The TRX Traceability Platform

Toshihiro Motoda, Takeshi Nagayoshi, Junya Akiba, and Kaku Takeuchi

Abstract

TRX, which was developed by the NTT Secure Platform Laboratories, is a traceability platform for visualizing various events that occur in a system. A visualization function in the system makes it possible to track operations such as *copy* and *move* that are made to files and virtual machine images. This function provides easy and unprecedented understanding of the flow of information within an enterprise. It can be used for file life-cycle management in offices and license management of virtual server operating systems for cloud providers.

Keywords: visualization, log, traceability

1. Introduction

The concept of traceability as applied to farm products such as beef and other foods in the Japanese market is well known. It makes it possible to know when, where, and by whom cattle or another product was raised, and how the products were distributed. This gives consumers more information on the products available to them and allows them greater choice in what products to buy.

But can this concept be applied to information systems? When you create digital materials and send them to another person, how can you find out what subsequently happens to them, for example, how, where, and by whom they are later used and modified? Unease concerning information systems may arise when we are not sure how information is handled within the system because we have no way to visualize the situation [1]. Information traceability makes it possible to track and provide a visual view of how documents and other information are moved around and altered within an information system.

2. TRX traceability platform

The NTT Secure Platform Laboratories has been developing the TRX traceability platform as a step toward achieving information traceability. The role of TRX is illustrated in **Fig. 1**. The TRX traceability platform makes events visible when they occur in a system. These events include file operations, the creation of virtual servers in the cloud or another system, or web accesses that occur during the provision of services. This visualization is a matter of making connections in the relationships between events, and displaying in a visual or easily processed format information that cannot be understood simply by looking at individual events.

The functional elements of the TRX system and the flow of log information are illustrated in **Fig. 2**. Here, various events can be visualized. For example, a function for precisely visualizing file operations performed manually and a function for detecting the copying of virtual machine (VM) images[*1] using active trace technology are provided as basic functions. These functions are features of the TRX traceability platform and are described in more detail in the following subsections.

2.1 File operation visualization (file tracing)

(1) Highly precise logging of file operation events

The flow of log generation for file operation events in TRX is shown in **Fig. 3**. The file trace log-generating function on the user terminal monitors the file I/O

[*1] VM image: An electronic file that stores individual virtual server instances when virtualization technology is used to construct servers etc. Because it is an electronic file, it is easy to create, delete, or copy a virtual server.

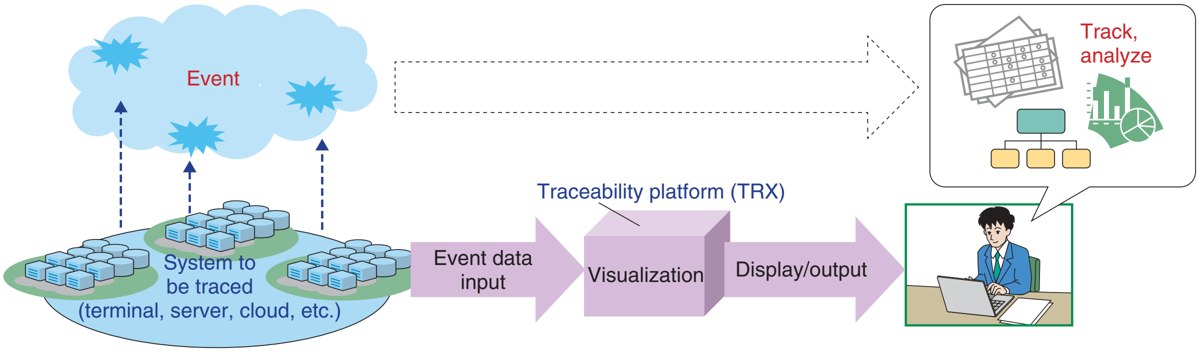
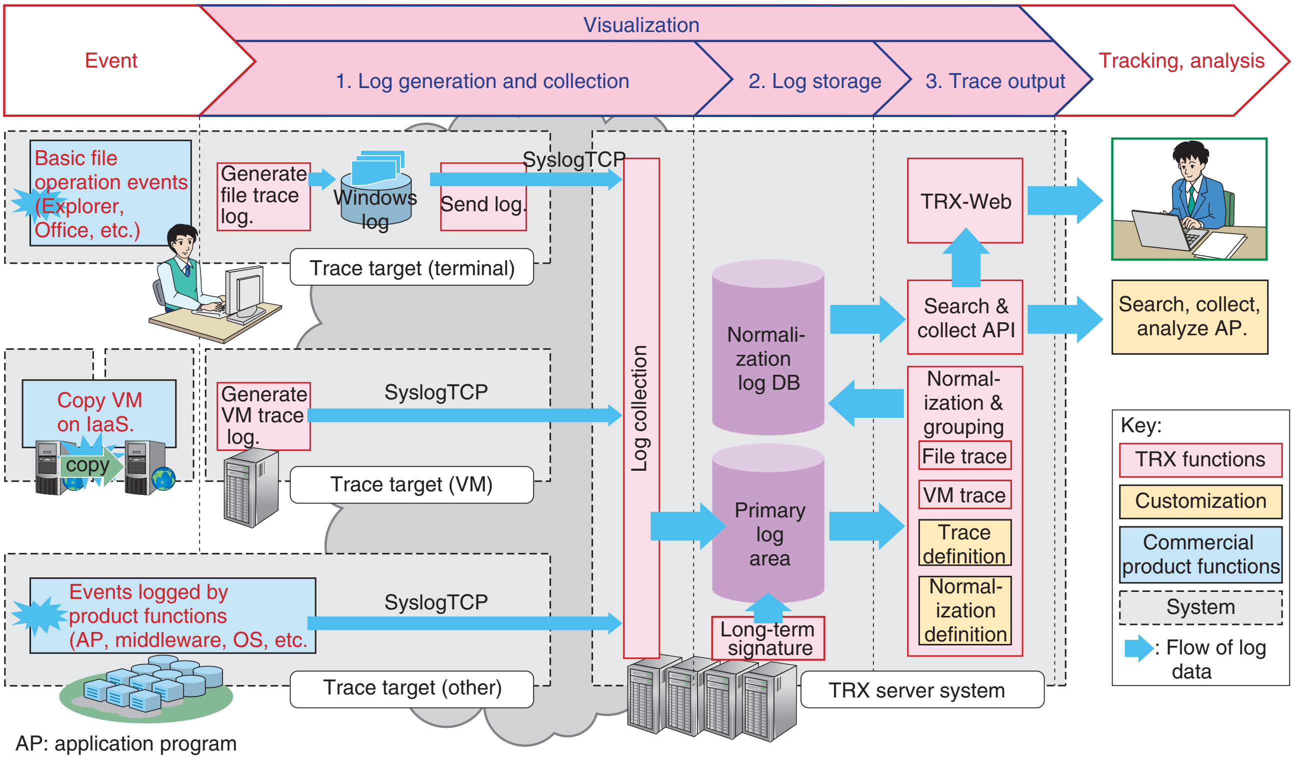

Event
Track, analyze
Traceability platform (TRX)
System to be traced (terminal, server, cloud, etc.)
Event data input
Visualization
Display/output

Fig. 1. TRX traceability platform.

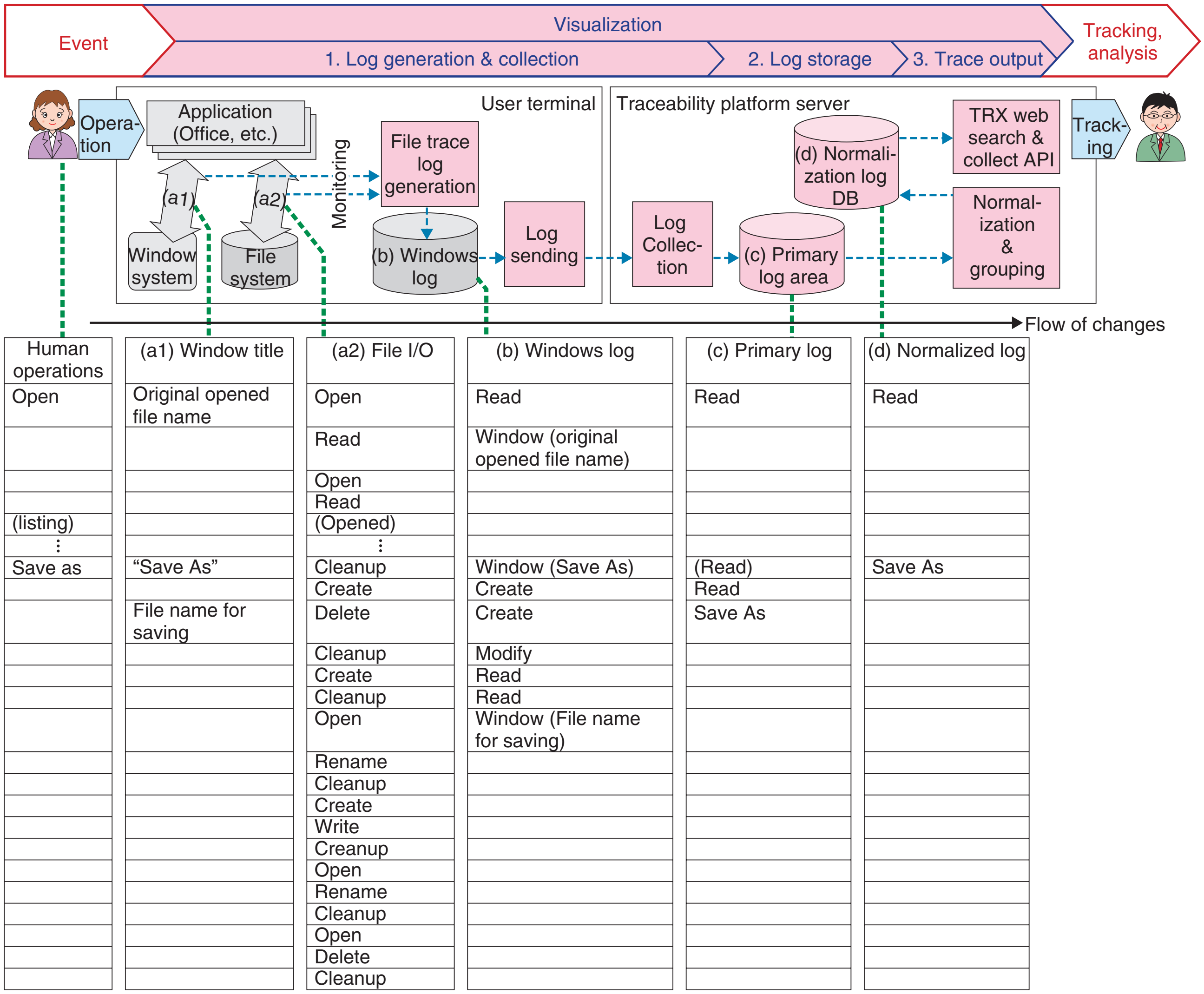


AP: application program
API: AP interface
DB: database
IaaS: infrastructure as a service
TCP: Transmission Control Protocol

Fig. 2. Traceability platform functions and log data flow.

(input/output) and window events of an application running on a Windows terminal, including the virtual desktop (Fig. 3(a1) and (a2)). Although the monitored events are very primitive such as *open file*, *read data*, *write data*, and *delete* for file I/O, the various TRX functions abstract the events in multiple stages to produce a log that corresponds more or less one-to-one with the original events, which are operations performed by human operators.

The primitive events occur in huge quantities, for example, 10,000 file I/O events per second. A high level of expertise is necessary in order to accurately



Human operations	(a1) Window title	(a2) File I/O	(b) Windows log	(c) Primary log	(d) Normalized log
Open	Original opened file name	Open	Read	Read	Read
		Read	Window (original opened file name)		
		Open			
		Read			
(listing)		(Opened)			
⋮		⋮			
Save as	“Save As”	Cleanup	Window (Save As)	(Read)	Save As
		Create	Create	Read	
	File name for saving	Delete	Create	Save As	
		Cleanup	Modify		
		Create	Read		
		Cleanup	Read		
		Open	Window (File name for saving)		
		Rename			
		Cleanup			
		Create			
		Write			
		Creanup			
		Open			
		Rename			
		Cleanup			
		Open			
		Delete			
		Cleanup			

Fig. 3. Accurate logging of file operation events.

convert them to a log with the number of operations per second that corresponds to human operations. For the TRX platform, we devised a customized commercially available product[*2] so that logs with especially high accuracy can be generated for popular applications such as Microsoft Office and Adobe Acrobat.

(2) Grouping for file operation event visualization

The flow of grouping for event visualization is illustrated in **Fig. 4**. The logs for files derived by copying information from the same file are normalized and assigned to a single group so that they can be handled together. The normalized log for file operations contains the information listed below.

- File name before operation: The name of the file before the operation was performed[*3]
- Date and time: The time and date the operation was performed
- User ID (identification): A code to identify the user who performed the operation
- Terminal address: The address of the terminal on which the operation was performed
- Operation type: Types of operations include create, copy, rename, move, delete, etc.
- File name after operation: The name of the file after the operation was performed[*4]

[*2] A customized version of the Log Audit Tracker product for logging file operations; produced by the dit Company Limited.

[*3] For ‘create’ operations, no “file name before the operation” is included.

[*4] For ‘delete’ operations, no “file name after operation” is included.

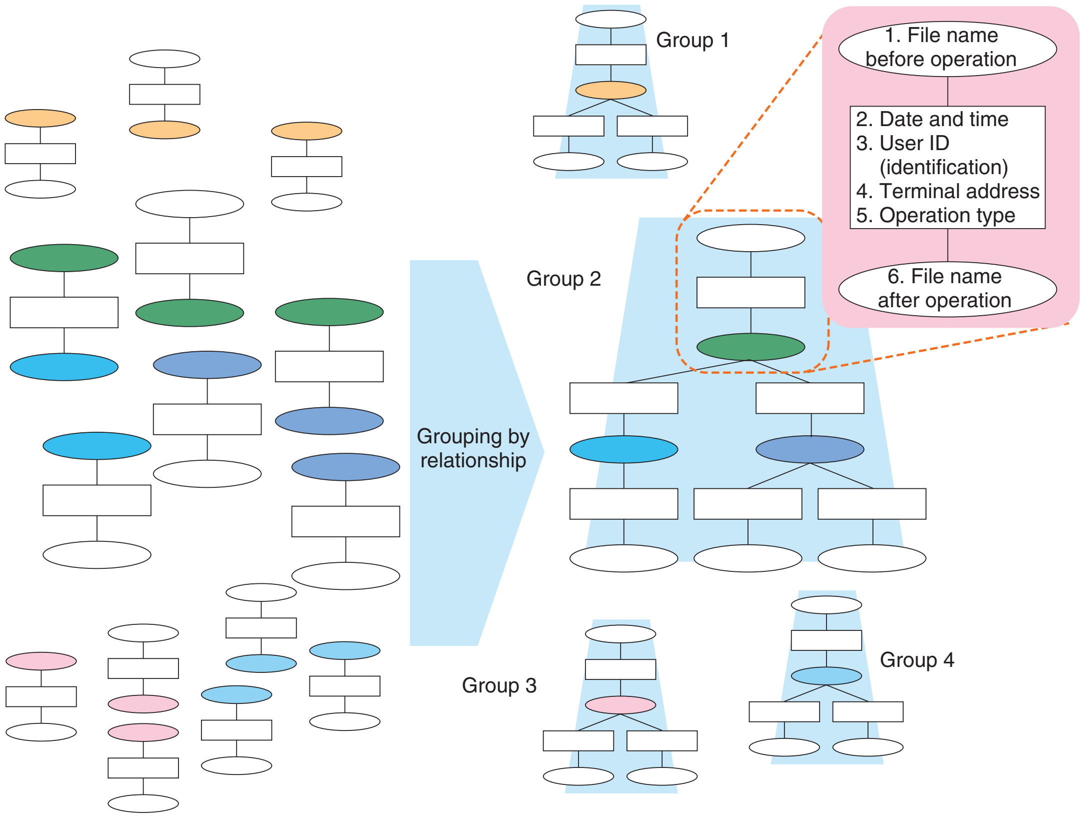


Fig. 4. Grouping for file operation event visualization.

In the grouping process of the normalized log, the names of the file before and after the operation are used. For example, if the file name before the operation for operation A is the same as the file name after the operation for operation B, then it is possible to relate the two operations as file operation events. Successively relating operations in this way makes it possible to manage files that have the same ancestor in the normalized log as a single group. This grouping can be processed at high speed by a proprietary algorithm that uses Hadoop Map Reduce [2]. Using groups constructed in this way makes it possible to search and display file operations at high speed.

(3) Scalability

File tracing in TRX uses a Hadoop distributed platform and is implemented with a scale-out-capable processing algorithm. The log storage can be scaled out from a single PostgreSQL relational database that can easily handle small-scale needs to HDFS (Hadoop Distributed File System) distributed processing by multiple units; HDFS processing is capable of conducting file tracing for offices with up to 100,000 employees.

2.2 Visualization of VM image copying (VM tracing)

In the past, a server was a fixed implementation in hardware, but the development of VM technology has changed the concept of a server to the form of a VM image file. This makes construction, addition, and copying of servers easy. Copying servers is advantageous, as it makes it easy to construct multiple servers when needed, such as for parallel processing tasks. However, there is also the disadvantage that license violations or information leaks may occur because the software license or critical information such as personal data that is included in the server image will also be copied. For these reasons, detecting the copying of servers has become a serious issue.

VM tracing is a function that detects the copying of a VM image and outputs a log when the virtual server boots up. This function embeds a notification mechanism called a *tracer* in the VM image and uses

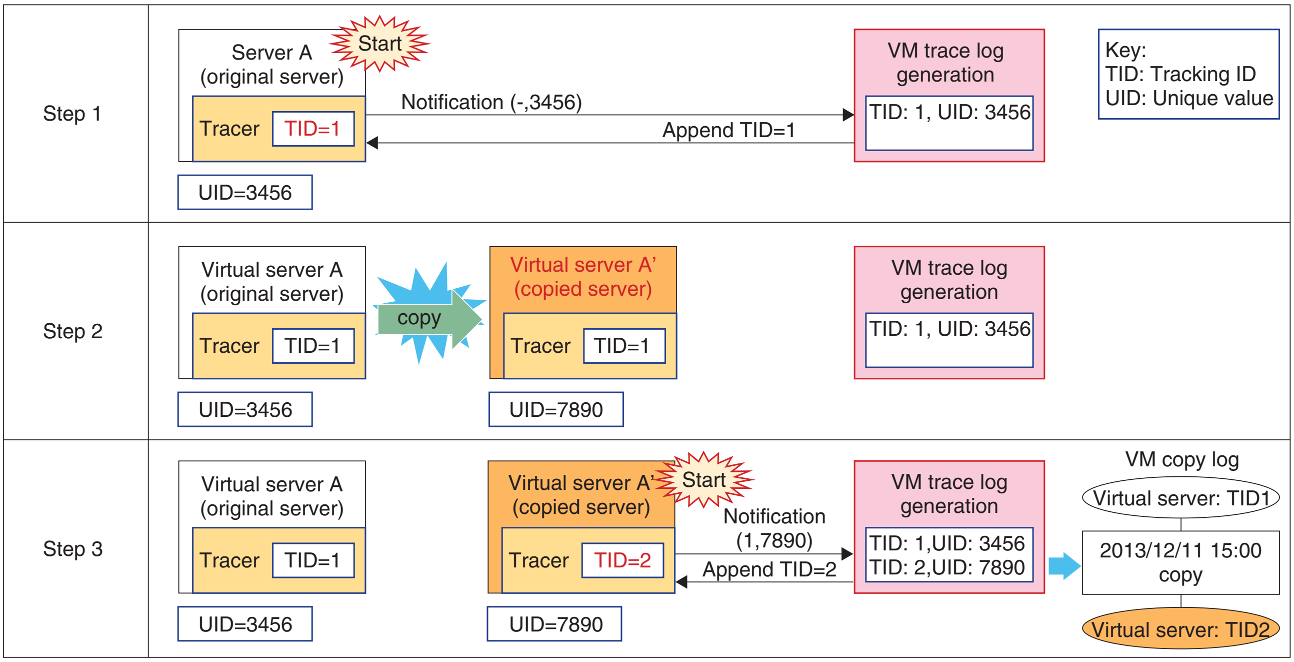


Fig. 5. VM trace log generation with active trace technology.

active trace technology to send a notification when the server boots up.

The mechanism of the VM trace log generator is illustrated in **Fig. 5**. In Step 1, the Unique Identification number (UID) 3456 is assigned to a virtual server instance when the instance is generated. Then the Tracking Identification number (TID) 1 is assigned and stored in the tracer when the virtual server is booted up. This pair of numbers is used for VM management. Consider the case in which virtual server A is copied, as in Step 2. When the copied server A' is booted up in Step 3, the notification contains TID=1 and UID=7890 (UID of server A'), but the VM trace log generator does not contain the pair of TID=1 and UID=7890; this means the VM image has been copied, and the event is logged. At the same time, a new TID=2 is assigned to the copied virtual server A'. In this way, VM image copying is detected and logged.

The remaining process of the generated log grouping is the same as for file tracing. This enables the VM image copy relationships to be tracked, and the grouping process enables the relationships to be visualized as a tree diagram. If VM images are restricted within a closed infrastructure-as-a-service (IaaS) provider system, tracking can also be done using the logs of hypervisors or other components. However, a function for uploading and downloading VM images makes it simple to move or copy images outside of the IaaS provider environment, which makes tracking difficult. Various products exist for protecting the data in a VM image, but they are ineffective when the entire VM image is copied. The VM tracing function of the TRX platform provides a post-incident method of tracking VM image copying, which has been a potential problem that may arise in the future.

3. Example of using visualization in the TRX platform

Examples of using the TRX traceability platform for file life-cycle management in an office and for virtual server operating system (OS) license management by a cloud provider are presented in **Table 1**.

3.1 File life-cycle management in an office

The need for strengthening internal control systems that pose a high risk of information leaks was described earlier. The TRX traceability platform can be used to manage the life cycle of files in a company. Installing an *agent* in the computers used by company personnel makes it possible to collect events related to company files, so that even in an environment where files are shared by multiple users, it is possible

Table 1. Example use of traceability platform.

	User (assumed)	Purpose	Effect
File life-cycle management in offices	- Company that has electronic files containing customer information or important internal information	- Strengthen internal control (electronic file management within the company)	- Confirm access and deletion of important information - Control information leaks - Rapidly respond to incidents
OS license management for virtual servers by cloud providers	- Cloud provider	- VM management - License management for OS etc.	- Easily confirm VM management in the cloud - Detect license violations for OS etc. - Control re-use of VMs that include royalty-bearing licenses

OS: operating system

to know when files are created, referenced, changed, copied, or deleted. It is also possible to retrace the path back to the original file, even when file names have been changed and the files have been moved between folders, or when derivative documents are created via common servers. This makes it easy to confirm that all files that contain erroneous information have been deleted and to find out whether important information has already been referenced by personnel.

3.2 Management of OS licenses in virtual servers by cloud providers

Cloud providers who rent out virtual servers to IaaS providers and other customers must properly manage the software licenses when the software included in the virtual servers is royalty-bearing software. Even when that is not the case, however, the software licenses may still require appropriate management.

Operations involving backup, redundancy, and the construction of test environments in the cloud provide many opportunities for copying VM images. It is also assumed that customers may download VM images enabling them to use services provided by other cloud providers. In such cases too, the VM tracing function described above can be used to ascertain that a VM image has been copied and booted up or that a problem concerning license management has occurred.

4. Future work

Introduction of the TRX traceability platform as a commercial product has begun. Specific points of improvement have come to light, and we are in the process of lowering the cost for a large-scale configuration, increasing the robustness of logging, and making improvements based on experience gained in operations.

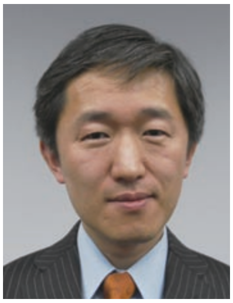
We intend to expand the range of tracing and are studying applications involving tracking of customer documents for which it is difficult to pre-install agents or other such mechanisms by applying the active trace technology that was developed for VM tracing to ordinary files [3].

References

[1] Ministry of Economy, Trade and Industry, Department of Commercial Information Policy, Office of Information Security Policy: "Information Security Management Guidelines for Use of Cloud Services," METI Website, April 1, 2011 (in Japanese). http://www.meti.go.jp/press/2011/04/20110401001/20110401001-2.pdf

[2] S. Nakahara and I. Tyou: "The Accountability of Cloud Services and Traceability Technology," IEICE Technical Report, Vol. 112, No. 22, ICM2012–8, pp. 81–85, 2012.

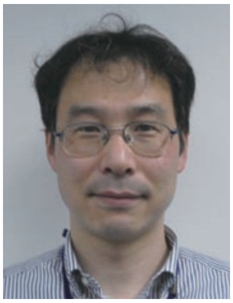
[3] I. Tyou, J. Akiba, T. Matsumura, and T. Motoda: "A Technology for Tracing General File Operations," Proc. of CSS (Computer Security Symposium) 2013, pp. 832–839, Zhangjiajie, China.



Toshihiro Motoda

Senior Research Engineer, Supervisor, Security Management Promotion Project, NTT Secure Platform Laboratories.

He received the B.E. and M.E. in computer science and engineering from Toyohashi University of Technology, Aichi, in 1987 and 1989, respectively. He joined NTT Communications and Information Processing Laboratories in 1989 and studied end-user computing. He is currently studying an accountable security and traceability platform. He is a member of the Information Processing Society of Japan (IPSJ).



Junya Akiba

Senior Research Engineer, Supervisor, Security Management Promotion Project, NTT Secure Platform Laboratories.

He received the B.E. and M.E. in applied physics from Tohoku University, Miyagi, in 1990 and 1992, respectively. He joined NTT Switching Systems Laboratories in 1992 and studied advanced intelligent network systems. He is currently studying accountable security including audit and log management. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.

Takeshi Nagayoshi

Senior Research Engineer, Supervisor, Security Management Promotion Project, NTT Secure Platform Laboratories.

He received the B.E. and M.E. in electrical and electronics engineering from Sophia University, Tokyo, in 1990 and 1992, respectively. He joined NTT Information and Communication Systems Laboratories in 1992 and studied information security systems. He is currently responsible for developing the traceability platform.

Kaku Takeuchi

Research Engineer, Security Management Promotion Project, NTT Secure Platform Laboratories.

He received the B.E. and M.E. in computer science from Keio University, Tokyo, in 1992 and 1994 respectively. He joined NTT Software Laboratories in 1994 and was engaged in studying software engineering. He is a member of IPSJ.