An Encryption Mechanism for Secure Sharing about Data Resources in Extended Virtual Machine System

Yunfa Li, Jian Wan, Rong Ouyang, Wei Zhang, Wanqing Li

Grid and Service Computing Lab, School of Computer Science and Technology
Hangzhou Dianzi University, 310018, Hangzhou, China
yunfali@mail.hust.edu.cn

Abstract. With the growth of system scale, it has become very difficult problem that how to ensure the secure sharing of data resources in extended virtual machine system. In order to resolve this problem, we propose an encryption mechanism in extended virtual machine system. In the encryption mechanism, data attributes are defined for each owner and corresponding algorithms are proposed for different processes. In order to justify the feasibility and availability of the encryption mechanism, a series of experiments have been done. The results show that it is feasible to ensure the security of data resources sharing in extended virtual machine system.

Keywords: Encryption Mechanism, Secure Sharing, Data Resources, Extended Virtual Machine System

1. Introduction

With the development of virtual machine technology, more and more hardware and software resources can be integrated into virtual machine. Thus, the virtual machine technology has become one of hottest research technology. In this case, people begin to explore a new virtual machine system structure for various applications. Because the virtualization technology can carve some individual physical machine into multiple virtual containers, people begin to extend the scale of virtual machine system and become some extended virtual machine systems. However, with the rapid growth of system scale, the secure sharing of data resources is becoming more and more difficult. It has become a great challenge that how to ensure the security of data resources sharing in extended virtual machine system. In order to resolve this problem, we propose an encryption mechanism which is based on CPA scheme [1].

2. Encryption Mechanism

In general, different data resources concerns different privacy concerns as private information in extended virtual machine system. To assure each legal member can access to the data resources of system, it is a promising method to encrypt the data resources before sharing. Therefore, the new encryption mechanism should support

these dynamic operations. In order to solve the question, we present a novel encryption mechanism which is based on the CPA scheme. In this section, we mainly describe corresponding algorithm for the novel encryption mechanism

2.1 Corresponding Algorithm

(1) The Setup process

Step 1: Choose a bilinear group G_1 of prime order p with a generator g and a bilinear map e: $G_1 \times G_1 \rightarrow G_T$

Step 2: Assuming that the signature verification key K_v has w bits and $W=\{1,2,...,w\}$, select random numbers y, t_1, …,t_{3n}, t_{3n+1}, $t_{3n+2w} \in Z_p$ and generate the public parameters as : $PK=(e, g, Y, T_1,...,T_{3n}, T_{3n+1},...,T_{3n+2w})$, where $Y=e(g,g)^y$ and $T_i=g^{t_i}$ for $1 \leq i \leq 3n+2w$.

Step 3: Generate the system master key as : $MK=(y, t_1,...t_{3n},...t_{3n+2w})$

Step 4: Initialize version number as $ver=1$ and publish (ver, Pk)

Step 5: (ver, MK) is kept by the authority

(2) The encryption process

Step 1: Assuming $M \in G_T$ and the access structure (AS) is a single AND gate of form $AS= \wedge_{\tilde{i} \in I} \tilde{i}$, chooses a random number $s \in Z_p$ and one-time signature key pair (K_v, K_s) and encrypt M as: (ver, AS, E', E^*, $\{E_i\}_{i \in u}$, $\{K_i\}_{i \in W}$,K_v), where ver is current version number, $E'=M*Y^s$, $E^*=g^s$. For each $i \in I$, $E_i=T_i^s$ if $\tilde{i}$ $=+i$; or $\tilde{i}$ $=-i$. if $i \in U \backslash I$, $E_i=T_{2n+i}^s$. For each $i \in W$, $K_i= T_{3n+i}^s$ if the ith bit of K_v is 0, otherwise, $K_i= T_{3n+w+i}^s$.

Step 2: Signs on tuple (AS, E', E^*, $\{K_i\}_{i \in W}$, K_v) with K_s, and obtain a signature δ.

Step 3: Output the ciphertext (ver, AS, E', E^*, $\{E_i\}_{i \in u}$, $\{K_i\}_{i \in W}$, K_v, δ) of M.

(3) The secret key generation process of user

Step 1: choose a random numbers $r_i \in Z_p$ for each $i \in U \cup W$.

Step 2: Let $r= \sum_{i=1}^{w+n} r_i$ and output $SK=(ver, S, D, \overline{D} =\{D_i, F_i\}_{i \in U}, \hat{D} =\{ \hat{D}_{i,0}, \hat{D}_{i,1} \}_{i \in W})$. Where ver is current version number, $D=g^{y-r}$, $\hat{D}$ is defined as: $\hat{D}_{i,0} = g^{\frac{r_{n+i}}{t_{3n+i}}}$ and $\hat{D}_{i,1} = g^{\frac{r_{n+i}}{t_{3n+i+w}}}$ for each $i \in W$.

(4) The master-public key updating process

Step 1: Define each item $i \in \gamma$, which is within the range of $[1,2n]$, respectively. For each $i \in \gamma$, randomly choose $t'_i \in Z_p$ and compute $rk_i = t'_i / t_i$. For each $i \in \{1,2,...,2n\} \backslash \gamma$, $rk_i=1$.

Step 2: Output proxy re-key as $rk=(ver,\{ rk_i \}_{1 \leq i \leq 2n})$, where ver is current version number

Step 3: Increase the system version number *ver* by 1 when everything is done.

(5) The re-encryption process of ciphertext

Step 1: Define β, which is within the range of *[1,2n],* if *CT* and *rk* contain different version numbers, output *CT*=(*ver*, *AS*, *E'*, *E**, $\{E_i\}_{i\in u}$, $\{K_i\}_{i\in W}$, K_v, δ). Go to Step 4. Otherwise, go to step2

Step 2: For each $i\in\beta$, $E_i^{'}=E_i^{rk_i}$ if $1\le i\le n$, or $E_{i-n}^{'}=(E_{i-n})^{rk_i}$ if $n\le i\le 2n$. For each $i\in U$, $E_i^{'}=E_i$ if $i\notin\beta$ and $i+n\notin\beta$,or $i\notin I$.

Step 3: Output the new re-encrypted ciphertext *CT'*=(*ver*, *AS*, *E'*, *E**, $\{E'_i\}_{i\in u}$, $\{K_i\}_{i\in W}$, K_v, δ).

Step 4: End

(6) The secret key components of user updating process

Step 1: Define each item in θ, which is within the range of *[1,2n],* respectively.

Step 2: If $\overline{D}$ and rk contain different version numbers, return with $\overline{D}$ immediately. Otherwise, go to step 3.

Step 3: For each $i\in\theta$, $D_i^{'}=D_i^{rk_i^{-1}}$ if $1\le i\le n$, or $D_{i-n}^{'}=(D_{i-n})^{rk_i^{-1}}$ if $n\le i\le 2n$. For each $i\in U$, $D_i^{'}=D_i$ if $i\notin\theta$ and $i+n\notin\theta$.

Step 4: Output $\overline{D}'=\{D_i^{'}, F_i\}_{i\in U}$. *ver* in the corresponding users secret key *SK* is increased by 1.

(7) The decryption process

Step 1: The ciphertext receiver verifies the ciphertext *CT* and the signature δ by using public parameters *PK* and the user secret key SK having the same version with *CT*. If the attribute set of *SK* satisfies the ciphertext access structure, the ciphertext receiver will ciphertext the message *M*. go to Step 4 . Otherwise, go to Step 2

Step 2: Suppose (*ver*, *AS*, *E'*, *E**, $\{E_i\}_{i\in u}$, $\{K_i\}_{i\in W}$, K_v, δ), *SK*=(*ver, S, D,* $\overline{D}=\{D_i, F_i\}_{i\in U}$, $\hat{D}=\{\hat{D}_{i,0}, \hat{D}_{i,1}\}_{i\in W}$) and denote *As* by $AS=\wedge_{\tilde{i}\in I}\tilde{i}$. For each $\tilde{i}\in I$, if $\tilde{i}=+i$ and $i\in S$, $e(E_i, D_i)=e(g^{t_i s}, g^{\frac{r_i}{t_i}})=e(g,g)^{r_i s}$. If $\tilde{i}=-i$ and $i\notin S$, $e(E_i, D_i)=e(g^{t_{n+i}s}, g^{\frac{r_i}{t_{n+i}}})=e(g,g)^{r_i s}$. If each $\tilde{i}\in U\backslash I$, $e(E_i, D_i)=e(g^{t_{2n+i}s}, g^{\frac{r_i}{t_{2n+i}}})=e(g,g)^{r_i s}$ If $i\in W$, $e(E_i, D_{i,0})=e(g^{t_{3n+i}s}, g^{\frac{r_{n+i}}{t_{3n+i}}})=e(g,g)^{s(t_{3n+i}*\frac{r_{n+i}}{t_{3n+i}}-r_n)}=e(g,g)^{r_i s}$ and $e(E_i, D_{i,1})=e(g^{t_{3n+i+w}s}, g^{\frac{r_{n+i}}{t_{3n+i+w}}})=e(g,g)^{r_i s}$. go to Step 1.

Step 3: Ciphertext is decrypted as follows: $M=E'/(e(E^*, D)\prod_{i=1}^{n}e(g,g)^{r_i s}$

Step 4: End