A Secure Sharing Mechanism for Data Resources in Extended Virtual Machine System

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

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

Abstract

With the growth of the requirement of users, it becomes a challenge that how to ensure the security of data resources sharing in extended virtual machine system. In order to solve this problem, we propose a secure sharing mechanism which is based on the ciphertext-policy attribute-based encryption scheme (CPA scheme). In the secure sharing mechanism, the corresponding data attributes are defined for each owner and some corresponding algorithms are proposed for different processes. In order to justify the feasibility and availability of the sharing 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: *secure sharing, data resources, virtual machine*

1. Introduction

Virtualization technology is being widely applied in various application domains, which can improve server consolidation and reduce operating costs. With the development of virtual machine technology, more and more data resources can be integrated into virtual machine. Thus, the secure sharing of data resources has become one of hottest research technology. In this case, people begin to explore some new encryption methods to ensure the security of data resources. Under the continuous hard working of people, some advances have been made in recent years. However, a number of challenges remain unaddressed although the advances have been made in extended virtual machine system. In order to resolve this problem, we propose a secure sharing mechanism about data resources which is based on CPA scheme [1].

2. Related Works

In this section, we mainly describe some of the important techniques that are related to our proposed secure sharing mechanism. These techniques can be simply shown as follows.

A trusted group signature architecture is introduced in virtual computing environment [2]. In the group signature architecture, a group signature scheme with the function of message checking is proposed which is based on the discrete logarithm problem. In order to address the correct configuration of all shared resources on multiple machines to achieve this overall security objective, a security architecture for virtual data centers is presented in [3], which is based on virtualization and Trusted Computing technologies. In [4], another secure architecture is also proposed in order to efficiently deal with attacks on virtual machines, which is based on virtual machine monitor. In [5], the authors of paper survey technologies related of using virtual machines to enhance system security. In [6], the authors of paper consider generic architecture for trusted virtual domains and discuss the design choices for

detecting the attacks. In [7], the authors of paper present SPARC, a Security and Privacy Aware Checkpointing mechanism, which enables users to exclude specific applications that contain users' confidential information from being checkpointed by the hypervisor. A method for representing Java security constraints is described in [8], which uses the Alloy modeling language.

Though above these security methods and strategies are very useful for correspondingly application, they will still confront a lot of difficulties. The main reasons are the above security methods and strategies don't consider the virtualization environment of multiple physical machines. In order to overcome these disadvantages, we present a secure sharing mechanism for data resources in extended virtual machine system.

3. Secure Sharing 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 secure sharing mechanism should support these dynamic operations. In order to solve the question, we present a novel secure sharing mechanism which is based on the CPA scheme. In this section, we first describe corresponding algorithm for the novel secure sharing mechanism, which contains seven processes. Then, we analyze the security of the secure sharing mechanism.

3.1 Corresponding Algorithm

In the extended virtual machine system, virtual machine revocation is a challenge issue because each data resource and its corresponding attribute are conceivably shared by multiple virtual machines. Revocation of any single virtual machine would affect others who share the same data resources and attributes.

(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 $W=\{1,2,\ldots,w\}$, select random numbers $y, t_1, \ldots, t_{3n}, t_{3n+1}, t_{3n+2w} \in Z_p$ and generate the public parameters as : $PK=(e, g, Y, T_1,\ldots,T_{3n}, T_{3n+1},\ldots,T_{3n+2w})$, where $Y=e(g,g)^y$ and $T_i=g^{t_i}$ for $1 \le i \le 3n+2w$.

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

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 $i \in \{1,2,\ldots,2n\} \setminus \gamma$, $rk_i=1$.

Step 2: Output proxy re-key as $rk=(ver,\{rk_i\}_{1\le i\le 2n})$, 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 $CT=(ver, AS, E', E^*, \{E_i\}_{i\in u}, \{K_i\}_{i\in W}, K_v, \delta)$. 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 $CT'=(ver, AS, E', E^*, \{E'_i\}_{i\in u}, \{K_i\}_{i\in W}, K_v, \delta)$.

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 $(ver, AS, E', E^*, \{E_i\}_{i \in u}, \{K_i\}_{i \in W}, K_v, \delta)$, $SK=(ver, S, D, \overline{D} = \{D_i, F_i\}_{i \in U}, \hat{D} = \{ \hat{D}_{i,0}, \hat{D}_{i,1} \}_{i \in W})$ 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

3.2 Security Analysis

In our proposed encryption mechanism, the correctness can be verified easily. Therefore, we don't describe the verification process of the encryption mechanism. Here, we only analyze the security of the encryption mechanism. The analysis processes can be described as follows.

Define 1: (*EVM* security) Our encryption mechanism is *EVM* secure if ADV_{EVM} is negligible for any polynomial time adversary.

The *EVM* security of our encryption mechanism can be shown by the following theorem.

Theorem 1. If a *PPT* algorithm (the adversary $\mathcal{A}$) wins our *EVM* security game with non-negligible advantage ADV_{EVM}, we can use this algorithm to construct another *PPT* algorithm $\mathcal{B}$ to solve the *DBDH* problem with advantage $1/2ADV_{EVM}$, assuming that the signature scheme is strongly existentially unforgeable.

Proof: In the DBDH game, the challenger chooses random numbers *a, b, c* form Z_p and flips a fair coin μ. If μ=0, set $z=abc$; If μ=1, set *z* as a random value in Z_p. $\mathcal{B}$ is given $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ and asked to output μ. To answer this challenge, first choose a signature key pair (K_v^*, K_s^*) and then simulates the CPA security game as follows.

Init: $\mathcal{A}$ executes in term of the **Init** Step of the CPA security proof.

Step: In this process, $\mathcal{B}$ generates (Y, T_1,..., T_{3n}) and ($rk^{(2)}$, $rk^{(3)}$, ...$rk^{(ver*)}$) in term of the **Step** process of the *CPA* security proof. $\mathcal{B}$ generates (T_{3n+1},..., T_{3n+2w}) as follows. For $i \in W$, select random numbers ω_i, $\psi_i \in Z_p$, and set $T_{3n+i} = g^{\omega_i}$ and $T_{3n+w+i} = B^{\psi_i}$ if the i^{th} bit of K_v^* is 0, denoted by $K_{v,i}^* = 0$; otherwise, set $T_{3n+w+i} = g^{\omega_i}$ and $T_{3n+i} = B^{\psi_i}$.

Phase 1. $\mathcal{B}$ answers queries for secret key and for decryption.

(1) $\mathcal{B}$ executes in term of the **Phase 1** of the *CPA* security game. When generating ($D_{j,0}$, $D_{j,1}$) for each $j \in W$, $\mathcal{B}$ deals in the same way as non-witness attributes in U except that, R_j^k or R_{n+j}^k are no longer needed when computing $D_{j,0}$ and $D_{j,1}$ since $\hat{D}$ part of a user secret key never needs update.

(2) $\mathcal{A}$ submits a ciphertext CT=(ver, AS, E', E^*, $\{E_i\}_{i \in u}$, $\{K_i\}_{i \in W}$, K_v, δ) for decryption. $\mathcal{B}$ first verifier the signature δ with K_v. If the signature is not valid, $\mathcal{B}$ terminates the *DBDH* simulation game without answering the *DBDH* challenger and starts a new game. Otherwise, proceed. In this case, we know with overwhelming probability. Otherwise, K_v can be used to successfully verify δ and the signature contained in the challenge ciphertext, which is assumed to happen with negligible probability since the signature scheme is strongly existentially unforgeable. In case of $K_v \neq K_v^*$, we can assume the j^{th} bits of them are different. Without loss of generality, we assume that the bit of K_v^* is 0. Therefore, $K_j = T_{3n+w+j}^s = g^{b \cdot s \cdot \psi_j}$. $\mathcal{B}$ then calculates $e(K_j, A) = e(g, g)^{abs\psi_i} = Y^{s\psi_j}$. Since ψ_j is known to $\mathcal{B}$, he gives $E'/(e(K_j, A)^{1/\psi_j})$ to $\mathcal{A}$ as the message M.

Challenge: The adversary submits two equal length messages M_0 and M_1. The challenger flips a random coin b, sets $E^* = M^b * Z$, and outputs the ciphertext CT^* as follows.

$$CT^* = (ver^*,\ AS^*,\ E',\ E^*,\ \{E^{\delta_i \cdot R_i^{(ver^*)}}\}_{i \in I \wedge \tilde{i} = +i},\ \{E^{\xi_i \cdot R_{n+i}^{(ver^*)}}\}_{i \in I \wedge \tilde{i} = -i},\ \{E^{\eta_i}\}_{i \notin I},\ \{E^{\omega_i}\}_{i \in W \wedge K_{v,i}^* = 0},\ \{E^{\psi_i}\}_{i \in W \wedge K_{v,i}^* = 1}\{K_i\}_{i \in W},\ K_v,\ \delta)$$

Phase 2. Repeat **Phase 1**. The only restriction is that, ciphertexts submitted for decryption are not allowed to be derivatives of CT^*. $\mathcal{B}$ is able to verify this by running the re-encryption process algorithm of ciphertext on proxy re-key's and CT^*, and compare the results with the ciphertexts he received from A.

Guess: $\mathcal{A}$ executes in term of the **Guess** Step of the CPA security proof.

Based on the above proof, we can know: Our propose encryption mechanism is secure.

4. Experiments and Results Analysis

In our experiments, our proposed encryption mechanism is used for secure data resource sharing in extended virtual machine system. The whole process can be described as follows.

4.1 A Series of Experiments

In our experiments, we built four virtual machines (each virtual machine denotes a user) and the attribute of each virtual machine is different each other in our extended virtual machine system. The four virtual machines are name VM_0, VM_1, VM_2, and VM_3, respectively. The number of each virtual machine's attribute $Rank_i$ (i=0, 1, 2, 3) and the authority of $Rank_i$ is bigger than that of $Rank_{i+1}$(i=0, 1, 2). The VM_0 is the provider of data resource and the other virtual machine is the receiver of data resource. Moreover, there is an illegal user M who wants to steal the sent data resource. By using the extended virtual machine system, we can get the sending state of the VM_0 and the receiving state that the other virtual machine receives data resource. The state is shown as Table 1 and Table 2, respectively. In addition, we also get the receiving state that the illegal user receives data resource. The receiving state of illegal user M is shown as Table 3.

Table 1. The sending state of the VM_0.

Legeal user	Attribute	Data to be sent
VM_1	$Rank_1=1$	$Rank_1=1$ 66 72 61 74 68 65 20 73 65 72 76 69 63 65 20 69 6E $Rank_2=2$ 69 76 65 20 67 72 69 64 2E 20 54 68 65 6E 2C 20 77 $Rank_3=3$ 65 20 6D 65 63 68 61 6E 69 73 6D 20 6F 66 20 67 72
VM_2,	$Rank_2=2$	
VM_3	$Rank_3=3$	

Table 2. The Receiving State that the other Virtual Machine Receives Data Resource

Legeal user	Attribute	Data Received
VM_1	$Rank_1=1$	66 72 61 74 68 65 20 73 65 72 76 69 63 65 20 69 6E 69 76 65 20 67 72 69 64 2E 20 54 68 65 6E 2C 20 77 65 20 6D 65 63 68 61 6E 69 73 6D 20 6F 66 20 67 72
VM_2,	$Rank_2=2$	69 76 65 20 67 72 69 64 2E 20 54 68 65 6E 2C 20 77 65 20 6D 65 63 68 61 6E 69 73 6D 20 6F 66 20 67 72
VM_3	$Rank_3=3$	65 20 6D 65 63 68 61 6E 69 73 6D 20 6F 66 20 67 72

Table 3. The Receiving State that Illegal User Receives Data Resource

Illegal user	Attribute	Data Received
M	*Unkown*	75 74 72 63 74 75 72 65 20 6F 66 20 20 67 72 69 64 2E 20 54 68 65 6E 2C 61 20 73 65 63 75 72 65 20 6D 65 63 6F 75 70 20 63 6F 6D 6D 75 6E 69 63 63 72 69 62 65 20 74 68 65 20 73 65 61 74 69 6F 6E 20 66 6F 72 20 70 65

4.2 Results Analysis

Comparing the "Data to be sent" column of Table 1 with the "Data Received" column of Table 1, we will find that VM_1 accurately receives all data resources that VM_0.sends, VM_2 accurately receives all data resources of $Rank_2=2$ and $Rank_3=3$,which is sent by VM_0 . VM_3 only receives all data resources of $Rank_3=3$, which is sent by VM_0 . The main reason that these situations generate is: The attribute number of each virtual

machine is different. Comparing the "Data to be sent" row of Table 1 with the "Data Received" row of Table 3, we will find that *M* don't accurately decrypts these data resources that VM_0.sends although he can steal some information. The main reason that these situations generate is using our proposed encryption mechanism in extended virtual machine system.

5. Conclusions

Based on the above analysis and the experiments, we can draw a conclusion that the mechanism is feasible and efficient to guarantee the security of data resources in extended virtual machine system.

Acknowledgments

This paper is supported by National Basic 973 Research Program of China under grant No.2007CB310900, Zhejiang Povincial Natural Science Foundation of China under Grant No. Y1090297 and Y6090312.

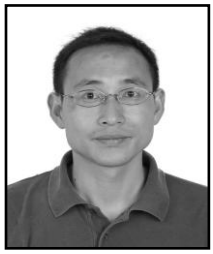
References

[1] L. Cheung and C. Newport (Eds.), "Provably Secure Ciphertext Policy ABE", Proceedings of the 14th ACM Conference on Computer and Communications Security, **(2007)** October 29–November 2, New York, NY, USA.
[2] D. Q. Zou, Y. F. Li, S. Wu and W. Z. Qiang (Eds.), "A Trusted Group Signature Architecture in Virtual Computing Environment", Proceeding of the 5th International Conference on Autonomic and Trusted Computing, **(2008)** June 23-25, Oslo, Norway.
[3] C. Serdar, D. Chris I, E. Konrad, K. Dirk, R. Harigovind V, R. Gianluca, S. Ahmad-Reza, S. Matthias and S. Christian, Journal of Computer Security. 18, 1 **(2010)**.
[4] T. Udaya, V. Vijay and B. Abhishek (Eds.), "Security Architecture for Virtual Machines", Proceedings of the 11th International Conference Algorithms and Architectures for Parallel Processing, **(2011)** October 24-26, Melbourne, VIC, Australia.
[5] S. Q. Zhao, K. Chen, W. M. Zheng (Eds.), "The Application of Virtual Machines on System Security", 4th ChinaGrid Annual Conference, **(2009)** August 21 - 22, Yantai, China.
[6] T. U. Kiran and V. Vijay (Eds.), "Detecting Security Attacks in Trusted Virtual Domains", Proceedings of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, **(2010)** December 11-13, Hong Kong, China.
[7] M. I. Gofman, R. Q Luo, P. Yang and K. Gopalan (Eds.), "SPARC: A Security and Privacy Aware Virtual Machine Checkpointing Mechanism", The 18th ACM Conference on Computer and Communications Security, **(2011)** October 17, Chicago, IL, United states.
[8] M. C Reynolds, "Lightweight modeling of Java virtual machine security constraints", The Second International Conference on Abstract State Machines, Alloy, B and Z, **(2010)** February 22-25, Orford, QC, Canada.

Authors

Yunfa Li, born in 1969. He received PhD degree from Huazhong University of Science and Technology (HUST) in 2008. Currently, he is an associate professor in school of Computer Science and Technology at Hangzhou Dianzi University. His research interests include cloud computing, virtual machine, performance evaluation of software, grid computing, parallel computing and network security. Contact him at yunfali@hust.edu.cn

Jian Wan, born in 1969. He received PhD degree from Zhejiang University of Science and Technology in 1996. Currently, he is a professor in school of Computer Science and Technology at Hangzhou Dianzi University. His research interests include cloud computing, virtual machine, performance evaluation of software, grid computing, and network security.



Rong Ouyang, received PhD degree from Beijing University of Posts and Telecommunications, China in 2007. He is currently a lecturer of the School of Computer Science and Technology, Hangzhou Dianzi Technology, China. His research interests include virtual computing, mobile system, and cloud computing.