T. Y. Chai, K. Z. Mao, and X. F. Qin, "Decoupling design of multivariable generalised predictive control," *IEE Proceedings:*

Control Theory and Applications, vol. 141, no. 3, pp. 197–201, 1994.

[28] M. Li, X. Chen, M. L. Xin et al., "The similarity metric," in *Proceedings of the IEEE Transactions Information Theory*, pp. 863–872, 2003.

[29] J. Deng, *The Control Problems of Grey Systems*, Huazhong University of Science & Technology, 1993.

[30] B. Gong, *Trusted Network Architecture Supporting Trusted Group Establishment and Key Technologies Research*, Beijing University of Technology, 2012.