Bridging the Gap Between Inter-communication Boundary and Internal Trusted Components

Yuji Watanabe, Sachiko Yoshihama, Takuya Mishina,
Michiharu Kudo, and Hiroshi Maruyama

IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato-shi,
Kanagawa, 242-8502, Japan
{muew, sachikoy, tmishina, kudo, maruyama}@jp.ibm.com

Abstract. Despite increasing needs for the coalition-based resource sharing, establishing trusted coalition of nodes in an untrusted computing environment is a long-standing yet increasingly important issue to be solved. The Trusted virtual domain (TVD) is a new model for establishing trusted coalitions over heterogeneous and highly decentralized computing environment. The key technology to enable TVD is the integrity assurance mechanism, which allows a remote challenger to verify the configuration and state of a node.

A modern computer system consists of a multi-layer stack of software, such as a hypervisor, a virtual machine, an operating system, middleware, etc. The integrity assurance of software components is established by chains of assurance from the trusted computing base (TCB) at the *lowest* layer, while the communication interface provided by nodes should be properly abstracted at a *higher* layer to support interoperable communication and the fine-grained handling of expressive messages.

To fill the gap between "secure communication between nodes" and "secure communication between trusted components", a notion of "*Secure Message Router (SMR)*", domain-independent, easy to verify, multi-functional communication wrapper for secure communication is introduced in this paper. The SMR provides essential features to establish TVDs : end-to-end secure channel establishment, policy-based message translation and routing, and attestability using fixed clean implementation. A virtual machine-based implementation with a Web service interface is also discussed.

Keywords: Trusted Virtual Domain, Distributed Coalition, Trusted Computing, Mandatory Access Control.

1 Introduction

1.1 Background

In computer and information security history, there have been many efforts to achieve controlled sharing of digital resources. Recent advances in hardware platforms and networking technology increase the need for sharing resources securely

D. Gollmann, J. Meier, and A. Sabelfeld (Eds.): ESORICS 2006, LNCS 4189, pp. 65–80, 2006.

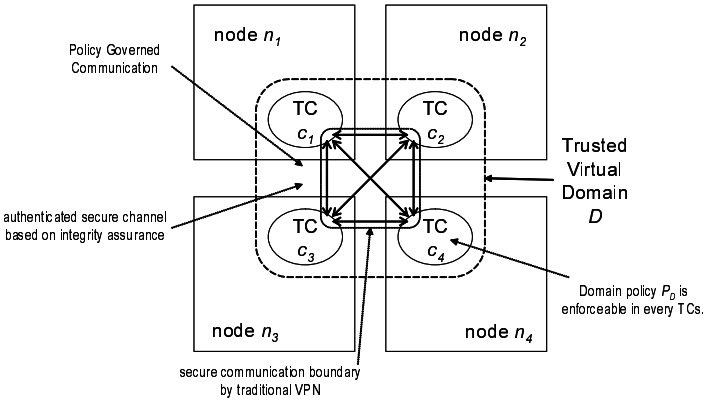

Fig. 1. Trusted Virtual Domain Model

across decentralized administrative nodes in an insecure network world. Moreover, there is a growing tendency for nodes to form coalitions in order to collaborate by sharing some of their resources, or by coordinating some of their activities. Such coalitions are quite common in various scenarios, such as grid computing, on-demand working environments, and virtual enterprises.

With the increasing needs for the coalition-based resource sharing, establishing trusted coalitions in untrusted computing environments is becoming increasingly important concern. The client-server-style authenticated key establishment techniques such as SSL/TLS and IPSec provide end-to-end secure communication channels to build virtual private networks in open but untrusted network environments. Traditional authentication has focused on identifying the identity of the subject, i.e., who is at the other end of the communication channel, in order to control what information can be released to each subject, while not dealing with how the information is managed in the container at the end-point after the release of the information. Thus, once information is released through the channel, the program at the container may, through error or malice, propagate the information improperly. This implies identity-based authentication provides the secure communication channel only between nodes (or the interfaces of nodes) participating in the coalition, while it does not provide any assurances on the information handling (or how information flows in the coalition) in an end-to-end manner.

1.2 Trusted Virtual Domain

To tackle this problem in a universal and flexible way, we are developing an integrity-based end-to-end trusted computing infrastructure called *Trusted Virtual Domain* (TVD), which was originally introduced in an ongoing project in our research division (see [1] for details). A number of recent research projects[2,3,4,5] support realization of the TVD concept in various application domains. The basic

idea of TVD is to establish a coalition of trusted components running on the nodes participating in the coalition in a decentralized, heterogeneous environment in order to allow them to simplify the management and to provide explicit infrastructure-level containment and trust guarantees. Our own view of TVD is formalized by the following definition (illustrated in Figure 1).

Definition 1 (Trusted Virtual Domain). *A coalition C which consists of a set $\{c_1, \ldots, c_l\}$ of (trusted) software components running on the nodes $\{n_1, \ldots, n_l\}$ respectively is a trusted virtual domain D as long as the following properties are satisfied:*

- *(Enforceability) Each component c_i works as a reference monitor for a set R_i of resources in the node n_i.*
- *(Authenticity of Enforceability) Any component c_i in C can provide strong evidence that domain policy P_D is properly enforced for any access to the resources R_i.*
- *(Authenticated secure channel) Any component in C can establish an end-to-end secure communication channel with the other components in C by checking the evidence given above.*
- *(Policy-Governed Communication) Communications between software components through the secure channels established above conforms to the domain policy P_D.*

We refer to a software component which satisfies the above properties as a "trusted component" ("TC" for short).

The major difference between a TVD-based coalition and a VPN-based one is that the software components in TVD are authorized to participate in the domain not only by authenticating their identities but also by attesting to the integrity of the other components (see remarks below). A security assurance of the coalition based on the VPN only covers the channel between nodes, while it does not cover the internal behavior of the nodes because there is no assurance after passing through the channel interfaces to the internal of the node. Meanwhile, the TVD expands its coverage of assurance in such a way that the trusted boundary is not limited to the interfaces of the node, but also includes the trusted software components in the nodes.

Remarks: In general, assurance of the integrity of components does not directly imply any assurance for the security properties of the components. The integrity assurance of components is considered as evidence that assures no unintended behavior will occur, but the intended behavior should be mapped to certain security properties with assurance based on a mathematical or analytical evaluation mechanism. For instance, a rigorous language-based information flow analysis shows the software component as a whole enforces the policies with respect to certain security properties such as confidentiality, integrity, or non-interference [6]. For more complex computer systems, an evaluation under the Common Criteria, which is an internationally recognized ISO standard used by governments and other organizations to assess the security and assurance of components,

provides weaker,but in practice useful, evidence of the assurance of security properties □ .

A modern computer system consists of multi-layer stacks of software, such as a hypervisor, virtual machine, an operating system, and application sandboxes including middleware stack. Without assuming assurance based on tamper-resistant software, no integrity assurance can be provided for any software component which is running on an untrusted software runtime environment, because the behavior of the software running in such an environment can easily be compromised. This characteristic implies that the integrity assurance of software components is established by the chain of assurance from the trusted computing base (TCB) at the *lowest* layer, which could be a hardware layer such as a Trusted Platform Module (TPM) [7][8]. Therefore, the higher the layer the software component is placed in, the harder it is to establish integrity assurance for it.

In contrast, the communication interface provided by nodes should be properly abstracted at the *highest possible* layer to support interoperable communications and fine-grained handling of expressive messages between TCs. For example, highly interoperable Web-service interfaces are implemented on Web application servers, which could be huge, multi-layered, highly complex, and thus often difficult to manage securely. To assure the integrity of the software behind the interface, the TC must include the trusted implementation of its interfaces. However, this could raise serious concerns:

- *Differences of abstraction layers between communication interfaces and TCs*: It is difficult to build an assured implementation which handles both enforceability and communication control, because of the gap between the layer of abstraction for interfaces and that for the trusted components which can be build from the bottom. The communication stack is usually provided by different mechanism from the TC.
- *Multiple TCs in a node*: It is quite common that a communication stack implementing an interface is directly controlled by a node, not by any TCs. In such cases, the interface implementation is never included in a single TC.

Due to these observations, there exists a non-trivial gap between "secure communications between nodes" and "secure communications between trusted components". This gap brings the needs for an independent implementation of the interface to be placed at the boundary and for a secure intra-node communication mechanism to be placed between that implementation and the trusted components.

1.3 Our Contribution

To fill the gap between the TC and the interfaces, this paper introduces the notion of a *Secure Message Router (SMR)*, a domain-independent, easy to verify, multi-functional communication wrapper which mediates the communications between trusted components.

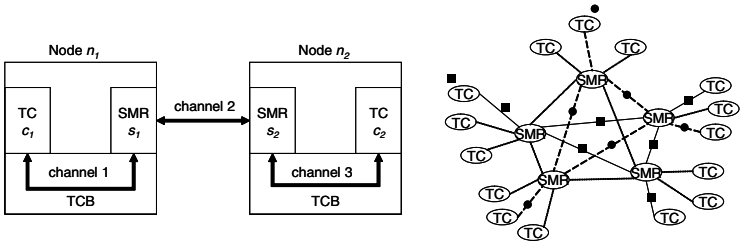

Fig. 2. Secure Messaging Router model

The model of SMR is described in Figure 2. The SMR is placed in each node in a coalition, and mediates the communications between TCs. The SMR works as a component which is governed by the TCB in the same node. The SMR employs the TCB's functions in order to control communication done by the TCs and to detect the state change of the TCs. The TCB controls inbound-outbound communication with TCs so that the communication cannot be done without SMR mediating it. This property provided by the TCB allows the SMR to perform complete mediation of communication done by TCs in the same node. Also, the SMR can manage the list of TCs and their states, and detect change of their states which are continuously observed by the TCB.

We say a communication channel from c_1 to c_2 is *end-to-end secure* if the rule derived from the policy for a domain which c_1 belongs to is enforced all of the messages which are transferred from c_1 to c_2 over the channel. The end-to-end secure communication channel between TCs is mediated by the SMRs s_1 and s_2. The secure channel between c_1 and c_2 consists of the three independent communication channels between c_1 and s_1 (channel 1), betweens_1 and s_2 (channel 2), and between s_2 and c_2 (channel 3). The SMR s_1 works as the end-point of channel 2 between nodes n_1 and n_2, while different secure channels such as channel 1 inside the node n_1 or channel 3 inside the node n_2 between SMR are established. We call channel 2 an *inter-communication channel*, and channel 1 or 3 an *intra-communication channel*. Thus, the inter-communication channel and the two intra-communication channels are collaboratively integrated with the mediation of the SMRs in order to establish an end-to-end secure channel between TCs. The correctness of the behavior of s_1 and s_2 is crucial to connect three secure channels in an end-to-end secure manner. This correctness is attested by integrity assurance which is checked by the TCs c_1 or c_2. This check is domain-independent, in the sense that the correct behavior of the SMR is defined regardless of the design or policy of any specific TVD. On the other hand, the attestation between TCs is domain-specific, which allows the designer of a TVD to support application-specific, finer-grained requirements for TVD.

Besides these features, the SMR provides the following capabilities which are required to establish a TVD.

– *Policy-based message routing*: In principle, only a single SMR is running on each node, while multiple TCs could be contained in the node. The end-

point of an inter-communication channel is an SMR, and thus the single SMR needs to be shared by multiple TCs. A capability for message routing allows nodes to participate in multiple coalitions simultaneously, just by each node maintaining multiple TCs and by the SMRs controlling the destinations of the message transfers according to the domain policy.
- *Secure message translation*: Some messages transmitted over the inter-communication channel could be domain-specific, in other words, dependent on which domain the message is transferred to. For example, to enable mandatory access control (MAC) in the TVD, a security label should be attached to the data which is conveyed in the messages transmitted. If heterogeneous implementations of TCs in coalitions are considered, the security label should be translated to appropriate security labels that can be handled by each TC. The SMR properly translates the messages from and to each TC according to the domain policy.
- *State Management*: Some of the domain policy is stateful in the sense that the destination of a message transfer could be dependent on the state of the coalition, (considering the history of message transfers, status of the TC, etc.) In addition, application-specific policies could be transactional, in the sense that the series of messages needs to be handled sequentially in order to form a unit of execution. The SMR provides stateful and transactional operations for routing and translating messages. The state of the domain is maintained by the SMR or possibly by the TCs in a decentralized manner. The SMR provides a function to integrate the domain state among the SMRs and TCs, which could be used through a unified interface when the SMRs or TCs in the domain evaluating the domain policy.
- *Thin clean implementation*: The SMR provides only a small set of functions for mediation: routing, translation, and state handling. This limited functionality allows us to implement SMR as a domain-independent and reusable trusted software component. Moreover, the feather-weight implementation makes it easy to obtain security assurance in the integrity of the SMR.

This paper is structured as follows: In Section 2, we describe the framework for realizing trusted virtual domain and discuss issues affecting the end-to-end secure communication addressed in this paper. We present our new approach, the Secure Messaging Router (SMR) and describe several protocols based on the SMR in Section 3. Section 4 discusses possible implementations of the SMR. Related works is discussed in Section 5. Finally, we conclude in Section 6 with a summary of our results and consider issues that still remain to be addressed.

2 Flexible Framework for Trusted Virtual Domain

The notion of a TVD can be implemented independently and concurrently using a variety of underlying technologies and mechanisms. One of the promising implementations of TVD is an approach using hypervisor-based isolation coupled with Trusted Computing Group (TCG)-based verification.

TCG technology provides hardware-based assurance of software integrity. This is based on a security module called the Trusted Platform Module (TPM) which is usually implemented as a tamper-resistant hardware module. The TPM measures and reports platform integrity correctly in a manner that cannot be compromised even by the platform owners or the software running on it. The platform measurement mechanism provides strong evidence for enforceability in Definition 1. Also, a challenge-response type confirmation protocol called TCG-attestation allows a remote challenger to verify the precise configuration and state of a computing platform in a reliable way, by using a tamper-resistant secure subsystem. This feature supports the realization of authenticity in Definition 1.

Another important requirement for enforceability and its authenticity is to guarantee the security property that any access to resources must be governed by the domain policy. This can be viewed as stating that all resources controlled by the TC are virtually isolated within exclusively controlled compartments. This fundamental feature is called "containment" in the context of TVD [1]. Hypervisor technologies such as the Xen Hypervisor provides very strong isolation among TCs on the same node, while secure OSes such as SELinux or sandbox mechanisms such as a JVM can also provide isolation among applications running in the same execution environment.

How to realize the authenticated secure channel and policy governed communication in Definition 1 depends on who mediates the communications between TCs and what kinds of information the messages contain. For example, sHype[2] presents how to establish secure communication channels between virtual machines running on Xen hypervisor, where the communications are governed by the MAC policy and the messages transmitted over the channel are associated with MAC labels. In this case as illustrated in Figure 3 (a), a secure communication channel is directly established between VMs only with the mediation of the hypervisor which can be verifiably assumed to be a TCB. This type of communication channel provides a very strong security assurance since it requires no mediation except for a TCB. However, some situations require finer-grained control of the channel and the messages on them. Typical examples include a policy for information flow control or a usage control policy. For example, a member in a coalition may be authorized to send information categorized as an "announcement" at most once a day, as long as the approvals from at least k of the members for that day are attached to the message, but the recipients will be limited to the members who gave approvals. This kind of fine-grained, stateful, and application-specific control is not suitable for the TCB for the following reasons.

- Configuration of the TCB, in general, is controlled only by strictly authorized subjects (such as node administrator).
- Flexible configuration of the TCB degrades its assurance, especially if the assurance of the TCB is attested by using the integrity measurement. (For example, under the Common Criteria evaluation, certification should be issued only for a rigorously fixed configuration of the system.)
- The TCB is placed at the lowest layer, and therefore the granularity of the control can be coarse compared with the application-specific requirements.

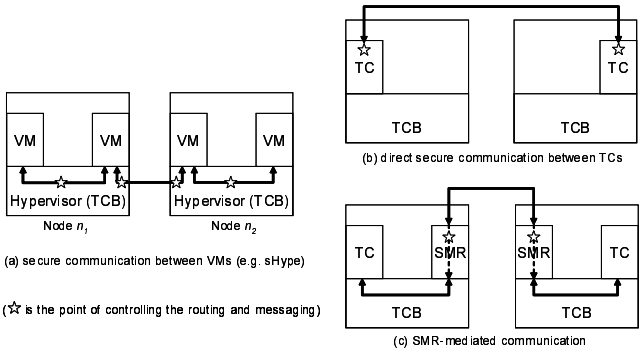

Fig. 3. Models of inter-communication channel

For example, some domain policy might require not only the security label associated with the data, but also more detailed information on who, when, and how the data has been created or manipulated in order to determine if the data transfer is allowed. In general, such state information is finer-grained compared with the information which is observable from the TCB. Furthermore, expressive representations of the data need to be handled in order to support heterogeneous implementations of TCs.

Therefore, an independent mechanism which handles the communications between TCs needs to be provided outside the TCB in order to protect the security of the TCB. There are two types of approach as illustrated in Figure 3 ((b) and (c)). In the first approach, the TC contains the communication stack to establish an end-to-end secure communication channel directly between the TCs. It is easy to establish a secure channel over a public channel by using existing techniques such as SSL/TLS or Web Service (WS) security. Also, arbitrary granularity of the messages can be handled if every TC in a coalition is aware of how to communicate with the others. This is reasonable in a single domain setting, but not in a collaborative multi-domain setting. Specifically, if a node manages multiple TCs in order to participate in multiple independent coalitions and wants to enjoy collaborative activities (such as data transfers, sharing resources, or event notifications) across the multiple coalitions, every TC need to prepare communication stacks on a domain-by-domain basis. This limitation is because the control point of routing and messaging provided by communication stack is tightly bound to a core part of a TC. To overcome the limitations of the first approach and to deal with interoperable communications for multi-domain settings, our proposed approach using the Secure Message Router (SMR) can be used. The SMR can be seen as a divided portion of the TC which only deals with the mediation of secure communications (which is why we call it a "router"). Figure 3 (c) illustrates our approach using SMR, where an end-to-end communications link between TCs is established with the mediation of the SMRs, and this is easily extensible to support multiple TCs simply by routing the messages at the SMR.

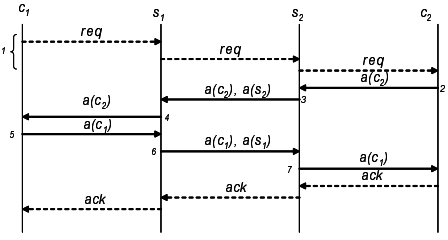

Fig. 4. Establishing end-to-end secure channel between TCs

3 Secure Messaging Router

This section gives an informal description of our proposed approach, *Secure Message Router (SMR)*, which provides complete mediation for the secure communication between TCs to establish a coalition based on the Trusted Virtual Domain (TVD) model.

3.1 End-to-End Secure Channel

Figure 4 depicts the protocol to establish a secure communication channel by using the remote attestation sub-protocol of the integrity measurement, which allows a remote challenger to verify the precise configuration and state of a computing platform. TCG-attestation is one of the promising implementations of this sub-protocol based on a tamper resistant secure subsystem such as TPM. Here, we will discuss how to use the sub-protocol to achieve our goal rather than how the sub-protocol works in detail. Without loss of generality, we focus on a protocol between two nodes hereafter though we believe there is no significant difficulty to extend the mechanism to multi-party scenarios. Suppose the two nodes are n_1 and n_2 and each node n_i, for $i = 1, 2$, manages a TC c_i and a SMR s_i, where c_i has an internal communication channel only with s_i, while s_i also has an external communication channel. We refer to the remote attestation sub-protocol which allows a challenger x to verify y as $RA(x, y)$.

This protocol allows c_1 to establish an end-to-end secure communication channel with the trusted mediation of s_1 and s_2. The basic idea of this protocol is to establish mutual trust based on the integrity assurance of the end points as well as the assurance of the SMRs. Since the message transfers are mediated, c_1 needs to invoke the remote attestation sub-protocol $RA(s_1, s_2)$ and $RA(c_1, c_2)$ simultaneously in order to check the capability of mediation. To establish mutual trust, the counterpart c_2 also needs to confirm the capabilities of c_1 and s_1 by using $RA(c_2, c_1)$ and $RA(s_2, s_1)$. Note that the TC does not need to confirm the integrity assurance of the SMR within the same node since the capability of the SMR for complete mediation is guaranteed by the TCB in the same node. Therefore, in this protocol, s_i and s_2 work as trusted communication wrappers for c_1 and c_2, respectively. On the other hand, the SMR acts as the liaison for the

TC in the same node from the viewpoint of the remote nodes. Thus, the SMR might need to monitor the integrity of the TC in the same node, since the SMR is responsible for the uninterrupted assurance as of the correct configuration and state of the TC in the same node within a session of the secure channel.

3.2 Multi-domain Handling

The basic functions provided by an SMR are the support of *translation* and *routing*, which allows the node to join with multiple coalitions based on the TVD model. Figure 5 illustrates the intuitive scenario for the two nodes n_1 and n_2. Suppose n_1 manages the SMR s_1 and two TCs c_1^A and c_1^B where c_i^x indicates a trusted component which is managed within c_i as a member of a coalition x. For this example, n_1 is participating in two coalitions A and B, and n_2 is participating in A and E.

The specifications of the TC could be node-specific in a heterogeneous computing environment. For example, to enforce a MAC policy such as the Bell-LaPadula policy in a coalition, a security label associated with the data needs to be transferred from node to node. If more fine-grained control is considered, messages which contain more information, such as contexts, states, or obligations, need to be exchanged between nodes in an expressive but interoperable way. The SMR translates the node-specific messages to interoperable messages and vice versa according to translation rules compliant with the domain policies. In the case of Figure 5, when n_1 joins those coalitions, n_1 sets up a translation function t_1 and its inverse t_1^{-1} in the SMR s_1. The function t_1 translates an out-bound message into an interoperable message, while t_1^{-1} translates an in-bound message into a node-specific (application-specific) representation. These translations support not only simple replacement of the term but also finer-grained, constraint-based control complying with the domain policy, e.g., certain constraints prohibit translation, or a part of the message is encrypted under certain conditions. The translation functions t_1 and t_1^{-1} could be changed over time if the node joins or leaves the domain. When those functions are changed, all of the secure channels which has been already established and mediated by the SMR must be re-established because the security of the channel is dependent on the integrity assurance of the SMR.

Routing is another important function of an SMR. For example, in Figure 5, if c_1^A wants to send a message to all of the members in a coalition A, c_1 can do so simply by designating the destination in the message and by sending it to s_1, c_1^A never manage any list of members in the coalition A. Similarly, c_1^A can send the message to c_2^A with the mediation of s_2 simply by designating the destination c_2^A. Note that the correctness of the behavior of the SMR ensures that the message is never transmitted to any other TC such as c_2^E. Intuitively, as illustrated in Figure 5, SMRs in a TVD collaboratively establish a virtual boundary which protects the TCs in the TVD in order for the coalition of the TCs to meet the TVD requirements described in Definition 1. The interactions among multiple domains are fully controlled over their boundaries, and the SMRs work as a gateway across the boundaries. The SMR's abstraction of the communication

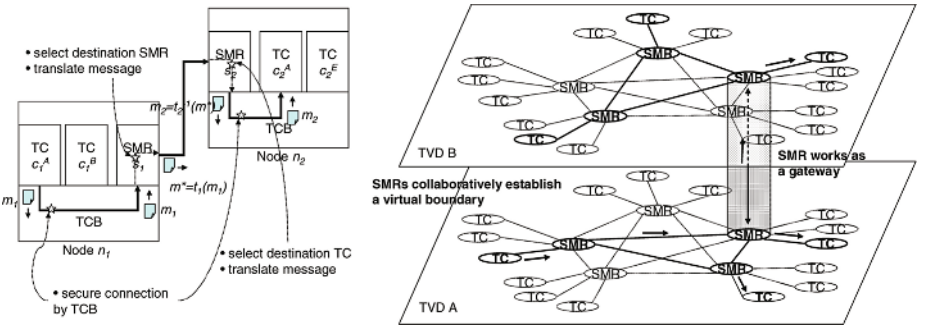

Fig. 5. Multi-domain handling

layer from the domain-specific layer allows the nodes in a coalition to simplify the management of the TVD and the specification of the domain policy.

3.3 TVD Management

When a node wants to join an existing coalition, a node prepares a TC and sets up rules for routing and translation in the SMR, and establishes end-to-end secure communication channels with all of the members in the coalition. This should be done at the time of joining, but some delegation of authority reduces the overhead. For example, all members in a coalition can give some special member (an administrative node) the right to check the membership qualifications of the new member.

The functionality of SMR is optimized for the objective of establishing TVD, in the sense that the SMR provides only the minimal but commonly required functionality for TVD management. In order to facilitate the management of the coalition, the SMR provides a reporting interface which can be accessed from any other node or administration service. This can be used in order to observe the configuration and status inside the node and to offer state management. This interface is also used for the remote attestation sub-protocol as mentioned in Section 3.1. The reporting interface does not allow any input such that it causes a state transition of the SMR, and therefore it does not introduce any side effects affecting the integrity assurance of the SMR.

The SMR is responsible for continuously observing the members in a coalition in order to confirm if the assurance is still valid. To prevent unnoticed alteration of the configuration or the state of the TC, the SMR provides the current status of the TC through the reporting interface. This can be implemented by periodically using the internal attestation protocol (with support by the TCB [7][8]).

4 Implementation of the SMR

Our model of the SMR does not limit the applicability to any specific implementation, though the feature of domain-independent attestability of the SMR

requires a formally-verifiable implementation with minimal but sufficient functionality. A possible implementation could be a thin clean implementation of a Web service handler running over a micro-kernel OS with the smallest set of functionality. In this case, the SMR is introduced within a virtual machine fully controlled by the hypervisor, which is considered to be a TCB. The attestability of the implementation running over the micro-kernel OS has been thoroughly discussed in the literature (e.g. [2]). The reason to exploit the Web service is its interoperability and security[9]. The Web service standard provides interoperability to realize easy connectivity over heterogeneous computing environments, and in addition, the Web service security mechanism allows us to establish secure communications even in the presence of an intermediary mediator, which could be regarded as a significant advantage compared with the other secure channel technologies such as SSL/TLS. Another extreme approach could be a hardware implementation such as an appliance, secure coprocessor, or hardware extension to the node, because it is widely accepted that hardware is more difficult to crack than software. In that case, the assurance of the capability of the SMR must be provided by a different mechanism, such as the Common Criteria evaluation.

A TCB needs to provide two functions; 1) controlling inbound-outbound communication with TCs so that the communication cannot be done without SMR mediating it, 2) monitoring the states of the TCs so that the SMR can detect the change of their states. Furthermore, according to Definition 1, the TC needs to work as a reference monitor for the resources under the control of the TC. A virtual-machine-based implementation meets these requirements. The Virtual Machine Monitor (VMM) works as the TCB, and the Virtual Machines (VMs) running on the VMM are considered as the TCs. In this case, the VM isolation and communication control mechanism provided by the VMM allow the SMR to fully mediate the communication done by the TCs. The VMM can measure the state of the VM by the integrity measurement mechanism such as the TCG technology. The resources in the VM are governed by the policy enforced by the OS which is introduced in the VM.

A VMM supports hardware-level isolation between VMs. This allows the explicit control of communication, while the granularity of the control is bound to the VM-level, because the VMM is not aware of any internal detail of the VM. In the meanwhile, the TCs could be placed at the higher layer. For example, the Integrity Measurement Architecture (IMA) [8] enhances Linux by a TPM-based Linux Security Module in order to generate verifiable representative information about the software stack running on the system. In this case, a process can work as a TC if the integrity of the process is verified by the IMA, which works as a TCB in the sense that the TCB guarantees the TC's correct behavior by allowing the process to access only privileged resources and to communicate with outside only with the mediation of the SMR.

A distributed information flow control is one of the most attractive application of the TVD. Though a TC controls the information flow inside it, the SMR controls the information flow across the TCs. Furthermore, even in the internal information flow, some style of the control requires a domain-wide context which

is never in the TC. The SMR integrates such domain-wide context by accessing the TC internally or by communicating with the other SMRs in the domain. The TC provides the SMR with an implementation of a callback interface which is predefined by the SMR. The TC can also access the SMR through an interface for accessing the domain-wide information. This allows the TC to delegate the domain-wide task such as the membership service or state management.

The SMRs collaboratively manage and share the list of the TCs and their state in the domain. When the node joins or leaves a domain, the state of the SMR in the node is modified because the SMR needs to maintain the state of the domain and the domain policy. Since the end-to-end secure channel between TCs requires that the correctness of the SMR's behavior is properly verified by an integrity assurance mechanism, the state change of the SMR triggers the channel re-establishment. This might be concern when the size of the domain is huge or the network structure is complicated. Another issue is the policy representation for multiple domains. Currently, we are assuming the existence of the mutual agreement on how to represent and evaluate the policy, while this might be too strong when we support highly heterogeneous environment. We need a common understanding on the representation of the policy, as well as a negotiation mechanism to establish mutual agreement on it.

5 Related Work

The Trusted Computing Group [7] has been gaining more attention than before, and various use-cases leveraging TCG technologies have been identified. Sailer et al. [10] utilizes TCG integrity measurements and attestation to protect remote access points, in order to enforce corporate security policies on remote clients in a seamless and scalable manner. One of the most active areas in the TCG is the Trusted Network Connect (TNC), with the main idea of using platform integrity information for network authorization control. One binding of TNC to an existing protocol is to use the TLS extension header [7] to extend EAP-TLS to support TCG attestation within that protocol.

NetTop [11] uses VMWare [12] to isolate execution environments, and allows connecting isolated environments to each other to establish a network of secure environments, and leverages secure OS such as SELinux to enhance the security on the host and the guest OS. Terra [13] realizes isolated trusted platforms on top of a virtual machine monitor, and allows attestation by using a binary image of each virtual machine, such as virtual disks, virtual BIOS, PROM, and VM descriptions. Terra exploits non-TCG based attestation to check the software stacks running in the guest OS, to build trusted relationship between multiple VMs.

Recent efforts on mitigating the drawbacks of TCG attestation include the Semantic Remote Attestation [14], which leverages language-based security and trusted virtual machines to verify the characteristics of a VM in a more semantic manner including attestation of dynamic, arbitrary, and system properties as well as the behavior of the portable code. Property-based Attestation [5,15]

proposes an attestation model with a trusted third party that translates low-level integrity information into a set of properties. WS-Attestation [9] proposes to exchange attestation in the form of a credential which asserts properties and binds those properties to hash-value-based attestation generated by a TPM chip, while protecting the configuration details from potentially malicious challengers.

The notion of an SMR and its goals have a lot in common with the Law-Governed Interaction (LGI), a notion that originally introduced by Minsky [16] with its prototype developed in subsequent work [17,18,19,20]. LGI is a decentralized coordination and control mechanism for distributed systems. The policy management mechanism in the LGI allows a distributed group of heterogeneous nodes to engage in a mode of interaction governed by an explicitly specified policy, called a *law*, which is enforced on every node in order to create a coalition in which members can rely on each other to comply with the given law. Their Moses middleware implements the concept of LGI. Even though our trusted virtual domain based on the SMR provides not only the policy-governed communications among nodes, but also infrastructure-level support for coalition management fully integrated with integrity-based assurance of the configuration and status of the node, nevertheless, the notion of LGI could significantly reduce the complexity of inter-node communication management. In fact, the statements of domain policy in the TVD model are initially specified at an abstract level, and decomposed in order to be enforced on the communications and behaviors of the TCs in each coalition.

Yin and Wang proposed an application-aware IPsec policy system as middleware to provide Internet applications with network-layer security protection [21]. In order to introduce application context into the IPsec policy model, they use a socket monitor which detects the socket activities of applications and reports them to the application policy engine. They also propose an application specification language to configure and distribute application-specific policies. Our approach has some similarity with their approach in terms of bringing the higher-layer's context and requirements such as interoperability to the lower-layer's base security model (the TCB in our case). Their approach employs the network layer's security mechanism as the basic security infrastructure, while our approach employs the isolation mechanism such as hypervisor technology and the integrity-based assurance mechanism such as the TCG technology. Furthermore, the SMR supports a coalition of TCs to establish a trusted virtual domain.

6 Conclusions

Despite the increasing needs for coalition-based resource sharing, establishing trusted coalitions of nodes in untrusted computing environments is a long-standing yet increasingly important problem. A Trusted Virtual Domain (TVD) could be a solution to this problem by establishing trusted coalitions based on integrity assurances in heterogeneous and highly decentralized computing environments. However, the integrity assurance mechanism which is the basis of

the TVD needs to be designed carefully with consideration of modern computer architectures that consist of multi-layer stacks of software, such as hypervisors, virtual machines, operating systems, JVMs, middleware, etc. The gap between "secure communication between nodes" and "secure communication between trusted components" requires a new notion of an SMR, which provides a number of functions such as end-to-end secure channel establishment, policy-based message translation and routing, and attestability through configuration-fixed, clean implementations.

In this paper, we considered virtual machine-based implementations with Web service interfaces as a possible implementation, but the issues related to formal verification of such implementations still remains to be solved. The approach using the language-based information-flow analysis[6] could be the next direction of this research.

The authors wish to thank our colleagues at IBM who helped us with their insights in earlier discussions of our collaborative work on TVD. In particular, we thank Matthias Schunter, Reiner Sailer, Ronald Perez, John L. Griffin, Yasuharu Katsuno, Megumi Nakamura, and Seiji Munetoh. We also benefited from insightful comments by Christian Stueble and Chris I. Dalton. Part of this work has been sponsored by the Ministry of Economy, Trade and Industry, Japan (METI) under contract, New-generation Information Security R&D Program.

References

1. Anthony Bussani, John Linwood Griffin, Bernhard Jansen, Klaus Julisch, Guenter Karjoth, Hiroshi Maruyama, Megumi Nakamura, Ronald Perez, Matthias Schunter, Axel Tanner, Leendert Van Doorn, Els A. Van Herreweghen an Michael Waidner, and Sachiko Yoshihama. Trusted Virtual Domains: Secure Foundations For Business and IT Services. In *IBM Research Report RC23792, IBM Corporation, November 2004 (available from* `http://www.research.ibm.com/ssd_tvd`*).*
2. Reiner Sailer, Trent Jaeger, Enriquillo Valdez, Ramón Cáceres, Ronald Perez, Stefan Berger, John Linwood Griffin, and Leendert van Doorn. Building a MAC-based security architecture for the Xen open-source hypervisor. In *Proc. of 21st Annual Computer Security Applications Conference (ACSAC 2005), Tucson, AZ, USA, pages 276–285, December 2005.*
3. John Linwood Griffin, Trent Jaeger, Ronald Perez, Reiner Sailer, Leendert van Doorn, and Ramon Caceres. Trusted virtual domains: Toward secure distributed services. In *IEEE First Workshop on Hot Topics in System Dependability (HotDep2005), Yokohama, Japan, June 2005.*
4. Hiroshi Maruyama, Frank Seliger, Nataraj Nagaratnam, Tim Ebringer, Seiji Munetoh, Sachiko Yoshihama, and Taiga Nakamura. Trusted platform on demand. In *IBM Research Report RT0564, IBM Corporation, February 2004.*
5. Jonathan Poritz, Matthias Schunter, Els Van Herreweghen, and Michael Waidner. Property attestation - scalable and privacy-friendly security assessment of peer computers. In *IBM Research Report RZ3548, IBM Corporation, May 2004.*
6. Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. In *IEEE Journal on Selected Areas in Communications, Vol 21, No. 1, January 2003.*

7. Trusted Computing Group, http://www.trustedcomputinggroup.org/.
8. Reiner Sailer, Xiaolan Zhang, Trent Jaeger, and Leendert van Doorn. Design and implementation of a TCG-based integrity measurement architecture. In *Proc. of the 11th USENIX Security Symposium. USENIX, San Diego, California, Augest 2004.*
9. Sachiko Yoshihama, Tim Ebringer, Megumi Nakamura, Seiji Munetoh, and Hiroshi Maruyama. WS-attestation: Efficient and fine-grained remote attestation on web services. In *Proc. of International Conference on Web Services (ICWS 2005), Orlando, Florida, USA, July 2005.*
10. Reiner Sailer, Trent Jaeger, Xiaolan Zhang, and Leendert van Doorn. Attestation-based policy enforcement for remote access. In *Proc. of the 11th ACM Conference on Computer and Communications Security (CCS2004), Washington, October, 2004.*
11. HP NetTop: A Technical Overview, http://h71028.www7.hp.com/enterprise/downloads/hp_nettop_whitepaper2.pdf.
12. VMWare, http://www.vmware.com/.
13. Tal Garfinkel, Ben Pfaff, Jim Chow, Mendel Rosenblum, and Dan Boneh. Terra: A virtual machine-based platform for trusted computing. In *Proc. of the 19th Symposium on Operating System Principles(SOSP 2003), October, 2003.*
14. Vivek Haldar, Deepak Chandra, and Michael Franz. Semantic remote attestation - virtual machine directed approach to trusted computing. In *Proc. of the 3rd Virtual Machine Research and Technology Symposium, San Jose, CA, USA, May 2004.*
15. Ahmad-Reza Sadeghi and Christian Stuble. Property-based attestation for computing platforms: Caring about properties, not mechanisms. In *Proc. of the 20th Annual Computer Security Applications Conference (ACSAC2004) December, 2004.*
16. Naftaly H. Minsky. The imposition of protocols over open distributed systems. *IEEE Trans. Softw. Eng.*, 17(2):183–195, 1991.
17. Naftaly H. Minsky and Victoria Ungureanu. Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *ACM Transactions on Software Engineering and Methodology*, 9(3):273–305, 2000.
18. Xuhui Ao and Naftaly H. Minsky. Flexible regulation of distributed coalitions. In *Proc. of the European Symposium on Research in Computer Security (ESORICS2003), Norway, October 2003.*
19. Moses - LGI, http://www.moses.rutgers.edu/.
20. Law Governed Interaction (LGI): A Distributed Coordination and Control Mechanism, http://www.moses.rutgers.edu/documentation/manual.pdf.
21. Heng Yin and Haining Wang. Building an application-aware IPsec policy system. In *Proc. of USENIX Security Symposium '05, Baltimore, MD, August 1-5, 2005.*