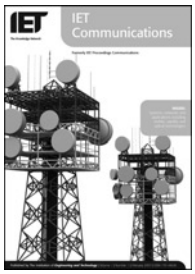
Published in IET Communications
Received on 7th January 2008
Revised on 19th May 2008
doi: 10.1049/iet-com:20080010



ISSN 1751-8628

Trusted virtual machine monitor-based group signature architecture

D. Zou[1] *H. Jin*[1] *J.H. Park*[2] *H.-C. Chao*[3,*] *Y. Li*[1]

[1]*Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, People's Republic of China*
[2]*Department of Computer Science and Engineering, Kyungnam University, Kyungnam, Korea*
[3]*Department of Electronic Engineering and Institute of Computer Science & Information Engineering, National Ilan University, I-Lan, Taiwan*
[*]*Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan*
E-mail: hcc@niu.edu.tw

Abstract: Group communication is an important technique for many network computing applications. In group communication, a member in a group sends a message to others normally by multicast. Group signature guarantees the integrity of the exchanged data and provides source authentication. In a virtual machine (VMs) based computing system, a virtual machine monitor (VMM) allows applications to run in different VMs strongly isolated from each other. A trusted VMM (TVMM) based platform can provide stronger security protection for group signature systems than traditional computing platforms can. The authors first introduce a TVMM-based group signature architecture and a TVMM security protection mechanism for group signature components. Then, the authors propose a group signature scheme using the function of message checking based on the discrete logarithm problem. Finally, the authors prove the correctness of the group signature scheme and analyse its security in virtual computing environments.

1 Introduction

Group communication is a highly efficient message-exchanging mechanism for multiple participants involved in a task, and group security aims to protect the exchanged message in group communication systems. Since in traditional computing environments, an operating system (OS) directly runs on the hardware, it is difficult for the OS to protect itself from being attacked. Applications running on an untrusted operating environment are vulnerable to attacks. A suspicious application can easily attack other applications because of the a lack of an effective isolation mechanism. Consequently, when group communication applications are implemented on top of an untrusted operating platform, sensitive information will be easily disclosed by the attackers even if group management itself can provide strong security.

In a virtual machine (VM) based computing system, multiple applications run in different VMs with their hardware protection domains strongly isolated from each other. Furthermore, migration can be easily implemented based on the virtual machine monitor (VMM) architecture. Normally, the VMM platforms, such as VMW and XEN, mainly provide VM management and physical resource allocation for each VM, and are simpler and stronger than traditional OSs. With its underlying trusted computing hardware, the trusted computers can provide security support for applications. The VMM architecture enables trusted computers to provide more flexible, isolated environments for each application than traditional architectures can do.

The group signature technique guarantees the integrity of exchanged data and provides source authentication. There are many group signature schemes, such as the group signature scheme based on a discrete logarithm, the threshold group signature scheme, the group blind signature scheme and the forward-secure group signature scheme. Although these schemes can meet the security requirements of some applications in some aspects, they fail to consider how to

use a trusted platform to protect the signature components from being attacked. For such applications, with trusted members and platform, using a single signer is more efficient and secure than using more than one signers. In this paper, we propose a group signature scheme with the function of message checking based on the discrete logarithm problem (DLP), and individual group members can sign a message on behalf of the group and only the specified receiver can recover, verify and check the message. Moreover, in case of disputes, the group controller can reveal the identity of the signer, and existing group signature schemes lack such a checking function. Additionally, in our trusted VMM (TVMM) based group signature architecture, the trusted platform can protect the group controller, group members and its sensitive information, such as a private key, residing on it.

The rest of this paper is organised as follows. We discuss related work in Section 2. In Section 3, we introduce a group signature architecture and a group signature scheme with the function of message checking in the trusted virtual computing environment. In Section 4, we prove the correctness and security of our scheme, and analyse its security in TVMM-based architecture. Finally, the conclusions and future work are presented in section 5.

2 Related Work

In the 1970s, VM was a software replica of an underlying real machine, and multiple isolated VMs could operate on the same host machine concurrently [1, 2]. Over the past few years, with the advent of (multicore) technologies, the VM has regained a great deal of attention. In fact, a VM environment is created by a VMM, referred to as 'operating system for operating systems' [3]. The monitor creates one or more VMs on top of a single real machine. Each VM provides facilities for an application or a 'guest system' regarded as an execution in a normal hardware environment.

There are two different methods to build a VM system. One is to implement the VMM between the hardware and the guest systems; examples include as a Xen [4] and VMware ESX Server [5]. The other is to implement the VMM as a normal process on top of a real OS, as adopted by VMware Workstation [6] and User-Mode Linux [5]. Some security research was conducted based on the VMM architecture. For example, Dunlap *et al.* proposed to use VMs to enhance system security. In [7], Revirt, an intermediate layer between the monitor and the host system, captures data sent to the host system through the VM's syslog process (the standard UNIX logging daemon). In the case of the virtual system being compromised, the invader may manipulate the log messages and impair their reliability. In [8], a virtual machine introspection intrusion detection system is described for searching intrusion evidences. In the system, the intrusion detection system executes in a privileged VM and scans data extracted from other VMs. The secure hypervisor (sHype) project [9] aims to support controlled sharing of resources among VMs on a platform, such as memory, CPU cycles and network bandwidth. In [10], a simple yet effective usage control model $UCON_{KI}$ with unique properties of decision continuity and attribute mutability is proposed for OS kernel integrity protection. Furthermore, to enforce $UCON_{KI}$ security policies, a VMM-based architecture is isolated and protected from other untrusted processes inside a VM. In [11], the risk flow policy describes the authorised risks because of covert flows. In this paper, we examine the ability of four policy models to express risk flow policies. The above-mentioned projects have ignored the security of the VMM itself.

The trusted computing group (TCG) defines a set of specifications to provide hardware-based root of trust and a set of primitive functions to propagate trust to application software as well as across platforms [12, 13]. The root of trust in the TCG is a hardware component on the motherboard of a platform called the trusted platform module (TPM). The TPM provides protected data (cryptographic secrets and arbitrary data) by never releasing a root key outside the TPM. In addition, the TPM presents some primitive cryptographic functions, such as random number generation, RSA key generation and RSA asymmetric key algorithms. Most importantly, a TPM provides a mechanism of integrity measurement, storage and reporting of a platform achieving strong protection capabilities and attestations. To utilise the functions provided by TPM, TCG defines TCG Software Stack (TSS) specification [14]. As an integral part of each platform, the TSS in this specification provides functions that can be used by enhanced OSs and applications, and supplies one entry point for applications to the TPM functionality.

In [15], the design and implementation of a TPM facility is presented. In this module, the TPM is virtualised and can support higher level services. Moreover, it can also support suspend and resume operations, as well as migration of a virtual TPM instance with its respective VM across platforms. In [16], a flexible architecture for trusted computing is presented refereed to as Terra. On Terra, applications with a wide range of security requirements run simultaneously on the current OS over commodity hardware implemented by a TVMM. As a high-assurance VM, the TVMM partitions a single tamper-resistant, general purpose platform into multiple isolated VMs.

A group signature is first introduced by Chaum and van Heyst in [17], allowing each group member to sign messages on behalf of a group anonymously and unlinkably. However, in case of later disputes, a designated group manager can open a group signature and then identify the true signer.

In 1998, Lee and Chang [18] presented an efficient group signature scheme based on the DLP. Since two same pieces of information are included in all group signatures generated by the same group member, their scheme is obviously linkable. Therefore it needs to be further improved. To provide unlinkability, an improved group signature scheme is proposed in [19]. Unfortunately, the improved scheme is still

doi: 10.1049/iet-com:20080010

shown to be linkable [20]. Therefore based on Shamir's idea of identity (ID) based cryptosystems [21], Tseng and Jan [22] proposed an ID-based group signature scheme. In this ID-based group signature scheme, anyone (not necessarily a group member) is able to generate a valid group signature on any message, which cannot be opened by the group manager. Therefore this scheme is forgeable. To solve the problem, in [23, 24], Tseng and Jan revised their schemes, and Popescu presented a modification to the Tseng-Jan ID-based scheme [25]. After that, Xian and You [26] proposed a new group signature scheme with strong separability such that the group manager can be split into a membership manager and a revocation manager. In addition, based on the above schemes, Wang [27] presented a security analysis for these group signature schemes. In this paper, we design a TVMM-based group signature scheme, and propose a group signature scheme with an additional function of message checking.

3 TVMM based group signature architecture

In this section, we describe TVMM-based group signature architecture, as depicted in Fig. 1, and introduce a group signature scheme. At the heart of the system architecture, the TVMM can virtualise machine resources by enabling VMs to run independently and concurrently.

3.1 TVMM-based group signature architecture

We first introduce the main components of TVMM-based group signature architecture and secure migration for the group controller, and then propose a group signature scheme.

3.1.1 Components of TVMM-based group signature architecture: There are three levels in the architecture, including hardware platform, TVMM and secured VM for group signature components.

Hardware platform: In the process of attestation, hardware embedded with cryptographic keys is the trust base of the attestation chain. The hardware supports virtualisation, secure I/O and device isolation. The hardware vendor signs their products to prevent leaking of privacy through the hardware private key in the process of attestation. The security chipsets have a set of rich cryptographic operations defined by the TCG and store small amounts of information such as cryptographic keys. Embedded in hardware with a carefully designed interface, the TPM is resistant to software attacks.

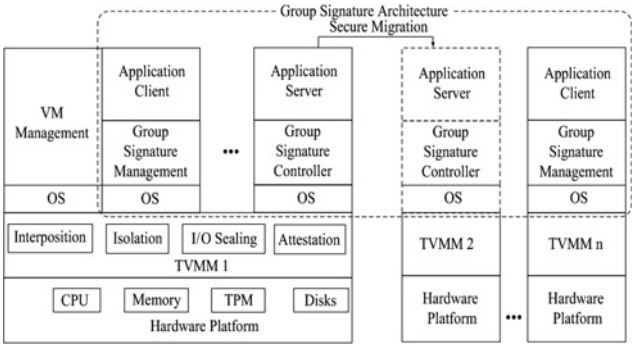


Figure 1 *TVMM-based group signature architecture*

Trusted virtual machine monitor: The VMM provides VM management with interfaces to create and manage VMs. Additionally, the TVMM provides security-enhanced functions such as interposition, I/O sealing, isolation and attestation. We introduce these functions as follows:

i. *Interposition mechanism.* This mechanism takes charge of kernel-user switches, saves/restores the CPU context owned by a trusted process and hides some general purpose registers to the OS kernel to avoid a replay attack.

ii. *I/O sealing mechanism.* This mechanism transparently encrypts and decrypts sensitive I/O data to prevent the OS kernel from observing the data. During being transferred to physical storage, sensitive data of a trusted process is transparently encrypted. The TVMM encrypts each of the I/O system call parameters before passing them to the OS kernel, and intercepts the memory-mapped I/O page table updating requests and decrypts the data on the first page fault.

iii. *Isolation mechanism.* The hardware protection domain can isolate its corresponding VM from other VMs, and a secure isolation is important for the confidentiality and integrity of the VM.

iv. *Attestation mechanism.* This mechanism can convince remote parties that the VM or an application is not tampedred. Specifically, an application running in a VM can authenticate itself to a remote party, and then the remote party puts trust in the application and has faith that the application will behave as desired. By this mechanism, our proposed group signature architecture can provide application-dependent attestation between the group controller and the group members.

Secured VM for group signature components: The open VM can provide the semantics of today's open platforms running OSs. The TVMM offers strong security functions to all VMs. OSs running in VMs may be either as simple as a bootstrap loader plus application code or as complex as a commodity OS with only one application running. On the other hand, applications can completely tailor the OS to their security needs.

There are two kinds of VMs, which are the privileged and normal VMs. For example, Dom 0 is a privileged VM and Dom U is a normal VM in the VM platform, XEN. The privileged VM can manage other normal VMs. For instance, a privileged VM can create, pause, resume and destroy a normal VM. Normally, the privileged VM utilises a tailored