# A Behavior based Policy Management for Adaptive Trustworthiness Assignment in Future Network

Akira Wada*, Yasuhiro Sato†, Xuan Liu‡, Tianyi Xing§,
Shingo Ata*, Deep Medhi‡, Dijiang Huang§ and Ikuo Oka*
*Graduate School of Engineering, Osaka City University
†Faculty of Maritime Safety Technology, Japan Coast Guard Academy
‡University of Missouri–Kansas City, U.S.A
§Arizona State University, U.S.A

***Abstract*—A secure network is considered to be an important goal of the Future Internet; one way which can be embodied is by having flexible and robust routing functionality with built-in security and trustworthy mechanisms. However, there is a fundamental and important challenge how to determine the trustworthiness to every traffic flow to realize trustable communications. In this paper, we propose a framework to manage the trustworthiness automatically based on the policy by administrator, hysteresis of the traffic, and/or behavior of end users. We describe the role and function on to manage policy and trustworthiness and illustrate the implementation of SeRViTR, which is a trust routing framework, with communication experiment.**

***Index Terms*—Trustworthiness, SeRViTR, Future Internet, Policy based Management, Routing**

## I. INTRODUCTION

The current Internet has many problems that had never been expected when designed. One such problem is that the current Internet lacks functionalities regarding security. With the background of the growth of the Internet, it was originally designed from the belief that human nature is fundamentally good. There was no consideration against the security issues for example anomaly attacks, and privacy protection, etc. However, as many researchers have noticed, many attacks that cheat vulnerabilities in the Internet have led to critical social problems, so the security is the highest prioritized issue to be solved in the current and future Internet [1]. Moreover, as there is an increase in the number of people who use the Internet, a demand for providing a safe and secure communication infrastructure where people can easily communicate without high-level literacy on security becomes much higher.

From this background, many proposals towards the future Internet are taking security issues into consideration [2]. We also propose a Virtual Trusted Routing and Provisioning Domain (VTRouPD) [3] that is a secure, flexible, scalable, and robust routing framework suitable for future Internet. As the security issue, we especially focus on *trustworthiness*, which presents a degree on how much we can trust a user, node, traffic, and/or network. We believe that *trustworthiness* is a fundamental principle for people to decide who, where, or what to communicate, or judge whether the communications have been really valuable, reliable, and safe. We consider that controlling the traffic based on *trustworthiness* should be a built-in functionality to provide a seamless communication to end users without any special knowledge on security. Our routing framework, called VTRouPD, provides a resilient network in which the traffic is controlled according to *trustworthiness*. Specifically, we first introduce *trust level* as a metric of *trustworthiness*. VTRouPD provides the number of virtualized networks realized by the network virtualization, and traffics with different *trust level* are transferred in different virtual networks that enable to completely isolate the traffic flows each other, based on their *trust level*.

As a proof-of-concept model, Secure and Resilient Virtual Trust Routing (SeRViTR) [4] has been proposed. We define some key functional components that are mandatory to realize the VTRouPD. We also define sequences of signaling and processing in SeRViTR to control the traffic according to *trustworthiness*. A prototype of SeRViTR has already been developed to validate our concept. However, in both VTRouPD and SeRViTR, some important problems still remain. The main questions are "What is *trustworthiness* in actuality?" "How is the *trust level* assigned to corresponding traffic?", "Is it possible to assign *trust level* automatically?" We consider that these functionalities would be a key of successful deployment of a trustworthy routing framework.

Based on the above-mentioned background, in this paper, we propose a framework of the component (called Policy Manager) to manage the *trustworthiness* automatically based on the policy by an administrator, the hysteresis of the traffic, and/or behavior of the end users. The policy manager can control all flows' trust based on the policy defined by the network administrator. At first, we describe the function of the policy manager, which centrally manages the trustworthiness of all traffic based on the policy. However a decision on the trust level for each flow must be performed automatically and must be changed according to the conditions. The difficulty to manage the trust level is that the trust level is a relative criterion to differentiate the traffic flow from others and must be changed by the event that occurred in the network. For this problem, we propose a behavior based dynamic trust level calculation mechanism in the policy manager.

This paper is organized as follows: we show the overview of SeRViTR and its policy manager in Sections II and III respectively. The designed functions in the policy manager

978-3-901882-50-0 

are shown in Section IV, and then we show the experiment on the flow management in Section V. We conclude this paper with future works in Section VI.

## II. Behavior based Policy Management with SeRViTR

### A. Overview of SeRViTR

We proposed a prototype of SeRViTR that realizes a fine-grained routing management with network virtualization based on the flow's trust level [4]. The SeRViTR framework consists of some key components. There are *Policy Manager*, *Domain Controller*, *Flow Controller*, and *Behavior Analyzer*, etc. In this paper, we describe the details of the policy manager that controls the whole of its Managed Domain according to the policy defined by the network administrator. More specific information and the other components about SeRViTR are available in [4].

### B. Behavior-based policy management

We utilize user behavior, which is based on the hysteresis of flow, to quantify the trust of flow. The behavior analyzer classifies the flows' behavior into some patterns according to the number of warnings reported by the traffic monitor, encrypted flow or not, traffic volume, and so on. This pattern is sent to the policy manager and the policy manager decides whether the trust of the flow should be updated based on the policy. If needed, the policy manager recalculates the trust and assigned network resource of the flow. The policy manager then sends the update message to the components to adjust the resource due to the change of the trust level.

## III. Function of Policy Manager

In this section, we describe the overview of the functions of the policy manager, which is a fundamental component of our framework.

### A. Policy setting

The network administrator defines the policy to specify the initial trust level of flow, treatments of flows, and judgement rules of user behavior. These rules are statically set by the administrator in the policy. The policy manager applies them to the behavior analyzer to classify user behaviors. Exchange information with the other managed domains is also defined in the policy.

Note here that to support various types of entities to set the trust level, we need to accept an arbitrary degree of aggregation of flows. Basically, we classify flows by regarding packets having the same values in the specified fields (called *ruleset*) of the packet header, as the same flow. The granularity of flows is varied by changing the number of the header to be specified. Though we only discuss the trust of *flows* in this paper, we can extend out discussion to a trust of hosts/networks/users by changing the aggregation level of flows. Therefore, throughout this paper, we refer to *flows* not only as the traffic flows themselves but also other entities like nodes, networks, and/or users.

### B. Behavior-based trust calculation

The assigned trust levels are maintained only in the policy manager protected to be read by other components because the trustworthiness of flow is confidential information for the routing that must be hidden from other domains. It is very important to prevent the case where a malicious user illegally specifies the tampered trust level to let an unfair forwarding of flows in the trusted network. By protecting the actual trust level to other components, the trust level is just an ID to distinguish a different trust level as observed from any components expect the policy manager.

Moreover, the policy manager usually updates the trust level of flow based on the recorded behaviors of the flow. When an anomaly warning is reported from the behavior analyzer, the policy manager degrades the trust level of the flow alerted. On the other hand, if anomaly warnings relevant to a flow are not reported within the specified observation window, such as time interval or transferred bytes, the policy manager then increases the trust level for the flow.

### C. Comprehensive control to forward flows

The comprehensive control to the other components is an important role of the policy manager. After determining the trust level for the unknown flow, the policy manager next checks whether the trust level is already associated with the virtual domain to forward. If there is no virtual domain associated with the trust level to forward a flow, the policy manager sends a request to create a new virtual domain for the newly created trust level. The request is sent to the domain controller and the policy manager receives Virtual Domain ID (VDI) and Virtual Link ID (VLI) as the reply. Otherwise, the policy manager sends an update request including the flow controller's address to add the associated virtual domain. For deletion, the policy manager first sends a deletion request message to remove the flow controller that has no flow to forward the virtual domain. After deletion of the flow controller, the policy manager next sends a request of virtual domain deletion if there is no flow controller in the virtual domain.

### D. Collaboration with other managed domains

In case of inter-domain communication where the flow is traversed across the multiple managed domains, one problem raised is how to decide the trust level for the flow that is coming from/going to another managed domain. Generally, the administrator of the managed domain can only control the trust level within his/her domain. It is impossible to give explicit control to the flow in other managed domains, even if the traffic is coming from the administrator's domain. A regulation scheme of the trust level between multiple managed domains to treat the same flow is necessary.

The trust level for traffic coming from another managed domain is again re-calculated at the edge of the managed domain. Indeed, the policy manager can assign the trust level of inbound traffic by itself, based on the results of behavior analysis, as that taken for in-domain flows. However, the

information of trust level assigned by the policy manager of the foreign managed domain would be useful and reliable to decide the trust level within the managed domain. Therefore, for inter-domain communications, the policy managers of both managed domains can exchange not only the trust levels used in their own managed domain (called Outbound Domain ID (ODI)) but also the priority order of these levels.

## IV. Design of Behavior-based Policy Management

We design behavior-based policy management according to the functions described above.

### A. Design concept and basic operation

As the basic principle, we consider designing the policy manager from the database-oriented approach. The reasons why we apply the database-oriented principle are (1) to achieve the consistency of the information easily against huge updates by various components, and (2) to operate the information update and the control of components independently and asynchronously.

Each function of the policy manager should treat various information that was generated in it or received from other components. However, the operations of these components and functions are basically independent and not synchronized. There may be a conflict that a certain component is updating information while another component is retrieving the information. To achieve the consistency of the information that is stored in the policy manager, we consider maintaining the information for the policy manager by a relational database.

### B. Description of functional blocks

The diagram of functional blocks in the policy manager is shown in Fig. 1. This figure clarifies how the information is transferred between function blocks.

*1) Host and flow management:* This group consists of the blocks to process when a new/unknown flow is coming. The Flow Information collects the information about hosts and flows, such as HostID, FlowID, Flow Controller Address, Flow GroupID (FGI), and Priority for ODI from the Flow Information Manager. Here, FGI is used to manage a set of flows such as the forward and the reverse flows between source and destination hosts.

Regarding the initial trust calculation, the policy manager calculates the initial trust level for the incoming FlowID to control forwarding the process in the managed domain. This calculation also assigns the initial trust level for the host joining the managed domain. The trust level of the host is calculated based on the summary of trust levels of flows in which the address of the host is specified as the source or destination of flows. Additionally, the trust level of other managed domain is calculated based on the priority of flows to make the decision easily.

After preparing virtual domains, the Flow Table Update function creates a new flow entry relevant to the flow and sends an update message to the flow controllers that are edges of its managed domain, which the flow passes through. This message includes FlowID and VLI to control how to forward the flow.

*2) Behavior management:* The main function of behavior management is to determine change in the trust level for every flow registered in the flow database. We call the function as *Behavior Judgement*, while the determination is *Judgement*. The judgement is the metric of whether the trust level of the registered flow should be changed. If the judgement means it would be updated, the Trust Level Update acquires the trust level and the judgement from the database and calculates the updated trust level. After that, this block updates trust in Trust table by the calculation result. If it changes the trust level for ODI, the policy manager may notify this change to other managed domains through the trust level regulator. Based on the policy, the policy manager determines whether it sends a Trust Level Change Notification message.

The Outbound Domain Notifications receive all of the notification messages from other managed domains, such as the trust level change notification messages. By receiving the ODI in the notification, the policy manager recognizes that an outside domain has judged the associated flow is untrusted flow. Based on the notification, the policy manager assigns a negative score to the flow and recalculates the trust level of the flow.

*3) Resource management:* The main role of the resource management is for traffic engineering of the managed domain. Specifically, in the managed domain, there are a number of virtual domains that are isolated from each other to transfer flows independently based on the trust level. The explicit resource of the network should also be assigned according to the trust level. The main function of the resource management is to calculate sufficient network resources for each virtual domain and send a resource request message to the resource mapper. Though the actual resource management is taken by the domain controller, the policy manager needs the information regarding the current assignments to the virtual domains and available resources in the network.

## V. Implementation of policy manager

### A. Implementation of policy manager's function

We implement function blocks of the policy manager as C++ programs running on a generic Linux based PC. Our relational database is operated on MySQL 5.1. As the initial implementation, we consider the scenario where the policy is statically defined by the administrator, and two hosts are newly connected to the managed domain. We use OpenFlow switches for the flow controller and the virtual router. Therefore, the ruleset is created based on OpenFlow semantics. Moreover, we adopt VLAN ID as VLI and define VDI is the same as VLI, for simplicity. We also assign the successive number to the trust level.

### B. Experimental environment

The network environment of our implementation is shown in Fig. 2. We established layer-2 GRE connections among three sites: Osaka City University (OCU), Arizona State University

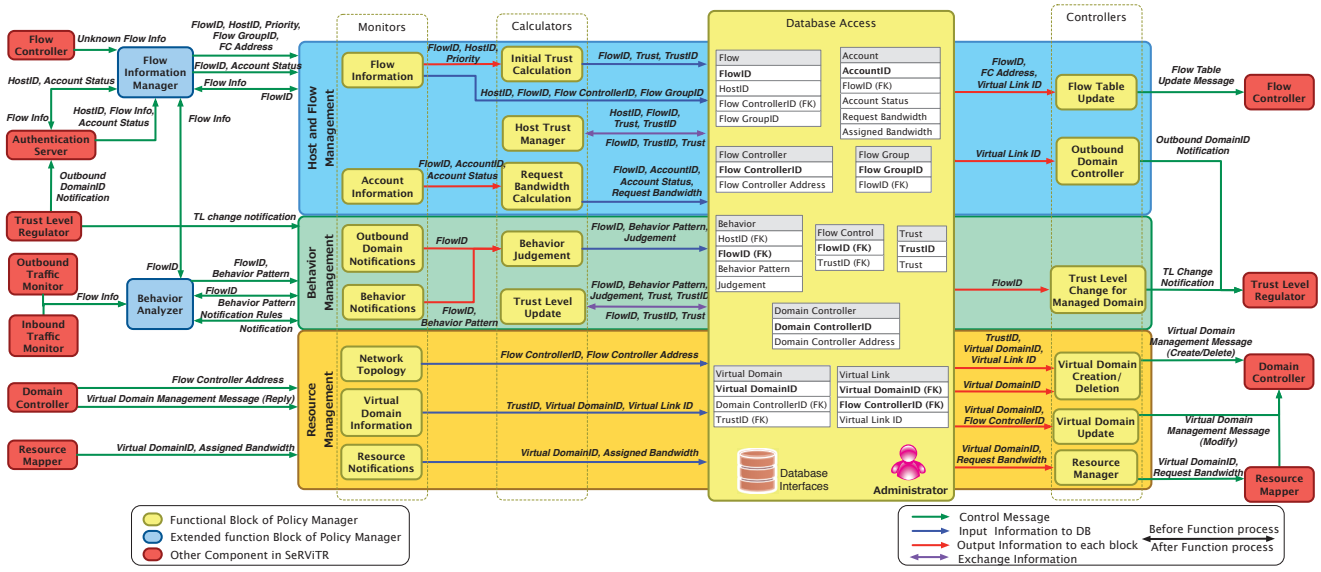


Fig. 1. Function block diagram of policy manager

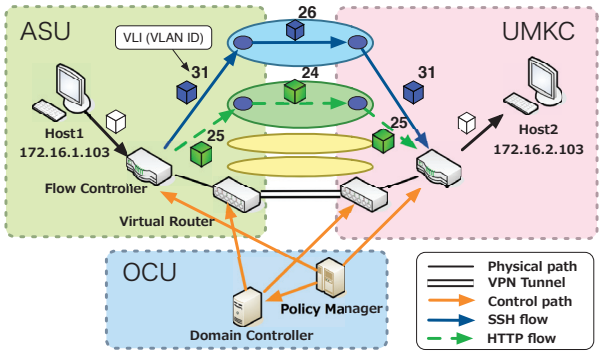


Fig. 2. Single managed domain environment

(ASU), and the University of Missouri-Kansas City (UMKC) by using Open vSwitch [5]. In this experiment, we used three sites as a single managed domain, and the policy manager was located in OCU. The policy manager assigned the value of VLAN (VLI) for each packet, which were randomly selected and ranged from 1 to 100.

### C. Validation of flow management

We considered two different applications: HTTP and SSH. We assumed that the SSH flow is more trustable than the HTTP flow. First, we showed the VLI of each flow in Fig. 2. We then validated that the flow was forwarded to a correct virtual domain according to the control of the policy manager. We checked whether each flow was forwarded to the appropriate virtual domain, by tracking two flows that were sent from Host1 to Host2. Although we did not show the packet traces due to space limitation, we could confirm that each flow was forwarded to the destination host via the appropriate virtual domain.

## VI. Conclusion and Future Work

In this paper, we described the functions of the policy manager. To show the flow control based on flow's trust, we implemented the functions of the policy manager, and performed some experiments in the network where three distinct sites were connected. As a result, the policy manager controlled the flow which was inputted to the managed domain according to its trust. In the future, we implement the remaining function of the policy manager.

## Acknowledgments

This work is supported by US NSF grants CNS-1029562 and CNS-1029546, the Office of Naval Research's (ONR) Young Investigator Program (YIP), an HP IRP grant, and a Japanese NICT International Collaborative Research Grant.

## References

[1] U. Kuter and J. Golbeck, "Using probabilistic confidence models for trust inference in web-based social networks," *ACM Transactions on Internet Technology*, vol. 10, no. 2, pp. 1–23, May 2010.

[2] T. Li-qin, L. Chuang, and Sunjinxia, "A kind of prediction method of user behaviour for future trustworthy network," in *Proceedings of the 10th IEEE International Conference on Communication Technology (ICCT 2006)*, Guilin, China, November 2006, pp. 1–4.

[3] D. Huang, S. Ata, and D. Medhi, "Establishing secure virtual trust routing and provisioning domains for future Internet," in *Proceedings of IEEE GLOBECOM 2010*, Miami, FL, USA, Decenber 2010, pp. 1–6.

[4] X. Liu, A. Wada, T. Xing, P. Juluri, Y. Sato, S. Ata, D. Huang, and D. Medhi, "SeRViTR: A framework for trust and policy management for a secure Internet and its proof-of-concept implementation," in *Proceedings of the 4th IEEE/IFIP International Workshop on Management of the Future Internet (ManFI)*, Maui, HI, USA, April 2012, pp. 1159–1166.

[5] T. Xing, X. Liu, C.-J. Chung, A. Wada, S. Ata, D. Huang, and D. Medhi, "Constructing a virtual networking environment in a geo-distributed programmable layer-2 networking environment (G-PLaNE)," in *Proceedings of the IEEE 5th International Workshop on the Network of the Future (FutureNet-V)*, Ottawa, Canada, June 2012, pp. 1–6.