2011 International Conference on Computer Science and Information Technology (ICCSIT 2011)
IPCSIT vol. 51 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V51.18

Research on Technologies of File Security Label Management

Li Bian*, Xingyuan Chen, Yunan Liu, Yongwei Wang and Wenchong Xie [a]

[a] Zhengzhou Information Science and Technology Institute, Zhengzhou 450004, China

Abstract. As the basis of mandatory access control, the security label itself is a kind of sensitive information and also has been drawn attention increasingly. Nowadays, files in operating system are massive. So it's urgent to resolve the problems that how to manage such a large number of security labels and ensure their security during generation, transmission and application. To resolve the security problems during the process of transmission and modification of security labels, the paper proposes the technologies of file security label transmission based on PKI and file label modification based on multi-dimensional domain. The analysis shows that the two technologies can resolve the security problems.

Keywords: Security label, PKI, Label management, Label transmission, Label modification

1. Introduction

To achieve hierarchical access control to files, file security attributes must be labeled. The file security label includes some security attributes, such as the security classification, category and so on. Then the label will be embedded in files through some embedding algorithms. During the process of access and transmission, hierarchical access control to files must be achieved by the comparison of security labels based on the requirements of multi-level security (MLS) policies. Security labels provide the basis for determining the sensitivity of files [1].

Nowadays, files in operating system are massive and the security label itself is a kind of sensitive information. So it's urgent to resolve the problems that how to manage such a large number of security labels and ensure their security during generation, transmission and application. Currently there are a few researches on the label management and only the label management processes in the operating system are briefly introduced in some literatures [2][3], which is not related to the security of the security label itself.

As an important part of hierarchical access control to files based on the label, the label management is aimed as follows:

- Ensure the confidentiality of label information

Security label is a kind of sensitive information, so its confidentiality must be ensured during the storage and transmission process. When the label is transmitted in the network, it's not allowed to identify its contents for illegal users and only the authenticated users can read. When the label is bounded to the file, its content must be identified by a specific identification program.

- Ensure the integrity of label information

When the label is been processed or transmitted, its contents aren't allowed to be arbitrarily tampered. When the label is bounded to the file, any operations to the file should not affect the contents of the security label. Furthermore, the label should not be deleted.

- Ensure the non-repudiation of label information

When the label is transmitted to the end user, it cannot be denied that the security label has been received by the end user and has been sent by the sender.

* Corresponding author. Tel.: +86-15603391006.
E-mail address: beleson@163.com.

- Ensure the security of the system during any modifications to labels

If there are needs to modify the contents of the security label, it must not affect the security of the system.

Aimed at the problems of label transmission and modification, this paper studies the security rules that the labels must be followed and the security of labels themselves during their transmission based on analyzing the life cycle of the security label.

2. The Technology of File Security Label Transmission Based on PKI

A label management system consists of a LMC (Label Management Center) and an APC (Application Processing Center). Its goal is to process user requests on security labels and ensure the security of labels in their life cycle. When the EU (End User) applies for a label to the LMC, the APC verifies and processes the request. If verification is success, the request will be sent to the LMC. Then LMC generates the security label and transmits the label to the EU. Because of the openness and non-reliability of the network, the security label during transmission is vulnerable to interception, tampering, forgery and other attacks. Network congestion can also easily cause the loss of the security label. Therefore it's necessary to take measures to ensure the security and reliability during security label distribution.

PKI is a security infrastructure to provide authentication, encryption, digital signature and other security services for network users [4]. The purpose is to ensure the security of the network communication through the technology of public key and authentication mechanism. The authentication, confidentiality, integrity and non-repudiation of the network communication entity can be ensured by PKI.

2.1. The Establishment of Secure Transmission Channel

Before the LMC transmits the security label to the EU, a secure transmission channel must be established to ensure the security during transmission. The process of establishing the secure transmission channel is shown in Fig. 1.

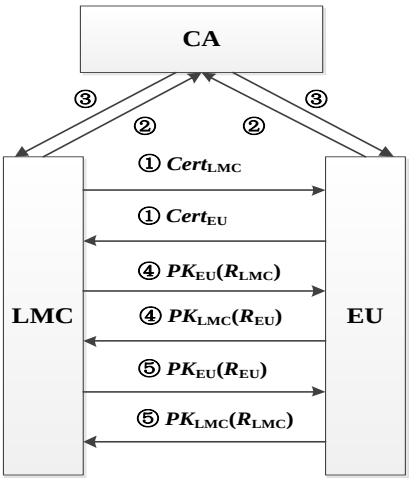

Fig. 1. The process of establishing the secure transmission channel

① LMC sends its own identity certificate $Cert_{LMC}$ to EU and EU also sends its own identity certificate $Cert_{EU}$ to LMC.

②After LMC and EU receive each other's identity certificate, CA verifies the legitimacy of the two sides' certificates with its signature public key.

③ CA sends the verification results to LMC and EU. If the verification result is valid, then skip to ④. Otherwise, require the other to update the certificate and the communication process should be suspended.

④ EU extracts the public key PK_{LMC} from the LMC's identity certificate. Meanwhile LMC extracts the public key PK_{EU} from the EU's identity certificate. LMC generates a random number R_{LMC}, which is encrypted by PK_{EU}, and sends the result $PK_{EU}(R_{LMC})$ to EU. Meanwhile, EU generates a random number R_{EU}, which is encrypted by PK_{LMC}, and sends the result $PK_{LMC}(R_{EU})$ to LMC.

⑤ When $PK_{EU}(R_{LMC})$ has been received by EU, EU uses its own private key to decrypt it to get the random number R_{LMC}, then uses the LMC's public key PK_{LMC} to encrypt R_{LMC}, and sends the result $PK_{LMC}(R_{LMC})$ to LMC. At the same time, when $PK_{LMC}(R_{EU})$ has been received by LMC, LMC decrypts it to get the random number R_{EU} with its own private key, then encrypts R_{EU} with the EU's public key PK_{EU} and sends the result $PK_{EU}(R_{EU})$ to EU.

⑥ When $PK_{EU}(R_{EU})$ has been received by EU, EU decrypts it to get the random number R_{EU} with its own private key and compares it with its generated random number previously. At the same time, When $PK_{LMC}(R_{LMC})$ has been received by LMC, LMC decrypts it to get the random number R_{LMC} with its own private key and compares it with its generated random number previously. If the comparison results are all the same, the negotiation and exchange of the session key between the two sides is conducted. Right now, a secure transmission channel between LMC and EU is established. Otherwise, it an error message is returned and the communication process is suspended.

2.2. The Secure Transmission of Security Labels

When the secure transmission channel has been established, LMC can transmit the security label to EU through the channel. The confidentiality and integrity of security labels during transmission through network can be ensured by security services provided by PKI. The process is shown as follows:

- Generate the label header *H*. *H* mainly stores the security parameters that should be determined by both the sender and receiver. These parameters mainly include the unique identification, active time and so on.

$$ID_{LMC}||ID_{EU}||T_{Start}||T_{End} \tag{1}$$

- Generate a session key *k* for the secure communication using the Diffie-Hellman protocol [5].
- Extract the serial number CID_{EU} and the public key PK_{EU} from $Cert_{EU}$, and protect the session key *k* with PK_{EU}.

$$CID_{EU}||\{k\}PK_{EU} \tag{2}$$

- Encrypt the label header *H* and the label content *M* to generate $\{H\}k$ and $\{M\}k$.
- Link the contents above, hash the linked result and sign the hashed result using the LMC's private key K_{LMC}.

$$[\text{Hash}(ID_{LMC}||ID_{EU}||\{H\}k||CID_{EU}||\{k\}PK_{EU}||\{M\}k)]K_{LMC} \tag{3}$$

- Link all the results above, form the label to be transmitted through network and send it to EU.

$$ID_{LMC}||ID_{EU}||\{H\}k||CID_{EU}||\{k\}PK_{EU}||\{M\}k||[\text{Hash}(ID_{LMC}||ID_{EU}||\{H\}k||CID_{EU}||\{k\}PK_{EU}||\{M\}k)]K_{LMC}$$

It can be simply expressed as:

$$ID_{LMC}||ID_{EU}||\text{Crypt0}||\text{Crypt1}||\text{Crypt2}||\text{Crypt3} \tag{4}$$

Crypt0 indicates the encryption result of the label header $\{H\}k$. Crypt1 indicates the protected session key $CID_{EU}||\{k\}PK_{EU}$. Crypt2 indicates the encryption result of label contents $\{M\}k$, Crypt3 indicates the signature result in the step (5).

2.3. Identification of Security Labels on EU

When EU has received the security label sent from LMC, he must identify it and extracts its contents. Assume that EU has received the message as follows:

$$ID_{LMC}'||ID_{EU}'||\text{Crypt0}||\text{Crypt1}||\text{Crypt2}||\text{Crypt3} \tag{5}$$

The process of identifying the message is as follows:

- EU verifies that whether ID_{LMC}' is equal to ID_{LMC} and ID_{EU}' is equal to ID_{EU}. If it is not true, an error message is returned to LMC.
- EU hashes $ID_{LMC}'||ID_{EU}'||\text{Crypt0}||\text{Crypt1}||\text{Crypt2}$ with the same hash function and decrypts Crypt3 with LMC's public key to PK_{LMC}. Compare the hashed result and the decrypted result of Crypt3. If they are not equal, then the label has been tampered and EU returns an error message to LMC.
- EU reads the corresponding CID_{EU} in Crypt1, and gets the session key *k* with its own private key K_{EU}.
- EU decrypts Crypt0 with *k* to get the label header. Verify the validity of T_{Start}' and T_{End}'. That is, determine whether T_{Start}' is earlier than the current time. If not, the received label is not correct; Then determine whether T_{End}' is later than the current time. If not, the received label has expired.

- EU decrypts Crypt2 with *k* to get the plain-text content of the security label.

2.4. Security Analysis

2.4.1. Confidentiality

The session key *k* is generated according to the Diffie-Hellman protocol. During the generation of the session key, it can resist the man-in-the-middle attacks as both sides have confirmed the identity of each other before communication, LMC encrypts the contents of the security label with *k* and protects *k* with the EU's public key PK_{EU}. It effectively prevents the security label from the risk of illegal stealing during the transmission from LMC to EU.

2.4.2. Integrity

LMC hashes the head and the content of the label. EU verifies the hashed value to determine whether the label has been tampered during transmission. So the integrity of the label can be ensured effectively.

2.4.3. Non-repudiation

Since both sides have determined the identity of each other before communication, and the LMC's identification, digital signature and other information are carried during the transmission of security label, EU can verify the signature to determine the sender.

3. The File Label Modification Technology Based on Multi-dimensional Domain

3.1. The File Label Structure Based on Multi-dimensional Domain

In today's increasingly complex application environments, the traditional one-dimensional security classification label, such as BLP [6], has been unable to meet the security needs and also increases the difficulty of tracking classified files. So the simple one-dimensional security classification label must be extended to the multi-dimensional label, and the traditional static label must be extended to the one which contains both static and dynamic attributes.

Referring to [7][8], according to the basic attributes and security attributes of the file, the contents of the file label can be divided to the static characteristic label and the dynamic security label. The label structure is shown in Fig. 2.

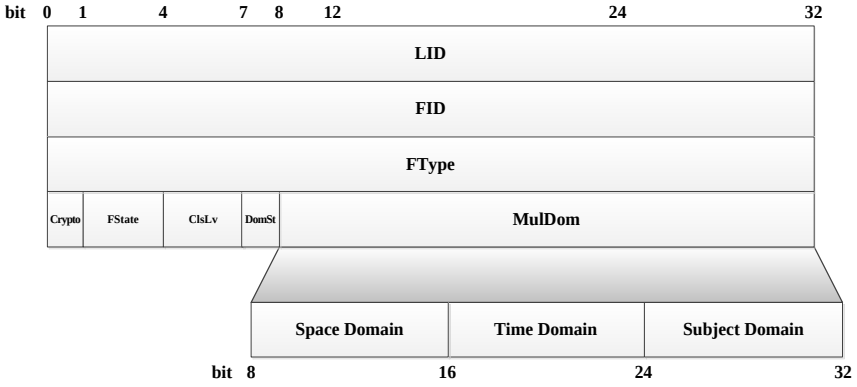

Fig. 2. The file label structure with multi-dimensional domain

The static characteristic label includes the unique label identification, the unique file identification and the file type. The meaning of each part is as follows:

- The unique label identification *LID* (32bit): It is automatically assigned by LMC when the label is generated and each label has a unique *LID*.
- The unique file identification *FID* (32bit): It is automatically generated by the system when the file is created. Each *FID* is corresponding to one file.
- The type of the file *FType* (32bit): Each file has one specific file type.

The dynamic security label includes the encryption label, the file state, the classification level, the domain switch label and the multi-dimensional domain. The meaning of each part is as follows:

- The encryption label *Crypto* (1bit): It indicates that whether the file has been encrypted.

- The file state *FState* (3bit): It indicates the inherent availability state of the file and is specified by the security administrator. There are three states: ALLOW, DENY and AC (i.e. according to Access Control policies).
- The classification Level *ClsLv* (3bit): It indicates the degree of the confidentiality of the file. It can be classified as follows from high to low: Top Secret, Secret, Confidential and Public.
- The domain switch label *DomSwt* (1bit): It indicates whether the multi-dimensional domain is enabled. When the value is 0, it's unable, which means that the environmental factors of files need not to be considered when judging the request. Otherwise, it's enabled, which means that the environmental factors should be considered first according to the rules when judging the request.
- The multi-dimensional domain *MulDom* (24bit): It indicates the current application environment of files, which can be further divided into the space domain, the time domain and the subject domain. Each domain can be dynamically configured by security administrators. The meaning of each domain is as follows:

① Space Domain (8bit): It means that the space in which the file can be accessed. In the narrow sense, the concept of space can be expressed as folders or a disk. In the broad sense, it can be expressed as a department, or an organization.

② Time Domain (8bit): It indicates the time period which the file can be accessed. For example, some files can only be accessed at work by setting this domain.

③ Subject Domain (8bit): It indicates the subjects which the file can be accessed. For example, there is a confidential file and the drafter doesn't want all secret users can access it. The access control can be achieved by setting this domain.

The static characteristic label describes the inherent attributes of the file, which is not allowed to change during the life cycle of the file. The dynamic security label, which can be modified dynamically according to the change of file application environments, describes the file security attributes in detail. With this combination of the static and dynamic label structure, the security attributes and access conditions in the life cycle of the file in a particular environment can be clearly described, which ensures the security and also strong flexibility.

3.2. The Rule of File Label Modification

Because the security label is not constant in its life cycle, it's necessary to modify the contents of the file label according to the application environment on the premise of security principles. When the file label needs to be modified, EU sends a modification request to LMC. When the request has been sent to LMC through the secure transmission channel, the contents which need to be modified will be matched by rules in APC. After a successful match, the request will be sent to LMC to implement label modification operation.

3.2.1. The state variables

Definition 1 Subject set: *S*; File set: *O*; Access mode set $A := \{r, a, w, e\}$.

Definition 2 Security label set: *L*, includes subject label L_S and file label L_O. Classification level label set: L_C.

Domain switch set: *DS*, includes subject domain switch set DS_S and file domain switch set DS_O.

Space domain set: *K*, includes subject space domain set K_S and file space domain set K_O.

Subject domain set: *P*, includes subject subject domain set P_S and file subject domain set P_O.

Time sequence: T:={1,…,t_i,t_i+1,…}. Time period: $TR \subseteq T \times T$, $\forall (t_i, t_j) \in TR \Leftrightarrow t_i < t_j$.

Definition 3 Access Matrix: *M*, M_{ij} indicates that the subject S_i can access the file O_j under the constraint of multi-dimensional domain.

Security functions set: *F*, $\forall f \in F, f = (f_S, f_O, f_C)$, f_S indicates the maximum clearance level function of the subject, f_C indicates the current clearance level function of the subject, and f_O indicates the classification level function of the file.

The current access set: $b := \mathbb{P}(S \times O \times A)$. It indicates what access mode which the subject *S* can access the file *O* in the certain state.

Definition 4 The state set: V:= $\mathbb{P}$ (b×M×F×H×K×T×P).
Definition 5 The request set: R:=S×O×L_C, which indicates the subject S requests to modify the security label L_C of the file O.
Definition 6 $LOD \subseteq L_O \times K_O \times TR \times P_O$: The label L_O can be used by the subject P_O during the time period TR and in the space K_O. $LSD \subseteq L_S \times K_S \times T \times P_S$: The label L_S can be used by the subject P_S at the time T and in the space K_S.

3.2.2. The functions of the domain operation

Definition 7 Get the start time of the time period *start*： $TR \to T$, *start*((t_i, t_j))= t_i. Get the end time of the time period *end*： $TR \to T$, *end*((t_i, t_j))= t_j.
Definition 8 Get the space domain of the subject label *L_getsub_k*： $L_S \to K_S$. Get the time domain of the subject label *L_getsub_t*： $L_S \to T$. Get the subject domain of the subject label *L_getsub_p*： $L_S \to P_S$. Get the space domain of the file label *L_getobj_k*： $L_O \to K_O$. Get the time domain of the file label *L_getobj_t*： $L_O \to TR$. Get the subject domain of the file label *L_getobj_p*： $L_O \to P_O$.
Definition 9 Judge whether the space domain of the subject is included in that of the file *L_in_range_k*: $K_S \times K_O \to \{true, false\}$. Judge whether the time domain of the subject is included in that of the file *L_in_range_tr*: $T \times TR \to \{true, false\}$.Judge whether the subject domain of the subject is included in that of the file *L_in_range_p*: $P_S \times P_O \to \{true, false\}$.
Definition 10 Get the space domain in which the access mode x can be activated *A_getobj_k*： $A \to K_O$. Get the time domain in which the access mode x can be activated *A_getobj_tr*： $A \to TR$. Get the subject domain in which the access mode x can be activated *A_getobj_p*： $A \to P_O$.
Definition 11 Judge whether the space domain of the subject is included in that of the file when the subject requests to access the file with the access mode x *A_in_range_k*: $K_S \times K_O \to \{true, false\}$. Judge whether the time domain of the subject is included in that of the file when the subject requests to access the file with the access mode x *A_in_range_tr*: $T \times TR \to \{true, false\}$. Judge whether the subject domain of the subject is included in that of the file when the subject requests to access the file with the access mode x *A_in_range_p*: $P_S \times P_O \to \{true, false\}$.
Definition 12 Get the domain switch value of the subject label: *getsub_DS*: $L_S \to DS_S$. Get the domain switch value of the file label: *getobj_DS*: $L_O \to DS_O$.

3.2.3. The rule of file label modification

In the state v=(b,M,f,H,k,t,p), when the subject S_i requests to modify the label of the file O_j to L_u, LMC handles this request $rq(S_i, O_j, L_u) \in R$ following below steps:

(i) If $getsub\_DS(f_s(S)) \&\& getobj\_DS(f_o(O)) = 1$:

① If the conditions below satisfy, skip to (iii);

$$\begin{cases} L\_in\_range\_k(L\_getsub\_k(f_s(S_i)), L\_getobj\_k(f_o(O_j))) = true \wedge \\ L\_in\_range\_tr(L\_getsub\_t(f_s(S_i)), L\_getobj\_tr(f_o(O_j))) = true \wedge \\ L\_in\_range\_p(L\_getsub\_p(f_s(S_i)), L\_getobj\_p(f_o(O_j))) = true \wedge \\ A\_in\_range\_k(L\_getsub\_k(f_s(S_i)), A\_get\_k(x)) = true \wedge \\ A\_in\_range\_tr(L\_getsub\_t(f_s(S_i)), A\_get\_tr(x)) = true \wedge \\ A\_in\_range\_p(L\_getsub\_p(f_s(S_i)), A\_get\_p(x)) = true \end{cases}$$

② Otherwise，reject $rq(S_i,O_j,L_u)$.

(ii) If $getsub\_DS(f_s(S)) \&\& getobj\_DS(f_o(O)) = 0$, skip to (iii)；

(iii) Judge according the following steps:

① If the conditions below satisfy, authorize $rq(S_i,O_j,L_u)$:

- $f_c(S_i) \succeq L_u \succeq f_o(O_j)$;
- $$\forall S_k \in S, \begin{cases} (S_k, O_j, a) \in b \Rightarrow L_u \succeq f_c(S_k) \wedge \\ (S_k, O_j, w) \in b \Rightarrow L_u = f_c(S_k) \wedge \\ (S_k, O_j, r) \in b \Rightarrow f_c(S_k) \succeq L_u \end{cases}$$
- $\forall S_k \in S, (S_k, O_j, r) \vee (S_k, O_j, w) \in b \Rightarrow f_c(S_k) \succeq L_u$;
- $\forall O_w \in H(O_j), O_j \in H(O_{P(j)}) \Rightarrow L_u \succeq f_o(O_{P(j)}) \wedge f_o(O_w) \succeq L_u$;
- The subject S_i has the authority to modify the label of the file O_j.

The next state changes in these aspects:

- $f^* = (f \cup \{(O_j, L_u)\}) \setminus \{(O_j, f_o(O_j))\}$;

• The other state variables don't change, and the system runs into the state $v^*=(b,M,f^*,H,k,t,p)$.

② Otherwise， reject $rq(S_i,O_j,L_u)$.

3.3. Proof of the Rule

Theorem: If the state $v=(b,M,f,H,k,t,p)$ is secure according to the axioms of ss-property, *-property and ds-property [5], the state v^* is also secure according to those axioms.

Proof: (1) Prove that v^* is secure according to ss-property.

According to the result of the judgment, we can conclude that $v^*=v$ or $v^*=(b,M,f^*,H,k,t,p)$. When $v^*=v$, v^* is secure according to ss-property obviously, as v is secure according to ss-property. When $v^*=(b,M,f^*,H,k,t,p)$, the following two condition are divided:

(i) $getsub\_DS(f_s(S))\&\&getobj\_DS(f_o(O))=1$

As the rule, in the state v^*, the multi-dimensional domains are not changed, so it doesn't affect the security of the system. As in:

$$f_s^*(S_i)=f_s(S_i),\ f_c^*(S_i)=f_c(S_i),\ f_c(S_i)\succeq L_u \tag{6}$$

We can also conclude from ss-property that:

$$f_s(S_i)\succeq f_c(S_i) \tag{7}$$

So we can conclude from (1) and (2) that:

$$f_s^*(S_i)\succeq f_c^*(S_i)\succeq L_u \tag{8}$$

So $v^*=(b,M,f^*,H,k,t,p)$ is secure according to ss-property.

(ii) $getsub\_DS(f_s(S))\&\&getobj\_DS(f_o(O))=0$

In this condition, the multi-dimensional domains are not judged. So according to the steps in (i), $v^*=(b,M,f^*,H,k,t,p)$ is secure according to ss-property.

Combining (i) and (ii), we can conclude that v^* is secure according to ss-property.

(2) Prove that v^* is secure according to *-property.

When $v^*=v$, v^* is secure according to *-property obviously, as v is secure according to *-property. When $v^*=(b,M,f^*,H,k,t,p)$, the following two condition are divided:

(i) $getsub\_DS(f_s(S))\&\&getobj\_DS(f_o(O))=1$

As the rule, in the state v^*, the multi-dimensional domains are not changed, so it doesn't affect the security of the system. From the rule, if the request is authorized, the subject label and the modified label of the file are also confirm to the *-property. So $v^*=(b,M,f^*,H,k,t,p)$ is secure according to *-property.

(ii) $getsub\_DS(f_s(S))\&\&getobj\_DS(f_o(O))=0$

In this condition, the multi-dimensional domains are not judged. So according to the steps in (i), $v^*=(b,M,f^*,H,k,t,p)$ is secure according to *-property.

Combining (i) and (ii), we can conclude that v^* is secure according to *-property.

(3) Prove that v^* is secure according to ds-property.

When $v^*=v$, v^* is secure according to ds-property obviously, as v is secure according to ds-property. When $v^*=(b,M,f^*,H,k,t,p)$, the following two condition are divided:

(i) $getsub\_DS(f_s(S))\&\&getobj\_DS(f_o(O))=1$

As the rule, in the state v^*, the multi-dimensional domains are not changed, so it doesn't affect the security of the system. From the result of the rule, b and M are not modified. So $v^*=(b,M,f^*,H,k,t,p)$ is secure according to ds-property.

(ii) $getsub\_DS(f_s(S))\&\&getobj\_DS(f_o(O))=0$

In this condition, the multi-dimensional domains are not judged. So according to the steps in (i), $v^*=(b,M,f^*,H,k,t,p)$ is secure according to ds-property.

Combining (i) and (ii), we can conclude that v^* is secure according to ds-property.

So far, the theorem has been proved correct.

4. Conclusion

To resolve the security problems during the process of transmission and modification of security labels, the paper proposes the technologies of file security label transmission based on PKI and file label modification based on multi-dimensional domain. The security of file labels in the operating system can be increasingly enhanced by introducing these two technologies. The files can also be prevented from being leaked which is caused by the security of file labels themselves. The analysis shows that the two technologies can resolve the security problems. But this paper doesn't consider the integrity aspect of files and operations to files by trusted subject. They need further investigation.

5. Acknowledgments

This work has been supported by the National High Technology Research and Development 863 Program of China under Grant No. 2009AA01Z438.

6. References

[1] Chuchang Liu, Angela Billard, Maris Ozols, Nikifor Jeremic. Access Control Models and Security Labelling[C]. In: 30th Australasian Computer Science Conference, Ballarat, Australia, 2007.

[2] R Watson, B Feldman, A Migus, C Vance. Design and Implementation of the TrustedBSD MAC Framework. In Proceedings 3rd DARPA Information Survivability Conference and Exposition, Washington DC, USA, Apr. 2003.

[3] C Vance, R Watson. Security Enhanced BSD. Net-work Associates Laboratories, Jul. 2003, pp. 17-22.

[4] A Nash, W Duane, C Joseph, D Rink. PKI: Implementing and Managing E-Security. The McGraw-Hill Companies, 2001

[5] Diffie W, Hellman ME. New Directions in Cyptography. IEEE Transactions on Information Theory, 1976, IT-22(6), pp. 644-654

[6] D.E Bell, L.J Lapadula. Secure computer systems: a mathematical model. Technical Report, ESD-TR-73-278. MITRE Corporation, Bedford, MA, USA, 1973, pp. 12-29.

[7] L Tan, MT Zhou. Design and Implementation of BLP with Time Character on Linux. Computer Science, 2007, 34(5), pp.92-95.

[8] L Bian, XY Chen, YW Wang. An Improved Multi-dimensional File Label Model. In Proceedings 2nd International Symposium on Computer Network and Multimedia Technology, Wuhan, China, Dec. 2010, pp. 283-286.