A Digital Forensic Framework for Automated User Activity Reconstruction

Jungin Kang, Sangwook Lee, and Heejo Lee

Division of Computer and Communication Engineering,
Korea University
Seoul, Republic of Korea
{keijin,ook7777,heejo}@korea.ac.kr

Abstract. User activity reconstruction is a technique used in digital forensic investigation. Using this technique, digital forensic investigators extract a list of user activities from digital artifacts confiscated at the crime scene. Based on the list, explicit knowledge about the crime, such as motive, method, time, and place, can be deduced. Until now, activity reconstruction has been conducted by manual analysis. This means that the domain of the reconstructed activities is limited to the personal knowledge of the investigators, so the result exhibits low accuracy due to human errors , and the process requires an excessive amount of time. To solve these problems, this paper proposes a digital forensic framework-SigDiff for automated user activity reconstruction. This framework uses a signature-based approach. It comprises an activity signature generation module, signature database, digital artifact collection module, and activity reconstruction module. Using SigDiff, the process of user activity reconstruction can be performed accurately with a high retrieval rate and in a reduced time span.

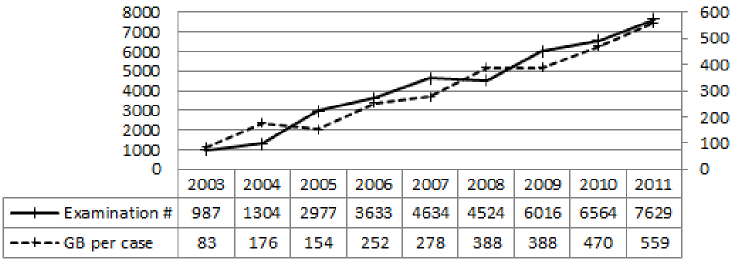
Keywords: digital forensic framework, activity reconstruction, signature-based forensics.

1 Introduction

With the increasing use of personal digital devices, the number of crimes that use digital devices as tools is rising. Criminals use digital devices to find information about victims or buy drugs and weapons. In some cases, the digital devices are used as tools for cybercrimes, such as information leakage and phishing. To respond to such crimes, investigators from governments and enterprises use digital forensic techniques. The investigators analyze digital devices to extract digital artifacts such as Web search histories and program histories . These artifacts can be evidence of user activities that were performed on the device. Using the extracted activity information, the investigators plan the direction of the investigation or present the artifacts to a court as proof of the guilt of a suspect.

According to FBI statistics [1], the number of digital forensic investigations and the storage size per case are increasing (Fig.1).The rise in storage size means

R.H. Deng and T. Feng (Eds.): ISPEC 2013, LNCS 7863, pp. 263–277, 2013.



	2003	2004	2005	2006	2007	2008	2009	2010	2011
Examination #	987	1304	2977	3633	4634	4524	6016	6564	7629
GB per case	83	176	154	252	278	388	388	470	559

Fig. 1. Increases in the number of digital forensic examinations and storage size

that the time required for each analysis is increasing. Garfinkel [2] classified this problem as the upcoming digital forensic crisis that needs to become a focus.

To solve the problem, digital forensic investigators use digital forensic tools to analyze digital artifacts. These tools abstract the digital data into easily understandable formats or automatically extract some important information. For example, listing the files on a disk or extracting an Internet history are frequently used functions of digital forensic tools.

However, the current tools only list the artifacts that are extracted from digital devices. This means that the reasoning process about what user activity generated the artifact is still manual work for an investigator. For example, when a user executes a messenger software on a digital device, the software will leave file and registry artifacts on the device. Current digital forensic tools only display the list of file and registry artifacts to the investigator. To deduce the messenger activity, the investigator should have additional knowledge about the relationship between the artifacts and the messenger activity.

The manual process causes the following problems: first, the domain of the reconstructed activities is highly limited by the personal knowledge and experience of the investigators; second, the activity reconstruction process is time-consuming, and the results suffer from low accuracy.

In this paper, we propose a digital forensic framework SigDiff to solve the problems of manual user activity reconstruction. This framework adopts the signature-based approach that is widely used for rapid but precise identification of data in numerous systems, such as antivirus engines or intrusion detection systems. SigDiff comprises an activity signature generation module, activity signature database, digital artifact collection module, and activity reconstruction module. Using these components, SigDiff accumulates user activity signatures based on a predefined activity model with corresponding artifacts. The activity signatures are used in digital forensic investigation for automated user activity reconstruction with a higher retrieval rate, increased accuracy, and reduced time.

The remainder of this paper is organized as follows: Section 2 presents the background on digital forensics and user activity reconstruction. Section 3 introduces previous work on signature-based user activity reconstruction. Section 4 provides a detailed description of SigDiff and its components. Section 5 presents the proof-of-concept tools and evaluation results. Finally, in Section 6 we list future research directions with our conclusions.

2 Background

When a digital device is used , the user gives the device some input for a specific purpose. The digital device processes the series of inputs and displays the respective results to the user. In the process, some artifacts will be left on the physical media of the device.

For example(Fig.2), the user may want to send messages to someone. Web browser software is used to access the Website of a messenger, download a client installer package, install the messenger, logon to the messenger, and send messages. When this series of user activities is conducted, the Web browser software, messenger installation package, and messenger client may leave Internet history, registry, and file artifacts [3].

Digital forensics is the process in which investigators collect digital devices from a crime scene, recover artifacts from the devices, reconstruct suspect activities from the artifacts, and present the artifacts or devices to a court as evidence for the activities.

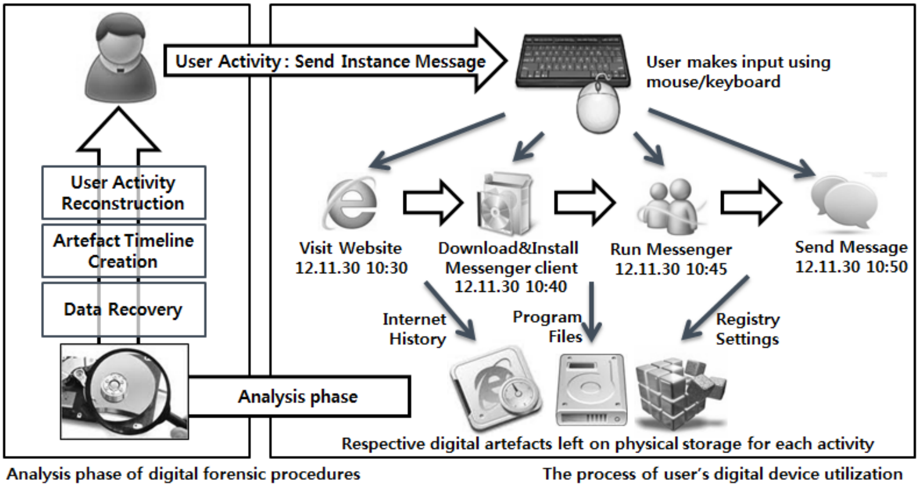

User Activity : Send Instance Message
User makes input using mouse/keyboard
User Activity Reconstruction
Artefact Timeline Creation
Data Recovery
Analysis phase
Visit Website 12.11.30 10:30
Download&Install Messenger client 12.11.30 10:40
Run Messenger 12.11.30 10:45
Send Message 12.11.30 10:50
Internet History
Program Files
Registry Settings
Respective digital artefacts left on physical storage for each activity
Analysis phase of digital forensic procedures
The process of user's digital device utilization

Fig. 2. Example of process for digital device utilization

In the previous example, the user may be under indictment for technology leakage. Investigators confiscate the digital device of the user and extract artifacts from the device. From the artifacts, the investigators can reconstruct a series of user activities that are related to the instant messaging.

The digital forensic investigation is performed by means of a predefined investigational procedure(Fig.3). Numerous procedural models have been published

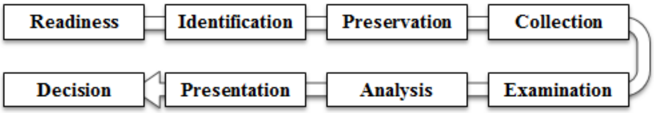


Fig. 3. DFRWS model with readiness phase

and adopted by various organizations. In this paper, the DFRWS model [4] proposed in the first Digital Forensic Research Workshop (DFRWS) is adopted, with an additional forensic readiness phase [5].

First, in the readiness phase, the organization prepares detailed procedures, tools, and human resources to prepare for an investigation. In the identification phase, the organization identifies an incident and arranges resources for an investigation. Subsequently, investigators preserve the crime scene and collect digital devices during preservation and collection phases. In the examination and analysis phases, the investigators gather meaningful information from the media seized. Finally, in the presentation phase, the investigators present the information and evidence to the court for a decision.

Typically, the collected evidence consists of physical media that store data in a bitwise manner. In the analysis phase, the investigators should interpret sequences of bits to obtain more meaningful information. In other words, the investigators abstract the data from bit level to a higher level. This process has been defined as digital forensic abstraction by Carrier [6].

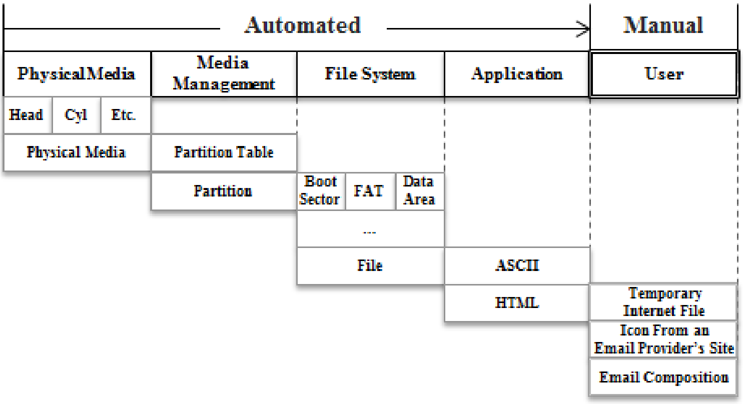


Fig. 4. Example of Carriers abstraction layers with a user layer for an HTML file

In Carriers model, the data on physical media can be successively abstracted into a media management layer, file system layer, and application layer. For some digital forensic investigations, there are requirements for one or more layers that describe user activity. For example(Fig.4), HTML data abstracted in the application layer may be interpreted as a temporary Internet file that was created by user activity for email composition. The data abstracted in the user layer are useful for digital investigations that target a person. Using the abstracted activities, such as for Web searches, email, and SNS, the investigators can easily deduce information such as the characteristics, mentality, or recent location of the suspect.

To move efficiently up the abstraction levels, some digital forensic tools such as EnCase [7] and FTK [8] have been developed. These tools are widely used in the field by real forensic investigators. The tools interpret the collected physical-level digital data into a human readable format, mostly at the application level.

However, most current tools do not support abstraction from the application level to the user level. Consequently, investigators necessarily analyze millions of application-level artifacts by a manual process. These circumstances cause the following problems:

1. Excessive Time Consumption
   The investigators primarily identify application-level artifacts individually. In the example of the HTML file, the investigator first extracts some meaningful words from the file name and file data. After that, information is gathered from the words and the source activity is deduced. Although there are keyword-based searching techniques [9], timeline-based approaches [10] and visualization techniques [11] for reducing the amount of data to analyze, this manual process still requires an excessive amount of time.
2. Low Retrieval Rate
   To deduce the source activity, investigators should have previous knowledge of the activity. In other words, investigators are unlikely to retrieve activities for domains that are unfamiliar. Moreover, in most cases, no meaningful words from an artifact can be recognized by the investigator, causing a disregard of the artifact. For these reasons, a considerable number of user activities are omitted in the analysis phase, resulting in a low retrieval rate.
3. Decreased Accuracy
   The fact that the activity reconstruction process is performed manually means that there can be human errors. Investigators may misunderstand the meaning of extracted words, resulting in an incorrect result. Although there are examination environments such as Vise [12], the examination is difficult to perform for the entire reasoning process due to the limitation of available time.

3 Related Work

To solve the problems of time consumption, retrieval rate, and accuracy, there has been work that shares information about artifacts and source activities. The researchers have analyzed user activities with frequently used applications such as messengers [13] and Internet download managers [14]. With the information gathered, some tools have started to support limited user activity abstraction; for example, the extraction of USB storage activities [15] or Internet activities[16].

Despite the efforts that have been made, the amount of shared information is still insufficient, and the tools that provide fixed extraction functions have limited scalability. Thus, a scalable automated digital forensic system that rapidly performs activity reconstruction with a high retrieval rate and high accuracy is required. James[17] and Hargreaves[18] adopted a signature-based approach to solve the problems. Using the signature-based approach, the signature of information is stored in a database that is queried when the information is required. This approach has the advantage that known information can be searched in a fast but accurate way, and it has been adopted in various identification systems such as antivirus software and IDS/IPS . Although the retrieval rate is limited by the size of the database, at least this system is scalable and retrieves information effectively.

James[17] has proposed a novel approach to signature-based activity reconstruction. A simple activity is performed repeatedly in a virtual machine, and then artifacts with changed timestamps are filtered out. The signature is generated using the generalized path string of the artifact. Hargreaves [18] proposed a script-based signature generation method. The signature is applied to the artifact super-timeline to reconstruct higher-level events.

The previous work on signature-based user activity reconstruction was focused on adopting the approach for digital forensics, thus simple methods of activity signature generation were proposed. In this paper, we continue the work to describe a signature-based digital forensic framework that covers the entire procedure for user activity reconstruction. Using this framework, investigators can automatically reconstruct complex user activities in a significantly reduced timespan, but with a higher retrieval rate and increased accuracy.

4 SigDiff: Signature-Based Digital Forensic Framework

SigDiff, the signature-based digital forensic framework, is composed of the following parts (Fig.5): an activity signature generation module, an activity signature database, a digital artifact collection module, and an activity reconstruction module.

The activity signature generation module is used to construct the activity signature database. This module is used in the forensic readiness phase. It generates signatures using a predefined user activity model and sends those signatures to the database. The digital artifact collection module is used in the collection phase of a digital investigation. It extracts artifacts from a collected digital device or directly collects artifacts from a live digital device. The activity reconstruction module matches the collected artifacts to the signatures stored in the database and reconstructs the user activity timeline. This module is used in the forensic examination and analysis phase.

4.1 Activity Signature Generation

The activity signature generation module performs a series of processes to generate a user activity signature (Fig.6).

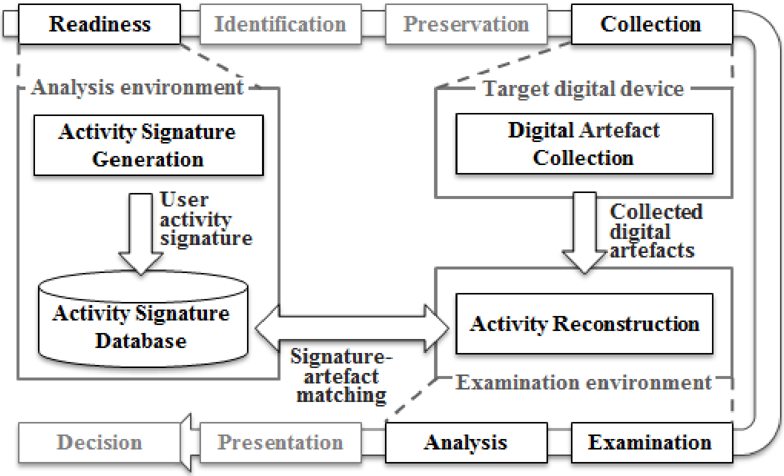


Fig. 5. SigDiff architecture

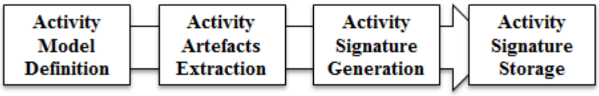

Activity Model Definition
Activity Artefacts Extraction
Activity Signature Generation
Activity Signature Storage

Fig. 6. Procedures for activity signature generation

First, in the activity model definition phase, the investigator defines the user activity model. Using the model, the module extracts artifacts of the activity model, generates the signature, and stores the signature in the activity signature database.

Activity Model Definition. When a user generates events such as mouse or keyboard input, the application processes the corresponding tasks. In the process, the application may leave artifacts on physical media. A user activity is a series of user-level events(Fig.7) performed for a single purpose. For example, when a user performs an activity defined as Install messenger software, a series of user inputs from clicking on Next and Finish buttons will be sent to the software installer. The software installer receives the inputs and writes messenger files on the physical media of the device. After the activity has been performed, there will be installed files. In other words, the *Activity artifacts*.

The ideal case of user activity definition is that the defined user activity includes only a single user-level event. For example, user inputs for clicking buttons of the installer may be defined as multiple activities such as first clicking the Next button and then clicking the Finish button. However, in this case, the number of activities defined will be too large, and the time required for defining

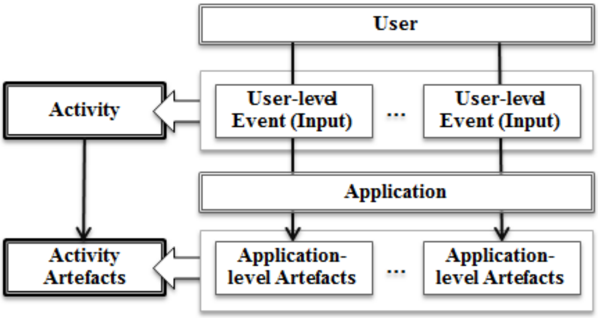


Fig. 7. Model for user activity and its artifacts

activities will be excessive. Thus, the investigator may bind a series of events that are performed for the same purpose, and define this as one activity.

There are two approaches for defining user activities with respect to corresponding user events: the model-first approach and the event-first approach.

Model-First Approach. In the model-first approach, the user activity model for a topic is first defined. The activity model is a sort of usage scenario and is formulated in a finite-state machine (FSM). Each state of the FSM is a state after some activity has been conducted. Each transition function represents an activity, which is a series of user events. For example, a simple activity scenario for a messenger can be defined with the model in (Fig.8). Once an activity model is defined, a series of user events is defined for each transition function.

Using the model-first approach, the activity artifact required by the investigation can be extracted quickly and with flexibility. In other words, this approach is adequate when the investigator has a crime situation composed of a series of activities.

Event-First Approach. In the event-first approach, a series of user-level events for a topic is first collected. The activity model FSM is defined with refined events. If the user events are collected from a large number of users, then the activity model can reflect a trend in user activity. Monitoring, collecting user events, and interpreting the events as user behavior at the user interface are research areas of human-computer interfaces[19]. Further research is required from a digital forensics perspective.

Activity Artefacts Extraction. The activity artifacts extraction process is performed on the basis of the predefined user activity model. In this framework, the user events are replicated on a virtual machine to extract the respective activity artifacts. In the virtual machine, an extraction method based on either state comparison or system monitoring is used to extract activity artifacts.

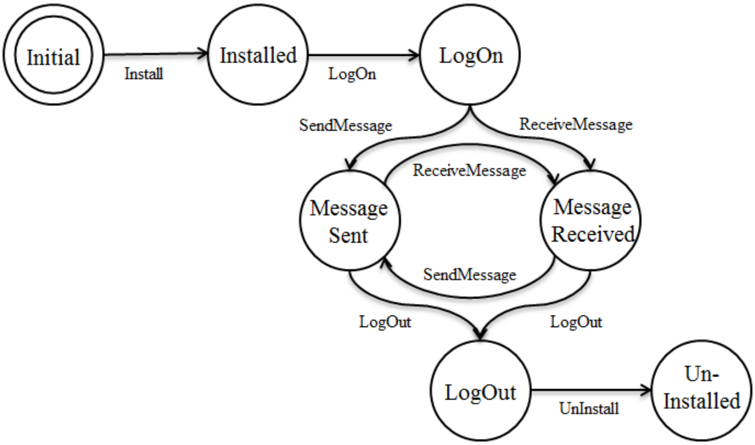


Fig. 8. Example of a messenger activity model

State-Comparison-Based Extraction. Using the state-comparison-based extraction method (Fig.9), a set of virtual machine snapshots is generated corresponding to the states in the predefined user activity model. Thus, the differences between two connected snapshots can be regarded as artifacts of an activity.

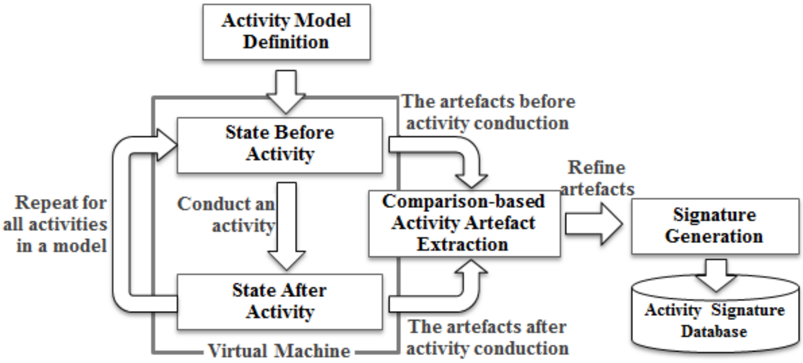

Activity Model Definition
The artefacts before activity conduction
State Before Activity
Refine artefacts
Repeat for all activities in a model
Conduct an activity
Comparison-based Activity Artefact Extraction
Signature Generation
State After Activity
Activity Signature Database
The artefacts after activity conduction
Virtual Machine

Fig. 9. State-comparison-based extraction

System-Monitoring-Based Extraction. The system-monitoring-based extraction method (Fig.10) does not save all the snapshots. A series of user inputs from the activity model is performed continuously. However, the alterations generated in the virtual machine are monitored in real time. For example, system tracking functions such as CreateFile or RegCreateKey in Windows are called to extract artifacts. Once all the activities in a model are performed, sets of artifacts and user activities are matched.

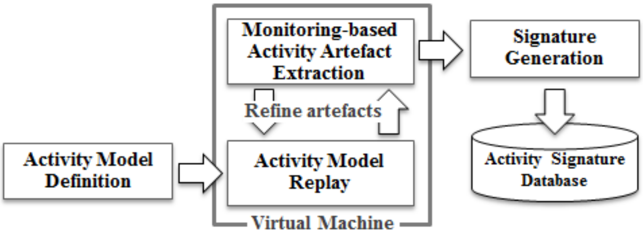

Monitoring-based Activity Artefact Extraction
Refine artefacts
Activity Model Replay
Virtual Machine
Activity Model Definition
Signature Generation
Activity Signature Database

Fig. 10. System-monitoring-based extraction

The comparison-based extraction method has the advantage of scalability. If the source activity model is extended, the new activities and states can easily be added to the saved virtual machine snapshots. However, additional time is consumed for creating and comparing the snapshots. The monitoring-based approach is good for rapid artifact extraction, because the time consumption for snapshots can be reduced. Moreover, it can track the source applications of artifacts for background noise filtering. However, this approach does not respond easily to extension of the source activity model. For all approaches, the artifact refinement process is required for eliminating background noises, as mentioned by James [17]. However, the artifacts that rarely appear cannot simply be omitted, because such an artifact may be a unique sign of a specific activity. In this framework, the artifact extraction for an activity model is performed multiple times. The frequency of an artifact for all repetitions is counted as the appearance probability and will be provided to the investigator. The background noise artifact, which is defined as the artifact that matches multiple activities in the database, is eliminated in the reconstruction phase.

Activity Signature Generation. An artifact is generally composed of timestamps, metadata, and data. For example, a file artifact in a file system is composed of the file data, the file path, the size as metadata, and timestamps of reading, writing, and creation. The activity signature is generated using these elements. For example, the NSRL of the NIST [20] uses hashed file data as the signature, James [17] used the path string, and Hargreaves [18] used various sources to generate a script as the signature.

In this paper, we use a predefined variable table to partially automate the signature generation procedure. First, the investigator defines a table that contains multiple variables such as environment values, specific paths, or user information. Each variable is composed of a tag and a value. The tag is simply the name of the variable. The value is a regular expression that describes the corresponding artifact string, which can be metadata such as the file path or a string extracted from the data. The source of the artifact string is dependent on the type of artifact. Table 1 is an example of a variable table.

Table 1. Example of variable table

Tag	Value
<name>	Investigator12
<userID>	invID12
<keywords>	KeywordA|KeywordB
<%TEMP%>	C:\\Users\\Investigator12\\AppData\\Local\\Temp
<%IE_TEMP%>	C:\\Users\\Investigator12\\AppData\\Local\\Microsoft\\Windows\\Temporary Internet Files\\Content.IE5\\[a-zA-Z0-9]{8}

Algorithm 1. Signature generation algorithm

```
procedure Signature_Generation(Artifact_strings $A[0...n]$, Variables $V[0...m]$)
    while $i$ from 0 to $n$ do
        while $j$ from 0 to $m$ do
            $S[i] \leftarrow$ Replace_Matched($A[i], V[j]$)        ▷ $S$ = list of signatures
        end while
    end while
    return $S$
end procedure
procedure Replace_Matched(Artifact_string $a$, Variable $v$)
    find matching part of $a$, $v$.regex
    $s \leftarrow$ replace matching part of $a$ to $v$.tag        ▷ $s$ = signature
    return $s$
end procedure
```

Using the predefined variable table, each artifact string is compared with a regular expression for every variable. If a matching part of the artifact string is found, then that part is replaced with the tag of the matching variable. The activity signature is the processed artifact string. Algorithm 1 describes the process. Table 2 contains examples of artifact strings and the corresponding signatures that are generated using the variable table described in table 1. The generated

Table 2. Examples of artifact strings and signatures

Artifact string	Signature
C:\Users\Investigator12\AppData\Local\Temp \keywordA\invID12.log	<%TEMP%>\<keywords>\<userID>.log
C:\Users\Investigator12\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZG8IPVA7\siteLogo.gif	<%IE_TEMP%>siteLogo.gif
Computer\HKEY_CURRENT_USER\Software\keywordA\invID12\key	Computer\HKEY_CURRENT_USER\Software\<keywords>\<userID>\key
http://www.keywordA.com/view.php?userid=invID12&mode=sendFile	http://www.<keywords>.com/view.php?userid=<userID>&mode=sendFile

activity signatures are sent to the activity signature database with some information, such as the activity model topic, activity model, and corresponding user inputs.

4.2 Digital Artefact Collection Module

The digital artifact collection module lists artifacts from media acquired at the crime scene. This can be performed on a live system or from a media image. The listed artifacts are generated as signatures by the method, which is exactly the same as the method that was used in the activity signature generation phase before the incident. The extracted artifact signatures are sent to the activity reconstruction module for analysis.

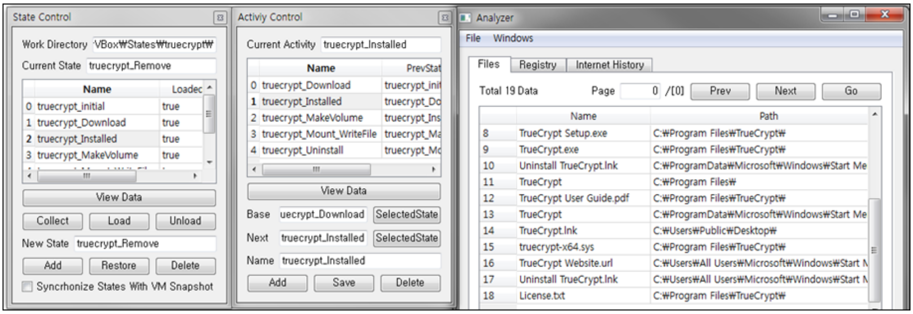
4.3 Activity Reconstruction Module

The activity reconstruction module queries the activity signature database with the artifact signatures extracted from the crime scene. If matching signatures are found, the database sends the corresponding information about the activities. Typically, the number of artifact signatures extracted from the crime scene exceeds one million. Although the reconstruction can be performed automatically, the time consumption is still excessive.

Traditional searching techniques based on time, category, or keyword can be applied to accelerate the reconstruction. Based on the timestamps of the artifacts, the investigator can request the information for an artifact that is used in a specific or recent timeline. To perform the search based on a category or keyword, the investigator submits a specific keyword to the database and receives a list of all related signatures. The acquired list of signatures is searched for the artifact signatures extracted from the crime scene. If a matching signature is found, then the information about the signature will be requested from the activity signature database. The reconstruction time can also be reduced by a frequency-based method. First, the database calculates the list of frequently used artifacts of directory or registry paths, such as the %Program Files% directory or the HCU\Software registry key. After that, the list is sent to the reconstruction module in order of priority. Using these acceleration methods, the extracted artifacts can be automatically abstracted as user activities with a high retrieval rate and high accuracy in a reduced time.

5 Implementation and Evaluation

For proof-of-concept, tools were implemented for each module. Figure 11 is a screenshot of the activity signature generation tool that modeled user activities on the topic TrueCrypt. The tool is based on the model-first approach with activity artifacts extraction based on state comparison. It supports automated signature generation for files, registry, and Internet history. The generated signatures are sent to a signature database.



Work Directory `VBox₩States₩truecrypt₩`

Current State truecrypt_Remove

	Name	Loadec
0	truecrypt_initial	true
1	truecrypt_Download	true
2	truecrypt_Installed	true
3	truecrypt_MakeVolume	true

New State truecrypt_Remove

Current Activity truecrypt_Installed

	Name	PrevStat
0	truecrypt_Download	truecrypt_init
1	truecrypt_Installed	truecrypt_Do
2	truecrypt_MakeVolume	truecrypt_Ins
3	truecrypt_Mount_WriteFile	truecrypt_Ma
4	truecrypt_Uninstall	truecrypt_Mc

Base uecrypt_Download

Next truecrypt_Installed

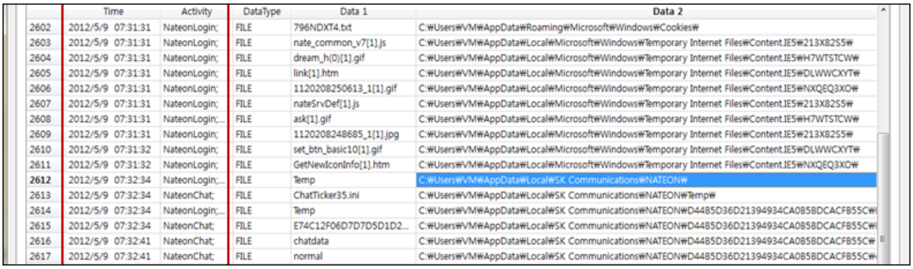
Name truecrypt_Installed

Total 19 Data Page 0 /[0]

	Name	Path
8	TrueCrypt Setup.exe	C:₩Program Files₩TrueCrypt₩
9	TrueCrypt.exe	C:₩Program Files₩TrueCrypt₩
10	Uninstall TrueCrypt.lnk	C:₩ProgramData₩Microsoft₩Windows₩Start Me
11	TrueCrypt	C:₩Program Files₩
12	TrueCrypt User Guide.pdf	C:₩Program Files₩TrueCrypt₩
13	TrueCrypt	C:₩ProgramData₩Microsoft₩Windows₩Start Me
14	TrueCrypt.lnk	C:₩Users₩Public₩Desktop₩
15	truecrypt-x64.sys	C:₩Program Files₩TrueCrypt₩
16	TrueCrypt Website.url	C:₩Users₩All Users₩Microsoft₩Windows₩Start M
17	Uninstall TrueCrypt.lnk	C:₩Users₩All Users₩Microsoft₩Windows₩Start M
18	License.txt	C:₩Program Files₩TrueCrypt₩

Fig. 11. Activity signature generation tool

Figure 12 is a result screen of the activity reconstruction tool after some messenger activities were automatically reconstructed. The tool compares artifact signatures extracted from the crime scene with those stored in the activity signature database.



	Time	Activity	DataType	Data 1	Data 2
2602	2012/5/9 07:31:31	NateonLogin;	FILE	796NDXT4.txt	C:₩Users₩VM₩AppData₩Roaming₩Microsoft₩Windows₩Cookies₩
2603	2012/5/9 07:31:31	NateonLogin;	FILE	nate_common_v7[1].js	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩213X82S5₩
2604	2012/5/9 07:31:31	NateonLogin;	FILE	dream_h(0)[1].gif	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩H7WTSTCW₩
2605	2012/5/9 07:31:31	NateonLogin;	FILE	link[1].htm	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩DLWWCXYT₩
2606	2012/5/9 07:31:31	NateonLogin;	FILE	1120208250613_1[1].gif	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩NXQEQ3XO₩
2607	2012/5/9 07:31:31	NateonLogin;	FILE	nateSrvDef[1].js	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩213X82S5₩
2608	2012/5/9 07:31:31	NateonLogin;...	FILE	ask[1].gif	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩H7WTSTCW₩
2609	2012/5/9 07:31:31	NateonLogin;	FILE	1120208248685_1[1].jpg	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩213X82S5₩
2610	2012/5/9 07:31:32	NateonLogin;	FILE	set_btn_basic10[1].gif	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩DLWWCXYT₩
2611	2012/5/9 07:31:32	NateonLogin;	FILE	GetNewIconInfo[1].htm	C:₩Users₩VM₩AppData₩Local₩Microsoft₩Windows₩Temporary Internet Files₩Content.IE5₩NXQEQ3XO₩
2612	2012/5/9 07:32:34	NateonLogin;...	FILE	Temp	C:₩Users₩VM₩AppData₩Local₩SK Communications₩NATEON₩
2613	2012/5/9 07:32:34	NateonChat;	FILE	ChatTicker35.ini	C:₩Users₩VM₩AppData₩Local₩SK Communications₩NATEON₩Temp₩
2614	2012/5/9 07:32:34	NateonLogin;...	FILE	Temp	C:₩Users₩VM₩AppData₩Local₩SK Communications₩NATEON₩D44B5D36D21394934CA0B5BDCACFB55C₩
2615	2012/5/9 07:32:34	NateonChat;	FILE	E74C12F06D7D7D5D1D2...	C:₩Users₩VM₩AppData₩Local₩SK Communications₩NATEON₩D44B5D36D21394934CA0B5BDCACFB55C₩
2616	2012/5/9 07:32:41	NateonChat;	FILE	chatdata	C:₩Users₩VM₩AppData₩Local₩SK Communications₩NATEON₩D44B5D36D21394934CA0B5BDCACFB55C₩
2617	2012/5/9 07:32:41	NateonChat;	FILE	normal	C:₩Users₩VM₩AppData₩Local₩SK Communications₩NATEON₩D44B5D36D21394934CA0B5BDCACFB55C₩

Fig. 12. Activity reconstruction tool

Using the tools, we designed multiple user activity models, including Google search, Gmail, MSN Messenger, Dropbox, and TrueCrypt as well as some popular Korean applications and Web services. During the signature generation phase, we generated 5000 more file artifact signatures from the activity models. For an evaluation, the signatures were tested on a machine so that all the modeled activities were performed. As a result, 100% of the activity signatures were successfully extracted from 85,000 file artifacts in 19.5 s on average.

6 Conclusion and Future Research Directions

In this paper, we proposed a novel signature-based digital forensic framework that assists investigators to reconstruct user activities automatically. We presented not only the processes in each module of the framework but also techniques for efficient and effective user activity reconstruction. Research on signature-based digital forensic approaches is still in the early stages. We propose the following directions for future research:

- *Definition of criminal user behavior on digital devices.* The research will focus on what kind of user activities should be defined as the user activity model.
- *User activityevent matching algorithm.* Since the collected sequences of user events are varying, it is difficult to define an activity as a sequence of events. Thus, a formalized algorithm for activityevent matching is required.
- *Research on efficient activity signature databases.* The database could involve technologies such as cloud computing and in-memory computing to improve the query speed.
- *Automated and generalized signature generation algorithm.* In this paper, the signature generation still requires investigators to define variables manually. In future work, the generation procedure could be automated using a string pattern recognition approach with improved retrieval rate and accuracy.
- *Fast signature matching algorithm.* Querying all the signatures extracted from a crime scene still requires an excessive amount of time. It is necessary to develop a fast signature algorithm without sacrificing the accuracy and retrieval rate.

Acknowledgements. This research was supported by the R&BD Support Center of Seoul Development Institute and the South Korean government (WR080951) and the Public welfare&Safety research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012M3A2A1051118 2012051118).

References

1. Regional Computer Forensics Laboratory: Annual report for fiscal year 2003-2011 (2011)
2. Garfinkel, S.L.: Digital forensics research: The next 10 years. Digital Investigation 7, S64–S73 (2010)
3. Van Dongen, W.S.: Forensic artefacts left by Windows Live Messenger 8.0. Digital Investigation 4(2), 73–87 (2007)
4. Palmer, G.: A road map for digital forensics research-report from the first Digital Forensics Research Workshop (DFRWS), Utica, New York (2001)
5. Rowlingson, R.: A ten step process for forensic readiness. International Journal of Digital Evidence 2(3), 1–28 (2004)
6. Carrier, B.: Defining digital forensic examination and analysis tools using abstraction layers. International Journal of Digital Evidence 1(4), 1–12 (2003)
7. EnCase forensic, `http://www.guidancesoftware.com/forensic.htm`
8. Forensic toolkit, `http://accessdata.com/products/computer-forensics/ftk`
9. Beebe, N.L., Clark, J.G.: Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results. Digital Investigation 4, 49–54 (2007)
10. log2timeline, `http://log2timeline.net/`

11. Teelink, S., Erbacher, R.F.: Improving the computer forensic analysis process through visualization. Communications of the ACM 49(2), 71–75 (2006)
12. Arnes, A., Haas, P., Vigna, G., Kemmerer, R.: Digital forensic reconstruction and the virtual security testbed ViSe. Detection of Intrusions and Malware & Vulnerability Assessment, 144–163 (2006)
13. Reust, J.: Case study: AOL instant messenger trace evidence. Digital Investigation 3(4), 238–243 (2006)
14. Yasin, M., Cheema, A.R., Kausar, F.: Analysis of Internet Download Manager for collection of digital forensic artefacts. Digital Investigation 7(1), 90–94 (2010)
15. Carvey, H., Altheide, C.: Tracking USB storage: Analysis of windows artifacts generated by USB storage devices. Digital Investigation 2(2), 94–100 (2005)
16. Oh, J., Lee, S., Lee, S.: Advanced evidence collection and analysis of web browser activity. Digital Investigation 8, S62–S70 (2011)
17. James, J.I., Gladyshev, P., Zhu, Y.: Signature Based Detection of User Events for Post-mortem Forensic Analysis. Digital Forensics and Cyber Crime, 96–109 (2011)
18. Hargreaves, C., Patterson, J.: An automated timeline reconstruction approach for digital forensic investigations. Digital Investigation 9, S69–S79 (2012)
19. Hilbert, D.M., Redmiles, D.F.: Extracting usability information from user interface events. ACM Computing Surveys (CSUR) 32(4), 384–421 (2000)
20. National Institute of standards and technology, National software reference library, http://www.nsrl.nist.gov/