2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.2.60

A Security Model CCSM in an Environment of Cloud Computing

He Shan[+] and Lin Guoyuan

School of Computer Science and Technology, China University of Mining and Technology

Xuzhou, China

Abstract—This paper aimed at the popular cloud computing technology, analyzed the security problems it faced. A model for protecting the data's integrity and confidentiality in the cloud server was raised, which took the reference of BLP (Bell-LaPadula) model and the CW (Clark-Wilson) model, and component, security axiom and realization of the model are given.

Keywords-Cloud Computing; BL P model; CW model; CCSM model

1. Introduction

Cloud computing[1] is an emerging computing model, which is based on grid computing, it is a distributed computing. At present, the three major security problems of that cloud computing security are identity and access control, Web security protection and virtualization security and so on. Many manufacturers have concerns about permissions and management authority of cloud model. In the virtual and complex environment, how to ensure that their applications and data are still clear and controllable concerns the users, and it is also the first problem that a cloud service providers must solve.

In the era of cloud computing, data storage means breaking down the traditional model, and storing all data all data in the cloud hosting server by the way of trusteeship. Moreover, the user can access the data and services they need anytime and anywhere easily by API (Application Programming Interface) from cloud service provider and a browser. Cloud computing information system's security issues are brought by the changing of service patterns, such as opening interfaces provide the possibility of illegal access, and the illegal users can enter information systems to steal sensitive data of enterprises and users, so the integrity and confidentiality of data are facing great threat. Facing security issues of integrity and confidentiality of date in cloud servers, this paper, based on BLP and the CW model, proposes CCSM model.

2. About BLP model and the CW model

2.1. BLP model

BLP model (also known as Bell-LaPadula model) was founded by El-liott Bell and Leonard J La Padula in 1973, which is a model of computer operations simulating military security policy[2][3]. BLP model is a state machine model, which formally defines the system, system state and system transition rules between states and gives a set of security features axiom, thus it can restrict and constraint the system state and state transition rules. For a system, if its initial state is secure, and a series of rules are also secure, then the system is secure. In the BLP model, the subject is defined as an entity can initiate actions, such as process; while the object is defined as a passive bearer of the main acts, such as data, documents.

[+] Corresponding author.
E-mail address: qdshengdi@yeah.net

The model mainly consists of[4][5]:

S represents a main collection;

O represents a set of objects;

M represents an access control matrix, and each element of the matrix represents that the corresponding subject corresponds to the corresponding object attribute value;

A represents an access authority , $(r, w, a, e) \in A$, *r* represents read access, *w* represents written permission, *a* represents read and write authority, *e* represents the implementation of the authority;

$b \in \{S \times O \times A\}$ represents which subject access which objects by which authority;

F represents the security level set, *fs* represents the highest security level of the main, *fc* represents the current security level of the subject, *fo* represents the security level of the object;

$f \in F$ represents the highest security level of the current main and the security level of the object

H said that the tree structure of the current object;

In the BLP model, each subject has a security level and each object belongs to a access class, and the access class is associated with a level of security. Each subject also has a current security level, and the current security level can not exceed its initial security level. The security level of the subject must be below its initial security level assigned.

The model has the following two security features:

(1) Allow reading next. If the object's security level is lower than the current security level of the main, the subject only has reading access for the object.

(2) Allowed to write. If the security level of the object was higher than the current security level of the main, the main only has writing access to the object.

The two features will ensure one-way flow of information, meaning information can only flow to the higher security. BLP model prevents the spread of information and resists the accusation of Trojan horses through one-way flow of information. BLP model is lack of the conformity of the integrity[6]. It is possible that the authorized users modify the data illegally .

BLP model satisfies the simple security property axiom and *-Property [7].

（1）Simple Security Property, also known as SS-features.

When the main access is made to the object, the subject's maximum security level must be greater than or equal to the object's security level. The state $v = (b, M, f, H)$ satisfy the SS-characteristics, to all $((s, o, x) \in b, x \in A)$, if and only if following was established:

1）$x=a$ or $x=e$

2）$x=w$ or $x=r$ and $fs(s)=fo(o)$

This indicates that only when the security level of a subject dominated the security level of another object, the subject has "read" or "write" access to this object. The purpose of this rule is to prevent the main reading the information of object whose security level is higher than the main, preventing the main access to information of the object which has a higher level.

（2）*-Property

To prevent the flow of information flows from the high security level to lower security level. *s* is a subset of *S*, state $v = (b, M, f, H)$ satisfy the * attribute, if and only if for all $s \in S$, there

1) $O \in b(S{:}a) \Rightarrow (fo(O) > fc(S))$

2) $O \in b(S{:}w) \Rightarrow (fo(0) = fc(S))$

3) $O \in b(S{:}r) \Rightarrow (fo(0) < fc(S))$

*-Property is used to prevent illegal flows of information caused by untrusted subjects from higher security level to lower level. An untrusted subject has "add" (append) permission for an object, only if the current security level of the object dominates the current security level of the main; an untrusted subject has "write" permission for an object, only if the security level of this object is equal to the main's current security level; an untrusted subject has "read" permission for an object, only if the security level of this object is dominated by the current security level of the subject. For example, when a user process operate in the top

secret level or in different categories , the user can not "write" or "add" a secret-level document. * Property has no limit on the credibility of the main. For untrusted subject, * attributes includes a simple security that meets the * attribute, and it must satisfy the simple security.

Despite this, two rules are necessary, because simple security is required for all subjects, but * Property is required only by the untrusted subject . The flow of information in the system is controlled when these two rules are satisfied. * Property prevents a user intentionally or unintentionally writing information from a file of higher security classification to lower security classification, causing leakage.

2.2. CW model

As early as 1980, Lee[8] and Schockley[9] talked about if CW model can be used for access control. Unlike BLP mode, CW model emphasis on unauthorized modification of information in many fields as business and others.

CW model mainly consists of three parts :

(1) Data, that the object set. In the CW, the system data is divided into two parts: constrained data item(CDI) and unconstrained data item(UDI).It is ensured that CDI has integrity constraints, UDI has no the integrity constraint ;

(2) Integrity Verification Procedure. The process used to verify data integrity, which is executed by the system's "security officials" (security officials do not implement TP);

(3)Transformation Procedure. The process changes the CDI from one valid state into another. (the so-called effective state is the state that the data is being bound). The process is implemented by the general user. TP is the access method from the main to the object.

CW model with the principles of integrity and separation of duties.

(1) The principle of separation of duties. Provides a task from beginning to end, can not be completed by one person, and this task will be distributed to at least two people in order to prevent possible individual cheating.

(2) Integrity principle. Users can not handle data arbitrarily, but they can by the methods of ensuring data integrity.

In this model, data operating unrelated to the level of security, it mainly focus on preventing illegal operations of data instead of the confidentiality of information, therefore prone to information leakage.

3. CCSM（Cloud Computing Security Model） Model

Not difficult to see from the above analysis, BLP model is the most classic multilevel security policy model in secure operating system, but it emphasizes too much on system security, but not enough considering on integrity and authenticity of the system. CW model is the integrity business model with the strong data types for the transaction, but the confidentiality control of the system is not strong.

Therefore, in order to protect the confidentiality and integrity of data in the cloud server, this paper presents a cloud computing security model based on BLP and CW, that is CCSM model. This model is mainly inherited simple and secure property and *-Property of the BLP model for strengthening the confidentiality of data, combined with the integrity of the verification process and the conversion state of the CW model, thus ensuring integrity of data.

3.1. CCSM model components

1) Element

To facilitate the model's formal description, the model elements are defined in a mathematical form, as shown in Table Ⅰ:

TABLE I: THE COMPOSITION OF THE ELEMENTS FOR CCSM MODEL

element set	element	explain
S	*{s1，s2，…，sn}*	Cloud users set
O	*{o1，o2，…，on}*	Collection of cloud server

D	*r said read access, w expressed written permission, a said read / write access, e said the implementation permission (can not see and modify) Note: To prevent a low security classification information users tampering with read access is greater than or equal to written permission.*	Access *permission*
F	*F={f,f'}* *fo is security level of the cloud user* *f is current security level of the cloud user* *f' is highest security level of the cloud user*	Security level set
type	*type(o,s)* said the correspondence relationship of the cloud user s and the cloud server o	*access function of s to o*

2) State transition rules

State transition rules are primarily intended to ensure that every state of the system is secure. In addition to this, it also needs to ensure every conversion of the system is from a secure state to another secure state. There are mainly get-read rules, get-append rules, get-execute rules, get-write rules, change-subject- current-security-level rules in CCSM model.

Get-read rules: using for requesting a read-only access that is from a collection of the cloud users to the collection of the cloud server.

When the cloud user *Si* has read-only access to the cloud server *Oj*, it must meet the following conditions：

- *Si* has read-only permissions on the *Oj* in access properties;
- *Si*'s security level dominated *Oj*'s;
- *Si* is the trusted cloud user or the cloud user's current security levels dominate the security level of the cloud server *Oj*;

get-write rules: requests only write access from the cloud users to the cloud server.

When the cloud users *Si* has written-only access to the cloud server *Oj*, it must meet the following conditions：

- *Si* has write-only permissions on the *Oj* in visit properties;
- *Si* is a trusted cloud user or the current security level of the cloud user dominates the security level of *Oj*;

get-execute rules: using for requesting a execute access that is from a collection of the cloud users to the collection of the cloud server

When the cloud users *Si* has execute access to the cloud server *Oj*, it must meet the following conditions:

- *Si* has execute permissions on the *Oj* in access properties, read permissions is necessary for completing the operation of the implementation.

get-append rules: using for requesting a read and write access that is from a collection of the cloud users to the collection of the cloud server

When the cloud users *Si* has read and write access to the cloud server *Oj*, it must meet the following conditions:

- *Si* has read and write permissions on the *Oj* in access properties;
- *Si*'s security level dominated *Oj*'s;
- *Si* is the trusted cloud user or the cloud user's current security levels dominate the security level of the cloud server *Oj*;

change-subject-current-security-level rules: using for requesting to change the current security levels.

When *Si* can change the *Oj*'s current security levels into *fo*, following conditions must be met:

- *Si* is a trustful cloud user or its security level has been changed to *fo* and lead the state to meet the * property;
- the security level of *S* dominates *fo*;

3) Integrity Verification Procedure

The process used to verify integrity of data and user's status.

4) Data items

That information set went into from cloud user to the cloud server.

5) transformation procedure

This process is to change data or Cloud users from a valid state into another valid state. The so-called effective state is that the data or the user is in the constrained state. This conversion process can also be interpreted as a way of access from the cloud users to the cloud server data set.

3.2. security Theorem of CCSM model

Theorem 1: CCSM model satisfies the simple and secure properties and * attributes of classical BLP model.

Proof: Suppose $\leqslant BLP$ and $\geqslant BLP$ expresses partial order, *Lw* is the write access of minimum safety level corresponding to $\geqslant BLP$, and *fo* is corresponding to the current security level of *o*.

（1）for non-credible cloud users *S*

$\forall o \in O , x = F(o)$;

$\forall s \in S$, So read permission set of s is the Drs, and its write permissions set is the Dws ;

The access set of S is $f'e , fr$ and fw ， $D (s) =\{f'e, f r , fw\}$， then

$Drs = \{ (o, r) \mid \exists for \leqslant fr [((o, r) ,for) \in D]\}$,

$Dws = \{ (o, w) \mid \exists fow \leqslant fw [((o, w) , fow) \in D]\}$,

Also because the access of every cloud server o is precisely assigned to the role $F (x) r$ and $F (x) w$,

That $D= \{ ((o, r) , f'(o) r) , ((o, w) , f'(o) w) \}$

Deduction：$Drs = \{ (o, r) \mid \exists for \leqslant fr[fo= f'(o)]\}$

$= \{ (o, r) \mid f'(o) r \leqslant fr\}$

$= \{ (o, r) \mid f'(o) \leqslant BLP f\}$,

$Dws = \{ (o, w) \mid \exists fow \leqslant fw[fo=f'(o)]\}$

$= \{ (o, w) \mid f(o) w \leqslant fw\}$

If s had read access $= \{ (o, w) \mid f'(o) \geqslant BLP f\}$.

o ,that $(o , r) \in Drs$, there must be $f'(o) \leqslant BLP f= f(s)$, that meets * property;

And because $f \leqslant BLP f' =f'(s)$, so $f'(o) \leqslant BLP f'(s)$, that satisfy the simple security property;

If *s* had read access to *o*，that $(o , w) \in Dws$, there must be $f'(o) \geqslant BLPf = f(s)$, that meets * property;

And because the simple security property is not bound to only write, it also satisfies the simple security property.

(2) For the credibility cloud user *S'*

$\forall o \in O , x = F(o)$， $\forall s \in S'$, So the read permission set of s is the *Drs*，Write permissions set is the *Dws*,

So the authority set of *S* is $f'e, fr$ and *Lw*, that $F(s) =\{f'\ e , fr , Lw\}$, Similarly with the last one，

So $Drs = \{ (o, r) \mid f'(o) \leqslant BLP f\}$，

$Dws = \{ (o, w) \mid f'(o) \geqslant BLP f \}$.

If *s* has access to *o*, that $(o , r) \in Drs$, there must be $f'(o) \leqslant BLP f= f(s)$，

And because $f \leqslant BLP f'= f'(s)$, so $f'(o) \leqslant BLP f'(s)$, satisfy the simple security property。

For the trustful cloud users, the visit to them from * property is naturally satisfied without any constraints.

In summary, CCSM model meets simple and secure property and property axioms * of the BLP model.

In CCSM model, a simple security attributes of model is required to be established for all users, that is to prevent the user from reading the cloud server information whose security level is higher than its current security level, and to prevent the cloud users from accessing directly cloud server access information without allowed level. The * property axiom only be required to be set up for the non-credible cloud users, this property of cloud prevents leakage caused by the user's intentional or unintentional writing of higher security level information into cloud server of lower security level.

3.3. The implementation of CCSM model

Without loss of generality, we assume that in this model system administrator defined k-level of security for the system*(1,2,3, b, ..., k),* and security level orders from high to low, in which 1 represents the highest level of security, k represents the lowest level of security. Cloud server O in the system has a certain level of security $f(o)\ (1 \leqslant f (o) \leqslant k)$,the cloud user s divided into several types, and each user belongs to a type , assuming that each type corresponds to a security level $f (z)\ (1 \leqslant f (z) \leqslant k)$,then the security level of a cloud user belonging to the type is $f (s) = f (z)$. The corresponding relationship is set up between task *T* and the cloud user *s*. Suppose a task *T* corresponds to a cloud user *s*, the security level of the cloud user $f (s) = n\ (1 \leqslant n \leqslant k)$, after the task is divided into sub-tasks $T (i), T = (T (i), a \leqslant i \leqslant m \text{ and } m \geqslant b)$, the cloud server set completing the task *T O = (Oi, a⩽i ⩽m and m ⩾b)*, a cloud server's security level is *f (Oi)*, when the acts between o and s is F, the relationship between *f (s)* and *f (Oi)* can be represented by M-matrix as Table Ⅱ shows.

TABLE II: M MATRIX

f(s) O *F*	*O1*	*O2*	*O3*	……	*On*
r	≤*n*				
w		≥*n*			≥*n*
a			=*n*	……	
e					≤*n*

The overall structure of the model is showed in Figure 1, the composition of the basic realization as the following steps.

The first step: Each cloud user who applies for the access to the cloud server must verify the legal status through IVP;

The second step: the cloud user with certification is authenticated;

The third step: Each cloud user with certification is gave by the role through TP transformation;

The forth step: the cloud user that has been given the role is authorized to perform, and they access the data of the cloud server according to the rules he has to abide;

The fifth step: Each data that will be deposited into the cloud server must be go through the IVP integrity validation and TP transformation, this step ensures the data integrity.

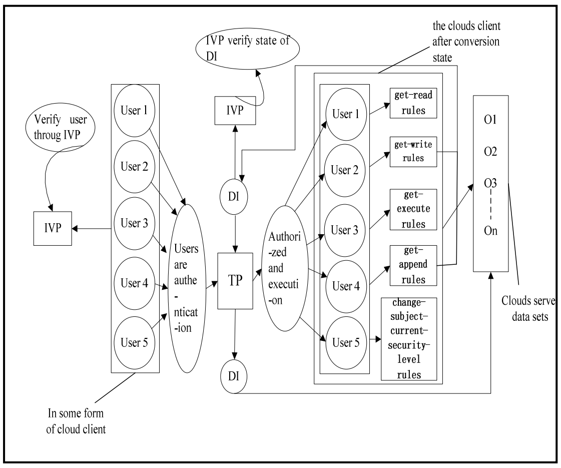

Fig.1.CCSM model of integrity and confidentiality rules

3.4. the application of CCSM model in the cloud computing

The key lies in the word "cloud" in the definition of cloud computing. For us, cloud is a large number of computers connected together. These computers can be a personal computer or network server, and they can be public or private. This computer cloud could extend beyond a separate company or enterprise.

Applications and data provided by Cloud can be widely used for many users, it can be cross-enterprise and cross-platform. Accessing to the cloud has been done via internet , and any authorized user can access these documents and applications from any computer through internet.

The following are application example of CCSM model in cloud environment, the basic process as Figure 2 shown.

```
1)  the cloud User s requests for cloud services
2)  system verifies IVP for S
3) This cloud user legally performs the following procedures, otherwise be directly forced quitting system.
4)  If（s∈S）
5)  {
```

```
If(s is the trusted cloud user)
{If($F(s)>F(o)$ )  // the security level of S dominates o's
    {
        If($o(r) \in A(s)$)
// S has read-only access to o in its properties.
        { System goes state transition TP to the user s
              $type(o,s)=A(r)$,
// said read access of S distributed by system,
        At this point, clouds users can watch movies and see pictures in the clouds through the implementation of get-read rule    }
        If ($o(r,v) \in A(s)$)
        // S has read and write access to o in its properties.
        {
System goes state transition TP to the user s, $type(o,s)=A(r,w)$
// said read and write access of S distributed by system
       At this point, clouds users can read and modify the documents in the clouds through the implementation of get-append rule
    and so on  }
If($o(e)$) // S has execution access to o in its properties.
        { System goes state transition TP to the user s, $type(o,s)=A(e)$
// said execution access of S distributed by system
        At this point, clouds users can browse the game in the clouds through the implementation of get-execute rule
        } } } }
Else s is the non-trusted users
{
    If(the cloud user meets the * Properties)
    {  If($F(s)>F(o)$)
     // the security level of S is higher than that o
   { System goes state transition TP to the user s, and execute
  get-write rules, The cloud user can write into cloud information    }
else if($F(s)=F(o)$)   // the security level of S is equal to o
   { System goes state transition TP to the user s ,and execute
  get-append rules, the cloud users can read and modify
   the documents in the clouds  }
      else
     { System goes state transition TP to the user s, and execute
  get-read rules, the cloud users can watch movie and so on     }  }  }
```

Please note, if the cloud user s modified some files of cloud and save them, he have to call the model IVP integrity verification and TP converter again before his saving.

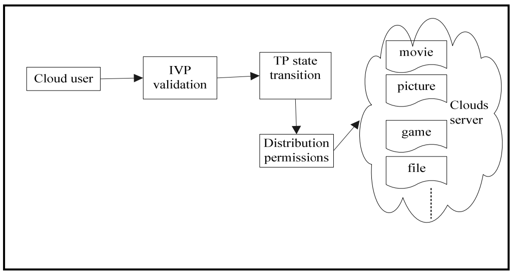

Fig.2. Application of CCSM model in the cloud environment

4. Conclusion

The security is one of the cores of cloud computing. In the era of cloud computing, the deployment of safety equipment and security measures are different, and the main of security responsibility has changed. This paper, under the security conditions that cloud computing faces, raises CCSM model which ensures integrity and confidentiality of data based on the BLP model and the CW model， and presents its components ,security theorems and the typical implementation process. The advantages of this model is that it combines the characteristics of strict confidentiality of the BLP model with the integrity features of the CW model. The further research of CCSM model is to improve "next time, read" feature of the model to improve availability of the system.

5. References

[1] Michael Miller. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, 2009.4

[2] Nat'l Computer Security Center. Trusted network interpretation of the trusted computer system evaluation criteria. NCSC-TG2005 ,1987

[3] Bell DE, La Padula LJ. Secure computer systems: Mathematical foundations. ESD-TR-73-278, I (AD) 770 768, Electronic Systems Division, Air Force System Command, Hanscom AFB. Bedford, 1973.

[4] Bell DE, La Padula LJ. Secure computer systems: A mathematical model. ESD-TR-73-278, II (AD) 771 543, Electronic Systems Division, Air Force System Command, Hanscom AFB. Bedford, 1973.

[5] ShiWenChang, SunYuFang, LiangHongLiang. Classic BLP safety axiom of an adaptive method and its correctness. Computer research and development,2001, pp.1366-1372

[6] Lin T Y. Bell and LaPadula axioms : A"new"paradigm for an"old"model . In : Proc. 1992 ACM SIGSAC New Security Paradigms Workshop.

[7] LiangBin, SunYuFang, ShiWenChang, SunBo. An Improved Method to Enforce BLP Model and Its Variations in Role2Based Access Control [J]. Journal of computer, 2004,pp.637-638

[8] Little Compton ,Rhode Island ,USA ,1992. 82～93Lee T., "Using mandatory integrity to enforce "commercial"security". 1988 IEEE Symposium on Security and Privacy. Oakland,CA, PP. 140-146,1988.

[9] Schockley W, "Implementing the Clark/Wilson integrity policy using current technology". In Proceedings of the 11th NationalComputer Security Conference, Gaithersburg. MD PP.29-37, 1988.