# Cloud Traceability (CBoC TRX)

*Shinichi Nakahara†, Naoto Fujiki, and Shigehiko Ushijima*

**Abstract**

In this article, we introduce a cloud forensics service for cloud security that can visualize a sequence of operations and the lifecycle of data and a traceability platform for implementing this service. This visualization contributes to overcoming the security worries and concerns about data leakage and unintended deletion that the majority of potential cloud users have over the inability to see the progress or operation of cloud services. The infrastructure has an architecture that allows various functions to be added or substituted, allowing large traceable logs of various types to be linked together to suit users' needs.

## 1. Introduction

For many potential new users, the inability to know the operational state of cloud services or where the service and data themselves actually reside is a barrier to introducing such services [1]. Therefore, it is important to inform users about what kind of processing the cloud service is performing and who performed what human actions and where their data is located [2], [3]. A traceability platform overcomes these non-transparency concerns by reproducing and displaying the chain of events from log information indicating human operations, file transfers, and process activity as well as information from related systems such as authentication and equipment management systems.

## 2. Cloud forensics

The cloud forensics service (**Fig. 1**) is a value-added service aimed at allowing anxious users to receive cloud services with confidence, as well as letting cloud services providers confidently provide services that will satisfy users, by giving the cloud system the ability to explain who (or what) did what, as well as when, where, in what way, and with what result. Specifically, the cloud forensics service maintains evidence (logs) of events occurring in the cloud and links them together according the users' or operators' intentions to provide event verification and visualization. For example, it enables checking of the lifecycle of a user file (creation, editing, transfer, and deletion) (from the user's perspective), checking of which operator performed which operations on a user's environment configuration (operator's perspective), and verification of whether the system is operating according to regulations (auditor's perspective).

For this purpose, the system must maintain traceable event information in the 4W1H1R (when, where, who, what, how, and what result) format.

## 3. Traceability platform architecture

To implement such cloud forensics, we have started developing the Common IT Bases over Cloud Computing (CBoC) cloud traceability system (CBoC TRX), which gathers, stores, links together, and refers to logs of events occurring in the cloud (IT: information technology). The system architecture consists mainly of a log-trace section, which gathers and normalizes large volumes of logs output from servers and access points; the log security section, which maintains the security of the logs; and the log operation management section, which performs log provisioning, which enables different logs to be linked and traced (**Fig. 2**).

† NTT Information Sharing Platform Laboratories
Musashino-shi, 180-8585 Japan

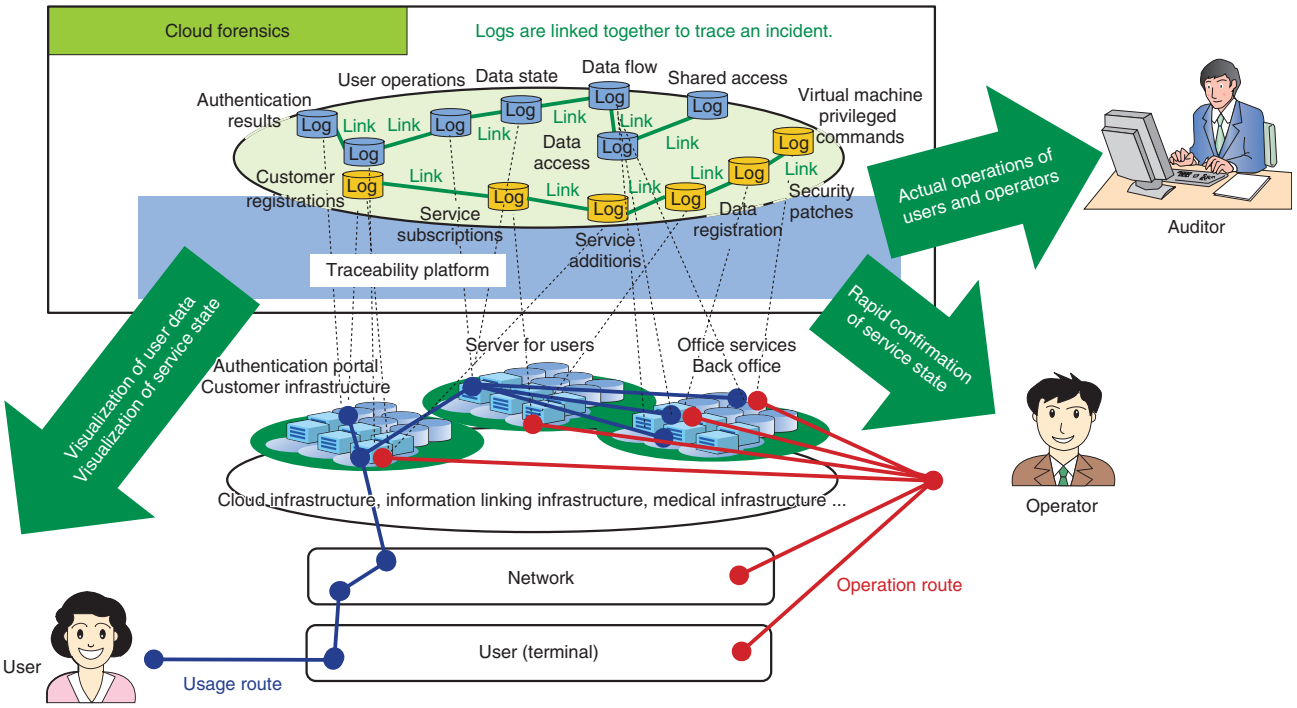
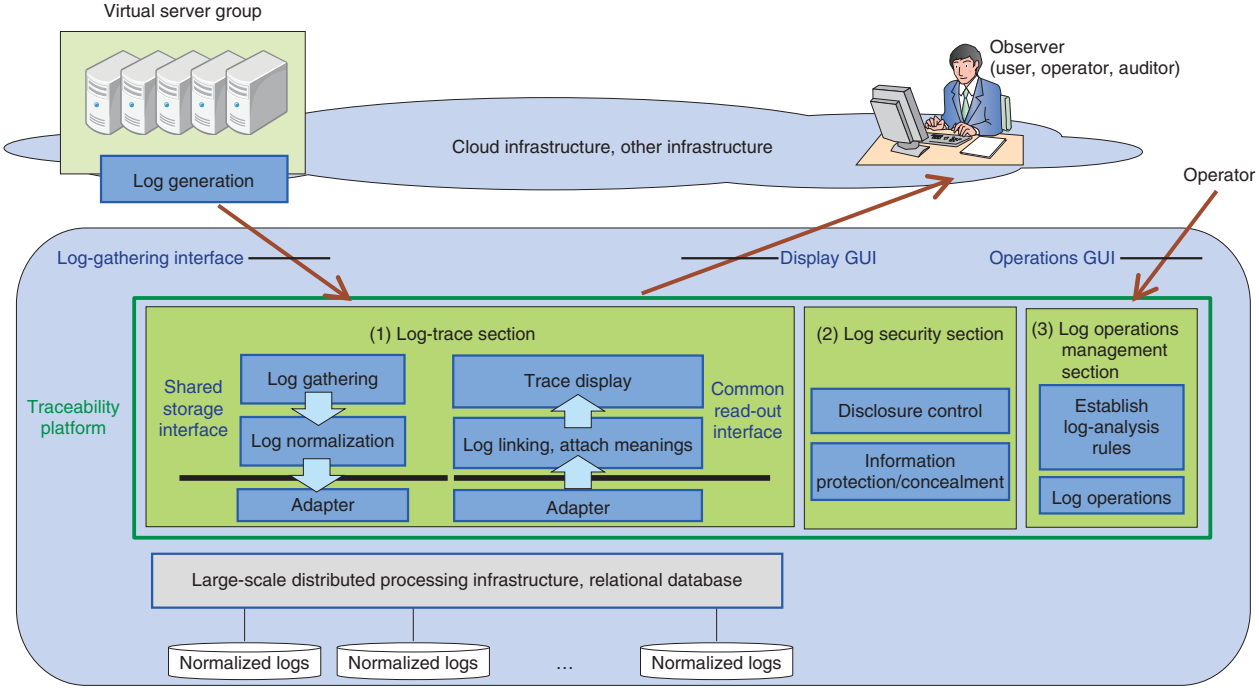



Fig. 1. Cloud forensics.




GUI: graphical user interface

Fig. 2. Traceability platform architecture.

The log-trace section processes, normalizes (mapping to 4W1H1R format), and stores the large volume of input logs. To maintain storage and data processing scalability, the storage of large logs can be linked to large-scale distributed processing infrastructures such as CBoC Type 2 or Hadoop[*1].

To avoid dependence on a particular infrastructure when linking with large-scale distributed processing infrastructures in particular, we decided on a common access interface (storing and reading out) for programs being used, and infrastructures are accessed through adapters supporting this interface. The log-trace section and the log security section were designed with a reusable component architecture, with uniform access interfaces between the functional blocks, and with independence at the launch and unit operation level being maintained.

## 4. Infrastructure features

Building log security, scalability, and log normalization into the log infrastructure brings the following features to the traceability platform and improves usability.

### 4.1 Log security

Log evidence needs to be admissible. To improve log admissibility, three measures are taken: (1) the accuracy of terminal operation log timestamps for the Windows operating system is improved by linking to an NTP (network time protocol) server, (2) input logs (primary logs) have file-level anti-interpolation protection provided by a tamper detection function, and (3) input logs are normalized and complemented to ensure that there is sufficient information to reconstruct actual events.

Evidence of any leakage of log data or tracing information is needed. To overcome one of the concerns about cloud technology, which is that information could leak to other users or operators, stored logs can be encrypted. Log data disclosure is also restricted according to the viewer's authorization, and user names and other private information are not displayed except to the actual person.

### 4.2 Scalability

To process the large volume of logs for cloud services, we implement scalability in the collection and parallel processing of log data using Flume (open source software) for the process of gathering logs from many locations, and we link to systems such as CBoC Type 2, Map/Reduce, and the Hadoop Distributed File System for storage and search processing.

### 4.3 Log normalization

Input log data (called log messages, since they are normally input as datagrams) is normalized by separating it into elements and mapping each element to a 4W1H1R category. If the information is insufficient for tracing at this point, the data is complemented with information from other logs or other pre-defined external data. Log normalization has three effects.

(1) Logs can be linked to support a variety of trace requirements.

By clearly defining the meaning of individual log message elements, we enable a single log message to be used for mapping trace-service requests for various purposes. In this way, conventional logs output for specific purposes can be used to reproduce actual events.

(2) Trace scopes can be extended by adding additional logs.

New logs are normalized when they are added to the traceability platform, so if there are log elements with the same meaning as elements already stored in the infrastructure, or if it is possible to add logs from a new time period or information about new services or functions in order to define how the logs are connected, the scope of the trace can be extended temporally or spatially.

(3) Enhance adaptability to key-value store

A fast search can be implemented for keys such as user or file names by storing data in the key-value store (KVS[*2]) format and by using 4W1H1R elements as key data.

### 4.4 Integration into the log infrastructure

To enable the traceability platform to handle logs in any existing format, we made it possible in the log operations management section to define parsing rules for logs and rules mapping from log elements to 4W1H1R elements. For this purpose, we provided a graphical user interface, which makes it easy to integrate new logs into the infrastructure, which already has normalized log data.

*1 Hadoop: A Java software framework for distributed processing of large-scale data.

*2 KVS is a format commonly used for Internet searching and other applications.

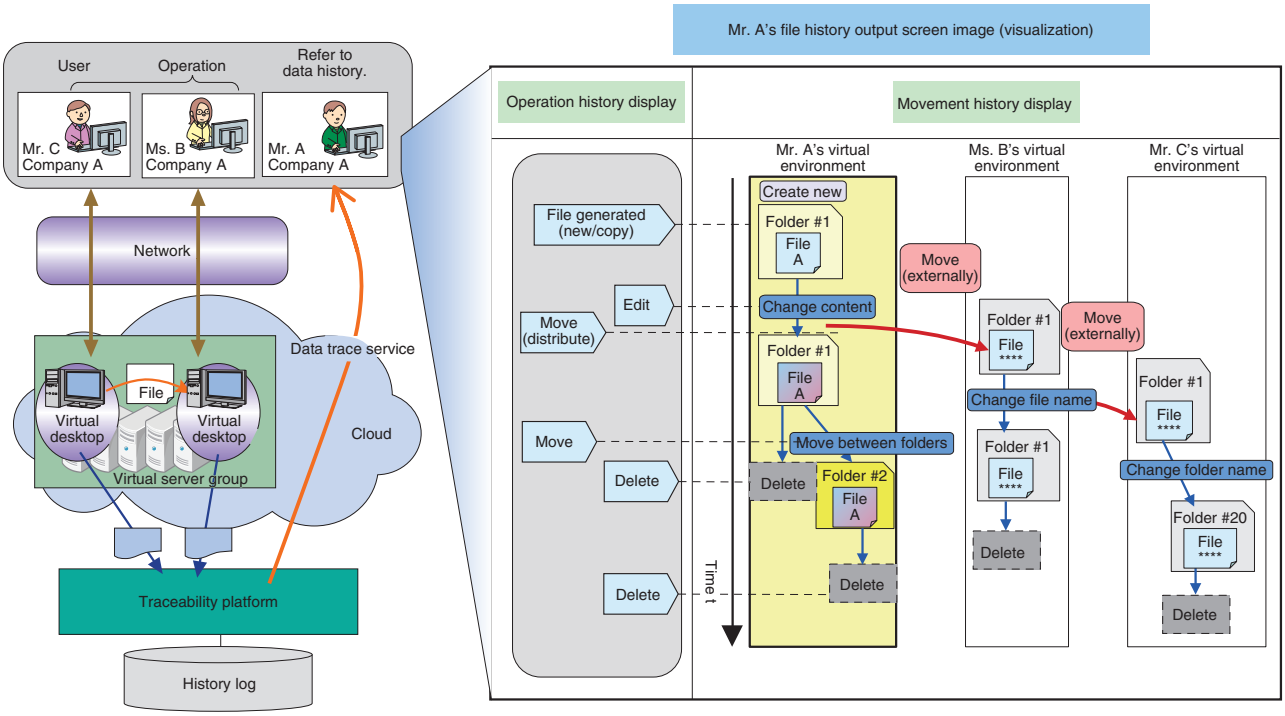



Fig. 3. Data trace concept.

## 5. Trace service concept

### 5.1 Data tracing

The log of a user's operations at a terminal is summarized and managed by CBoC TRX in 4W1H1R format, revealing what operations the user did on what files (data) (**Fig. 3**). This allows the record of what operations—creating, editing, copying, moving, or deleting—were done when and by whom on a given file.

The user can visually check the historical sequence of file creations, deletions, and transfers to other people even though the data is entrusted to the cloud operator, so the history of a file can be checked easily, including its source, where it went, or whether it was deleted according to requirements. In principle, for files passed to other people, the history is displayed with personal information such as file names anonymized, e.g., letters replaced by meaningless characters such as *, but it is also possible to suppress any display of this information.

### 5.2 Operation tracing

As with data tracing, terminal operation logs (4W1H1R) are also collected. The system summarizes and links together both user and operator logs, so the relationships between operations of both can also be seen (**Fig. 4**).

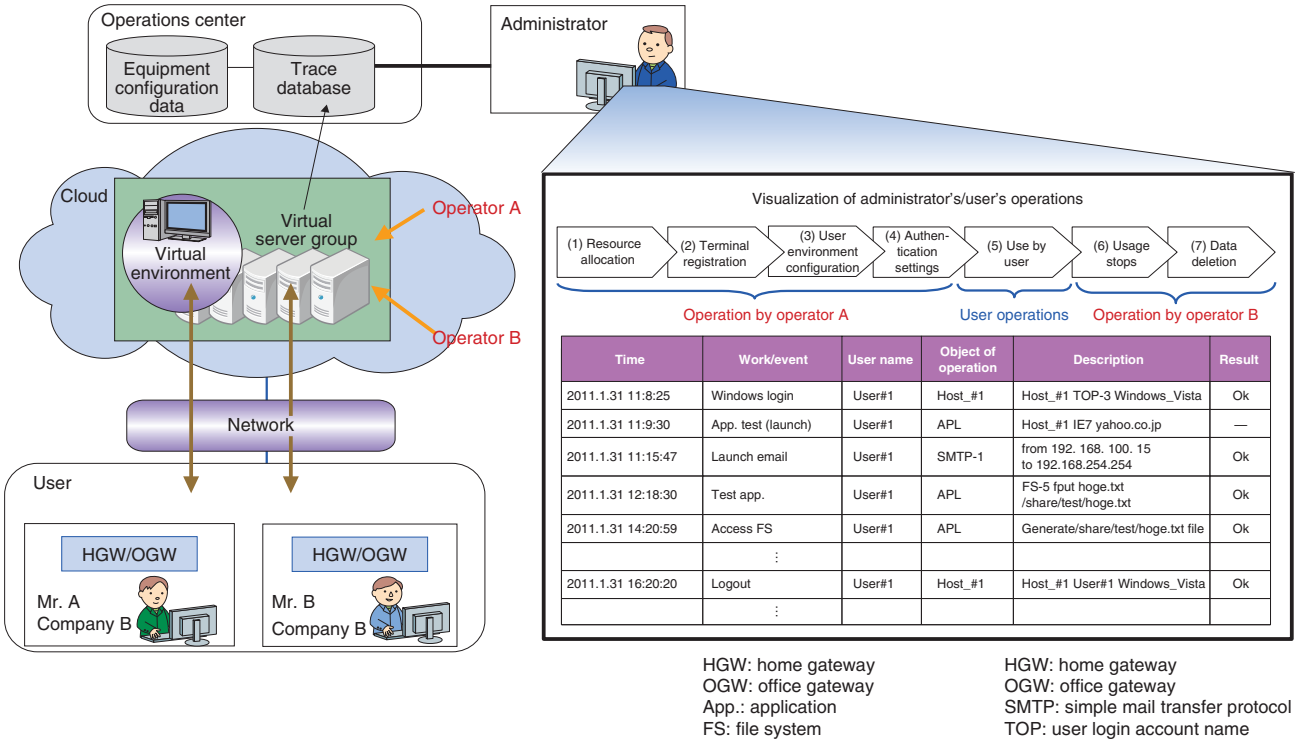
Normally, it is impossible to link user logs with operator logs, but adding additional information that can be parsed and linked, such as operation logs from the same virtual machine or computer, enables the sequence of operations to be reconstructed despite the barrier of independence between user and operator.

Even if a cloud user's environment is configured by different operators, the multiple logs from individual operators are linked, and the sequence of operations can be recognized. This also applies when a service system is composed of multiple services and servers. Using the common information in the logs enables operations spanning people, services, and systems to be rapidly linked together and understood.

## 6. Future initiatives

### 6.1 Strengthen security

The security functions of CBoC TRX were implemented from the perspective of making the logs admissible and preventing information leaks, but we




| Time | Work/event | User name | Object of operation | Description | Result |
|---|---|---|---|---|---|
| 2011.1.31 11:8:25 | Windows login | User#1 | Host_#1 | Host_#1 TOP-3 Windows_Vista | Ok |
| 2011.1.31 11:9:30 | App. test (launch) | User#1 | APL | Host_#1 IE7 yahoo.co.jp | — |
| 2011.1.31 11:15:47 | Launch email | User#1 | SMTP-1 | from 192. 168. 100. 15 to 192.168.254.254 | Ok |
| 2011.1.31 12:18:30 | Test app. | User#1 | APL | FS-5 fput hoge.txt /share/test/hoge.txt | Ok |
| 2011.1.31 14:20:59 | Access FS | User#1 | APL | Generate/share/test/hoge.txt file | Ok |
| | ⋮ | | | | |
| 2011.1.31 16:20:20 | Logout | User#1 | Host_#1 | Host_#1 User#1 Windows_Vista | Ok |
| | ⋮ | | | | |

HGW: home gateway
OGW: office gateway
App.: application
FS: file system

HGW: home gateway
OGW: office gateway
SMTP: simple mail transfer protocol
TOP: user login account name

Fig. 4. Operations traceability.

plan to strengthen them further in the future taking into consideration issues regarding the use of public information and protection of private or sensitive information and using additional technology such as long-term digital signatures and evidence record syntax.

### 6.2 More active use of logs

We will make improvements in log linking through techniques such as (1) resolving the various IDs (identities) of the same person on different services and using them to link logs and (2) integrating logs with other logs by recognizing the configuration between a physical device and logical devices. This is expected to meet the accountability obligations of cloud service providers and improve operational efficiency.

## References

[1] Ministry of Economy, Trade and Industry, "Survey Report on Information Security Audits of Cloud Services," Jan. 2010 (in Japanese).
[2] S. Nakahara and H. Ishimoto, "A Study on the Requirements of Accountable Cloud Services and Log Management," Proc. of APSITT 2010, pp. 1–6, Kuching, Malaysia, 2010.
[3] Hewlett Packard report. http://www.hpl.hp.com/techreports/2011/HPL-2011-38.pdf

**Shinichi Nakahara**
Senior Research Engineer, Supervisor, Security Management SE Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in electrical engineering from Osaka University in 1984 and 1986, respectively. He joined NTT Yokosuka Laboratories in 1986, where he engaged in research on operating systems. He is currently studying cloud security and data & operation traceability in the cloud. He is a member of IEEE and the Information Processing Society of Japan (IPSJ).

**Naoto Fujiki**
Senior Research Engineer, Security Management SE Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in precision engineering from Niigata University in 1989 and 1991, respectively. He joined NTT in 1991 and studied information sharing systems, network operation systems, and network security. He is currently studying traceability systems. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and IPSJ.

**Shigehiko Ushijima**
Senior Research Engineer, Supervisor, Communication Platform SE Project, NTT Information Sharing Platform Laboratories.

He received the B.E. and M.E. degrees in electronic engineering from Keio University, Kanagawa, in 1986 and 1988, respectively. He joined NTT Communication Switching Laboratories in 1988. His recent research area is cloud computing architecture and security. He is a member of IEICE.