UACEE International Journal of Advances in Computer Science and its Applications – IJCSIA
Volume 3 : Issue 3 [ISSN 2250 – 3765]
Publication Date : 09 September 2013

Securing Virtual Machine in Cloud Enviroment using OVF and Hashing Function

Rajinder Sandhu

Dr. Inderveer Chana

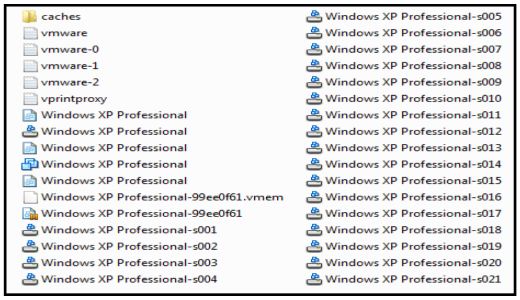
***Abstract*— Over the past decade, with Cloud Computing we are advancing to an industry where IT resources are delivered as a service rather than a product. Cloud promises return on investment, scalability and dynamic provisioning. Virtual machines are backbone for achieving all these promises of Cloud Computing. Despite of best effort from industry Cloud faces many challenges with respect to maintaining integrity and security of virtual machines during their lifecycle. Relationship between virtual machine and security is debatable, which divides research in two groups: Virtual machine for security and Security of virtual machine. Open Virtualization Format developed by Distributed Management Task Force is future for security of virtual machines. In this paper, we propose an algorithm which helps to maintain and check integrity of virtual machine during its migration using Open Virtualization format and Hashing functions. Experiment results demonstrate the algorithm. Algorithm increases trust of user in Cloud environment by increasing integrity in migrated virtual machine.**

Keywords—Cloud Computing, Virtualization, Hashing Function

I. Introduction

Introduction of Cloud Computing in IT industry defines new ways on how we use our infrastructure. This new trend makes commercial products as a service to end user which increases Cloud demand in recent times. From Cloud infrastructure point of view, virtualization is responsible for abstracting physical resource of single computational environment into many separate environments or logical resources [*1*]. Virtualization concept is very valuable in Cloud environment due to its benefits in terms of scalability, dynamic provisioning, cost and portability. Virtualization is adopted by many industries and number is increasing. Survey conducted by Prism Microsystems states 11% industries virtualized their more than 60% resources [*2*].

With increased use of virtualization in industries we often need to migrate virtual machine from one data centre to another.



caches
vmware
vmware-0
vmware-1
vmware-2
vprintproxy
Windows XP Professional
Windows XP Professional
Windows XP Professional
Windows XP Professional
Windows XP Professional
Windows XP Professional-99ee0f61.vmem
Windows XP Professional-99ee0f61
Windows XP Professional-s001
Windows XP Professional-s002
Windows XP Professional-s003
Windows XP Professional-s004
Windows XP Professional-s005
Windows XP Professional-s006
Windows XP Professional-s007
Windows XP Professional-s008
Windows XP Professional-s009
Windows XP Professional-s010
Windows XP Professional-s011
Windows XP Professional-s012
Windows XP Professional-s013
Windows XP Professional-s014
Windows XP Professional-s015
Windows XP Professional-s016
Windows XP Professional-s017
Windows XP Professional-s018
Windows XP Professional-s019
Windows XP Professional-s020
Windows XP Professional-s021

Figure 1. Number of files of a virtual machine

Migration of virtual machine faces many challenges and risks whether it is live or static migration. One reason is number of files created of a single virtual operating system due to splitting of hard disk, as shown in Fig. 1. Maintaining integrity of these files during migration is very difficult. Distributed Management Task Force (DMTF), supported by some big names like Dell, HP, Citrix and Intel, took initiative to zip files of a virtual machine and include all information required for it to run properly in a single file named Open Virtualization format [*3*]. DMTF released specifications of OVF 1.0 and it is now supported by almost all virtual machine monitors.

In this paper, focusing on this problem and presenting how virtual machine integrity can be achieved by using open virtualization format, hashing functions and encryption together.

Rest of this paper is organized as follows. Section 2 summarizes all background related to virtualization, OVF and hashing functions. Section 3 reviews problems and risk while migration of virtual machine. Section 4 introduces a solution to these problems by proposing an algorithm followed by demonstration of this solution in Section 5. Result and discussion of demonstration is shown in Section 6. We finalize by providing conclusion.

II. Motivation and Background

Virtual machines are crucial for accurate working and adoption of cloud computing. Secure portability of virtual machines from one data centre to another is need of hour because it facilities many important properties required by cloud for example interoperability and portability [*4*].

A. *Virtualization*

Virtualization reduces administrative overhead and make system management easier by allowing user to create, read, save, share, modify, migrate and rollback previous state of virtual machines [*5*]. Virtualization is technology used to

Rajinder Sandhu
Thapar University, Patiala
India
rajsandhu1989@gmail.com

Dr. Inderveer Chana
Thapar University, Patiala
inderveer@thapar.

create logical IT resources from physical resources. These resources can be computation power, storage, operating system or network. Each logical view is an independent working environment known as virtual environment or virtual machine (VM). A virtual machine is an identical replica of operating system running on logical resources rather than physical resources. Virtual machines are created by use of hypervisor or virtual machine monitor (VMM). Hypervisor provides two basic characteristics. First, Virtual operating system is identical to original operating system. Second, hypervisor has complete control of physical resources. Hypervisor are of two types: Type 1 (Bare metal) and Type 2 (hosted). Both type of hypervisor are shown in Fig. 2.

B. *Open Virtualization Format*

Open virtualization format is a hypervisor neutral, efficient, extensible and open specification for packaging and distribution of virtual machines in the form of virtual appliances [3].

OVF creates a single directory for multiple files of a virtual machine along with that a XML file called as OVF descriptor is also attached. It contains information about hardware requirements, network description, information about operating system, list of virtual drives, and reference to other files, as shown in Fig. 4.Virtual appliance can be composed of one or more virtual machines. It is developed by DMTF and it is adopted by ANSI and ISO/IEC [6]. OVF 1.0 only supports Packaging, distribution and installation, as shown in Fig. 3, but more support will be provided in coming versions [7].

C. *Hashing Function*

Hashing functions are used in computer science from very long period. Many developers used these hash functions as a black box with magic properties for cryptographic schemes with specific security requirements [8]. MD4, SHA-1 are examples of hashing functions. Different hashing functions are shown in Table 1.

III. Migration Issues

In this section, the risk and problems during migration of a virtual machine has been discussed.

Security of virtual machine is very critical because it contains enterprise/user sensitive data and configuration settings [9].

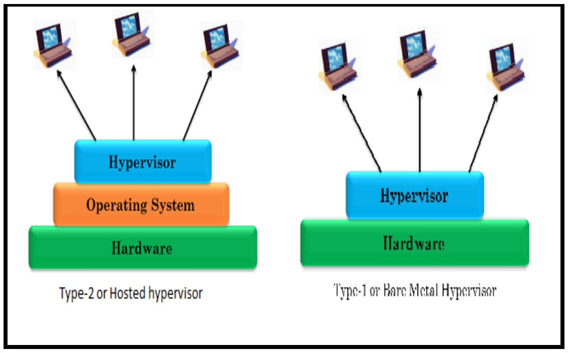


Figure 2. Type of hypervisors

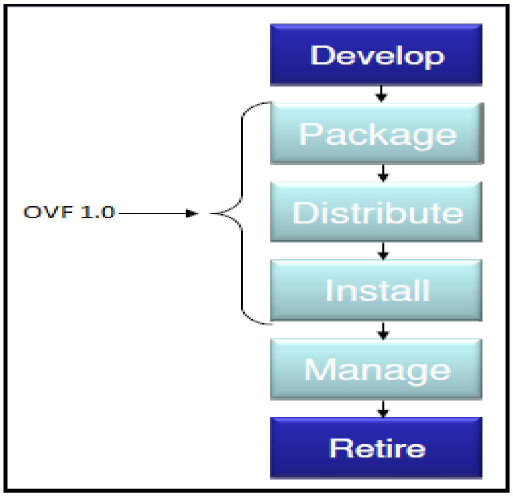


Figure 3. OVF 1.0 Support in Virtual Machine Lifecycle

TABLE I. Different Hash Functions

Name	Block Size (bits)	Word Size (bits)	Output Size (bits)
MD4	512	32	128
MD5	512	32	128
SHA-0	512	32	160
SHA- 1	512	32	160
SHA-224	512	32	224
SHA- 256	512	32	256
SHA- 384	1024	64	384
SHA- 512	1024	64	512

Some of security issues during migration of virtual machine are as follows:

- **Confidentiality:** It is very important in cloud as data is on third party infrastructure and every virtual machine contains device drivers for network, sound, display, private data and configuration files of user which are under attack while migration of virtual machine over a network.
- **Integrity: I**ntegrity of data refers to concept that whether data is changed or not by accident or deliberately. In cloud environment, during migration content of virtual machine can be changed by security attack or any network problem. This changed virtual machine may not run as desired on data centre where it is migrated or it may contain virus which can effect whole data centre.

- **Availability:** A resource not accessible during required time is as bad as none at all. In cloud environment, due to above issues, Confidentiality & Integrity, availability of virtual machine decreases from as desired to much low level. So, users became very reluctant to migrate the virtual machine which increases monopoly and hurt cloud environment in a large version.

In the next section, we propose an algorithm which solves these issues using OVF and hash functions.

IV. Proposed Solution

A. *Algorithm*

Algorithm 1: Packing Virtual Machine

Step-1: Suspend the running virtual machine.

Step-2: Consolidate all hard disk files of virtual machine using ovftool.

Step-3: Store hash function of above files in a text file.

Step-4: Pack the virtual machine using ovftool. Now virtual machine is converted into .ovf file and a volume disk.

Step-5: Calculate hash checksum of ovf file and volume disk and store it in file where another checksum are stored.

Step-6: Encrypt the hash function containing text file using any encryption algorithm.

Step-7: Migrate the virtual machine.

Above stated algorithm shows all steps for packaging virtual machine and attaining hashing functions of different files used for checking integrity of virtual machine. Calculation of hash checksum can be done using any hashing function such as MD5 or SHA-1. Algorithm 2 shows steps for unpacking virtual machine after migration.

Algorithm 2: Unpacking Virtual machine

```
Step-1: De-encrypt the text file containing all hash functions.
Step-2: If (checksum of ovf file and volume disk = checksum in text file)
        Goto step-3
else
        Integrity of ovf file is lost resend it.
Step-3: Unpack the file using ovftool.
Step-4: for (i = file)
if (checksum file[i]=checksum stored)
                goto step 5
        else
integrity of file is lost resend it.
Step-5: Integrity of file is checked and it is completely secure and complete.
```

Above algorithms propose a way in which we can secure and maintain integrity of migrated virtual machines. Complete flow chart of proposed algorithm is shown in Fig. 5. (A) flowchart shows activities for packaging the virtual machine and (B) flowchart depicts activities to unpack the virtual machine after migration.

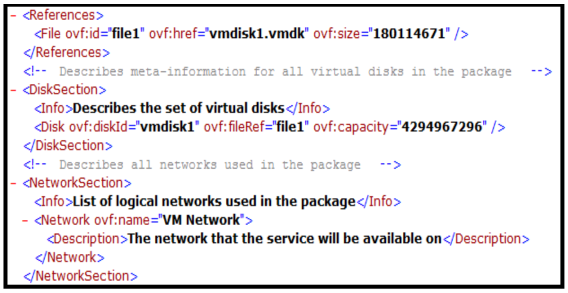
B. *Security Section*

OVF use a standard based Xml descriptor file containing installation and configuration parameter for one or more virtual machine which can be deployed on any hypervisor such as VMware, Microsoft, Citrix or others. Descriptors XML file is capable of managing and troubleshoots VMs.

In addition to algorithm, we propose to include security descriptor in XML file of OVF. This security field will help ovftool to automatically check the checksum by providing information about files and algorithm used for hashing on those files. Sample OVF descriptor format is shown in Fig. 4.

As shown in Fig. 4, there is Network, Disk and Reference field in OVF descriptor file. The proposed OVF security descriptor is shown below:

```
- <SecuritySection>
- <!-- Describes meta-information about files on which hashing function is used and which hash function is used -->
<Info>Describes the set of hash function and files</Info>
<Security ovf:SecurityID="hash1" ovf:SecurityRef="file 1.vmx" ovf:function="MD5" ovf:capacity="2766" />
<Security ovf:SecurityID="hash3" ovf:SecurityRef="file2.vmdk" ovf:function="SHA-1" ovf:capacity="2139947" />
<Encryption ovf:file= “encrypt.txt” ovf:function= “DES” />
</ SecuritySection >
```



```
- <References>
    <File ovf:id="file1" ovf:href="vmdisk1.vmdk" ovf:size="180114671" />
  </References>
  <!--  Describes meta-information for all virtual disks in the package  -->
- <DiskSection>
    <Info>Describes the set of virtual disks</Info>
    <Disk ovf:diskId="vmdisk1" ovf:fileRef="file1" ovf:capacity="4294967296" />
  </DiskSection>
  <!--  Describes all networks used in the package  -->
- <NetworkSection>
    <Info>List of logical networks used in the package</Info>
  - <Network ovf:name="VM Network">
      <Description>The network that the service will be available on</Description>
    </Network>
  </NetworkSection>
```

Figure 4. Sample OVF descriptor file format

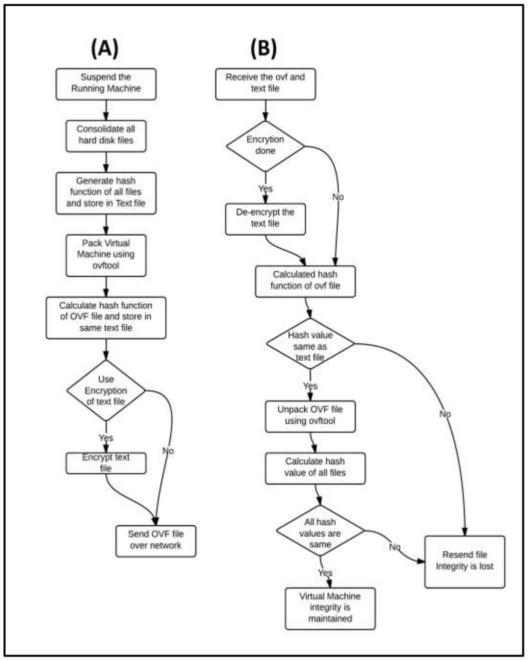

(A)
Suspend the Running Machine
Consolidate all hard disk files
Generate hash function of all files and store in Text file
Pack Virtual Machine using ovftool
Calculate hash function of OVF file and store in same text file
Use Encryption of text file
Yes
No
Encrypt text file
Send OVF file over network
(B)
Receive the ovf and text file
Encrytion done
Yes
No
De-encrypt the text file
Calculated hash function of ovf file
Hash value same as text file
Yes
No
Unpack OVF file using ovftool
Calculate hash value of all files
All hash values are same
No
Yes
Resend file Integrity is lost
Virtual Machine integrity is maintained

Figure 5. Proposed Algorihtm Flowchart

In the security section, two xml fields named Security and Encryption are added. Security has sub fields SecurityID, SecurityRef, function and Capacity. Encryption filed has file and function sub fields. Table II describes about these fields.

It makes virtual machine more secure because we can use multiple hashing techniques on single file also. All hash function values are stored in a text file. Encrypt text file and send over network along with OVF file. Next section shows a complete demonstration of above proposed algorithm.

I. Demonstration

This section demonstrates the proposed algorithm. For demonstration Type-2 (Hosted) hypervisor has been used.

(i) Firstly, created a virtual machine of Ubuntu 12.04 LTS on vmware workstation 8.0. Fig. 6 below shows different files created when virtual machine is created.

(ii) Suspend the virtual machine.

(iii) Consolidate the hard disk drives into one using ovftool. All hard disk files are joined and became a single file as shown in Fig. 7.

(iv) We calculate hash function of 2 files shown in Figure 7:

- ubun.vmx
- ubun-disk1.vmdk

(v) These checksums are stored in a text file whose content are shown in Fig. 8.

(vi) Then convert virtual machine to OVF file using ovftool in command prompt as shown in Fig. 9. Content of ovf file are shown in Fig. 10.

(vii) Calculated the checksum of ovf file and store in text file where previous hash values were stored.

TABLE II SECURITY SECTION FIELDS

Main Field	Sub Field	Description
Security	SecurityID	It provides different identification for files so that they can be recognised easily.
	SecurityRef	It tells name of file upon which hash function is used.
	Function	It tells about the function used for hashing foe example MD5, SHA-1.
	Capacity	It gives size of file.
Encryption	File	It provides name of text file where all hash functions are stored.
	Function	It states algorithm used for encryption of content of text file.

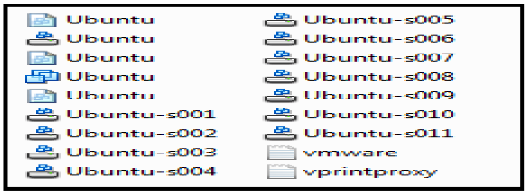
(viii) Send ovf to another server where vmware workstation is already installed.

(ix) At another server, we calculated hash function of ovf file. As shown in Table II, it is same so integrity of ovf file is maintained.

(x) OVF is unpacked using ovftool.

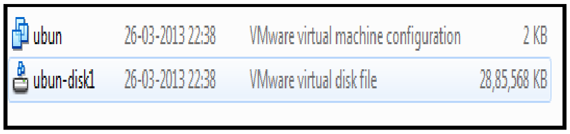
(xi) Hash functions are applied to files after unpacking ovf file, shown in Table III.

(xii) As all hash function values are same integrity of virtual machine is stored.



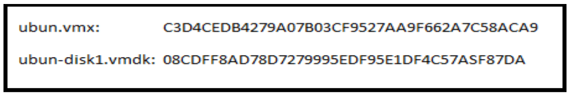
```
Ubuntu          Ubuntu-s005
Ubuntu          Ubuntu-s006
Ubuntu          Ubuntu-s007
Ubuntu          Ubuntu-s008
Ubuntu          Ubuntu-s009
Ubuntu-s001     Ubuntu-s010
Ubuntu-s002     Ubuntu-s011
Ubuntu-s003     vmware
Ubuntu-s004     vprintproxy
```

Figure 6. No. of files in created virtual machine



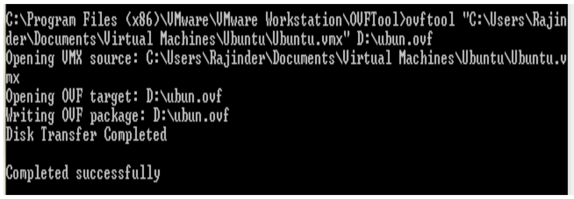
```
ubun         26-03-2013 22:38   VMware virtual machine configuration   2 KB
ubun-disk1   26-03-2013 22:38   VMware virtual disk file               28,85,568 KB
```

Figure 7. Consolidated Virtual Machine File



```
ubun.vmx:        C3D4CEDB4279A07B03CF9527AA9F662A7C58ACA9
ubun-disk1.vmdk: 08CDFF8AD78D7279995EDF95E1DF4C57ASF87DA
```

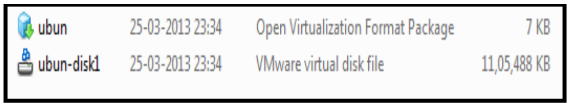
Figure 8. Content of file containing hash values



```
C:\Program Files (x86)\VMware\VMware Workstation\OVFTool>ovftool "C:\Users\Rajin
der\Documents\Virtual Machines\Ubuntu\Ubuntu.vmx" D:\ubun.ovf
Opening VMX source: C:\Users\Rajinder\Documents\Virtual Machines\Ubuntu\Ubuntu.v
mx
Opening OVF target: D:\ubun.ovf
Writing OVF package: D:\ubun.ovf
Disk Transfer Completed

Completed successfully
```

Figure 9. Conversion Completed to OVF



```
ubun          25-03-2013 23:34    Open Virtualization Format Package          7 KB
ubun-disk1    25-03-2013 23:34    VMware virtual disk file            11,05,488 KB
```

Figure 10. Content after conversion

TABLE II. Checksum of OVF file

Original Checksum	New Checksum	Integrity Maintained
d78b419eae3efe2e30234c046777ca4bd18dfac7	d78b419eae3efe2e30234c046777ca4bd18dfac7	Yes

TABLE III. CHECKSUM OF FILES AFTER UNPACKING

Name of File	Original Checksum	New Checksum
Ubun.vmx	C3D4CEDB4279A07B03CF9527AA9F662A7C58ACA9	C3D4CEDB4279A07B03CF9527AA9F662A7C58ACA9
Ubun-disk1.vmdk	08CDFF8AD78D7279995EDF95E1DF4C57ASF87DA	08CDFF8AD78D7279995EDF95E1DF4C57ASF87DA

II. Results and Discussion

As shown in results listed in Table II and Table III checksum of migrated virtual machines can be checked. In this demonstration these checksums are same showing secure and complete migration of virtual machine.

Cloud user trust in easy and secure migration of virtual machine is very necessary for its adoption at higher level. Using above proposed algorithms user can be double sure that virtual machine migrated is not changed before deploying. This also helps to save deploying effort and power because we can check migrated virtual machine in suspended mode.

III. Conclusion

As all IT resources in cloud are in the form of virtual machines whether it is virtual storage, virtual network or virtual server. For the adoption of cloud computing by MSB's and SSB's, secure portability of the virtual machine is need of the hour. In this paper, we propose an algorithm to ensure secure migration of virtual machines over a network. Mechanism includes conversion to OVF, which is universally adopted standard for packaging and distribution of virtual machines, hashing functions and encryption. Demonstrate it with vmware workstation 8 and Ubuntu 12.04 LTS. By using above algorithm we can securely migrate virtual machines. Trust of Cloud user in migration of virtual machines plays crucial role in adoption of Cloud Computing. Secure migration can be road for some of key properties such as interoperability and portability. In future work, encryption of complete ovf descriptor file and automatically recognition of security field by ovftool can be done.

References

[1] Mahesa Jeyakanthan and Amiya Nayak, "Policy management: leveraging the open virtualization format with contract and solution models," *Network, IEEE*, vol. 26, no. 5, pp. 22-27, September-October 2012.

[2] Prism Microsystems, "State of Virtualization Security Survey," Columbia, Survey 877.333.1433, April, 2010.

[3] T. Binz, G. Breiter, F. Leyman, and T. Spatzier, "Portable Cloud Services Using TOSCA," *Internet Computing, IEEE*, vol. 16, no. 3, pp. 80-85, May 2012.

[4] Tal Garfinkel and Mendel Rosenblum, "When Virtual is Harder than Real: Security Challenges in Virtual Machine," in *Proc. Conf. Hot Topics in Operating Systems*, 2005, pp. 20-25.

[5] DMTF, "Open Virtualization Format White Paper," White Paper DSP2017, 2009.

[6] DMTF. (2011, August) DMTF Gains International Recognition with Two ISO/IEC Standards. [Online]. http://dmtf.org/news/pr/2011/8/dmtf-gains-international-recognition-two-isoiec-standards

[7] V Kowalski et al., "Open Virtualization Format Specification," DMTF, White Paper DSP0243 2.0.0b, 2012.

[8] Ilya Mironov, "Hash functions: Theory, attacks, and applications," Microsoft Research, Silicon Valley Campus, Survey 2005.

[9] Chunxiao Li, Anand Raghunathan, and Niraj Kumar Jha, "A Trusted Virtual Machine in an Untrusted Management Environment," *IEEE TRANSACTIONS ON SERVICES COMPUTING*, vol. 5, no. 4, pp. 472-483, October- December 2012.

About Authors:

Rajider Sandhu obtained his B.Tech with Distinction From MMEC, Mullana in 2011. Presently he is pursuing M.E. (Software Engineering) from Thapar University, Patiala. His research areas are Cloud Computing, Virtualization and Software Engineering

Dr. Inderveer Chana is Ph.D in Computer Science with specialization in Grid Computing and M.E. in Software Engineering from Thapar University and B.E. in Computer Science and Engineering. She joined Thapar University in 1997 as Lecturer and has over fourteen years of experience.