

## Keystroke and Mouse Movement Profiling for Data Loss Prevention

JAIN-SHING WU<sup>1,2</sup>, CHIH-TA LIN<sup>1,2</sup>, YUH-JYE LEE<sup>2</sup> AND SONG-KONG CHONG<sup>3</sup>

<sup>1</sup>*CyberTrust Technology Institute*

*Institute for Information Industry*

*Taipei, 105 Taiwan*

<sup>2</sup>*Department of Computer Science and Information Engineering*

*National Taiwan University of Science and Technology*

*Taipei, 106 Taiwan*

<sup>3</sup>*UBIC Taiwan, Inc.*

*Taipei, 104 Taiwan*

*E-mail: {jsw; cheetah}@iii.org.tw; yuh-jye@mail.ntust.edu.tw; alex\_chong@ubictw.com*

Data leakage is a serious problem for many large organizations. In order to provide the user with information about confidential data, many prevalent data leakage prevention (DLP) solutions rely on scanning the content of the relevant files. This approach requires the capability to parse various file formats. However, risks of data breach persist for unsupported file formats. To address this issue, we propose in this paper an active behavior-based DLP model that hooks the keyboard and mouse application programming interfaces (APIs) to track and profile user behavior. This model has two major advantages: (1) it can help discover sensitive data without parsing file formats, and (2) a data creator can be identified according to his/her keystroke and mouse movement behavior. Since this model is based on profiling user behavior, it eliminates the risk of data leakage from unsupported file formats and can identify the creator of a file. The experiments showcase the effectiveness of the proposed model with data creator identification method yields an accuracy rate of 92.64%, which is promising considering that the features of keystroke and mouse movement behavior are dealing together.

**Keywords:** keystroke profiling, data leakage prevention, file parser, data creator identification, sensitive data protection, mouse movement behavior, machine learning

### 1. INTRODUCTION

Data leakage is a serious security issue where sensitive data is disclosed to unauthorized personnel either maliciously or inadvertently. The issue is particularly severe for organizations because a single data leakage incident can result in loss of customer loyalty, unanticipated lawsuits, costs involving the compensation of affected parties, and so on [1, 2]. This issue has become even more significant with the proliferation of mobile devices, widespread use of removable devices, and ubiquitous Internet access [3].

Symantec has reported that more than 232.4 million identities were stolen in 2011 [4]. A data breach investigation report by Verizon revealed that 174 million data records had been compromised through 855 data breach incidents in 2011 [5]. According to statistics released by DataLossDB [6], 1,646 data leakage incidents were reported worldwide in 2012, a far higher number than in the past. As recently as in 2013, 1,459 data leakage incidents occurred. As a consequence, several data loss prevention (DLP) sys-

---

Received December 10, 2013; revised August 20, 2014; accepted October 24, 2014.

Communicated by Chao-Lin Liu, Yung-Jen Hsu, Shou-De Lin, Kuo-Wei Hsu and Ming-Feng Tsai.

tems have been developed to discover, monitor, and protect data through deep content inspection [7]. As DLPs focus on discovering sensitive data within files, they are classified as content-aware DLP systems [7].

Content-aware DLP systems protect data through deep content inspection. To discover sensitive data, these systems first parse the suspicious file and then extract its text streams, which can be temporarily stored in memory or as a file in storage. By verifying text streams containing predefined patterns or keywords, sensitive data can be identified. The file containing sensitive data is then tagged, deleted, quarantined, encrypted, or moved to a safe place for centralized management.

There are many commercial content-aware DLP systems in the market. For instance, the data security suite released by Websense includes three modules – Data Security Gateway, Data Discover, and Data Endpoint – to analyze sensitive data using a variety of techniques [3]. RSA has developed a DLP suite to protect data in data centers, on networks, and at end-points [8]. McAfee’s Total Protection for DLP contains several modules to ensure safe data handling, *e.g.*, DLP Discover, DLP Monitor, DLP Endpoint, *etc.* [9]. The company amXecure has developed a DLP tool called PrivacyID to identify sensitive content in files [10]. Palo Alto Networks has also announced a next generation firewall with DLP functionality to detect critical personally identifiable information (PII), such as social security numbers (SSNs) or credit card numbers [11].

The above content-based DLP model has two main issues. First, to determine whether a file is sensitive or not, it need to understand various file formats so that contents of files can be extracted and examined. Second, it can only identify the data creators (such as author and latest modifier) which recorded in metadata of a file. It cannot provide the information about the data creators of each sentence, because most of the metadata does not record such information. Accordingly, the model cannot determine or adjust the confidentiality of the file according to the identity of data creators. These two issues are described in detail as follows.

### 1.1 The File Parser Issue

State-of-the-art DLP systems use regular expressions, statistical pattern matching, keyword comparison, and document fingerprinting to discover sensitive data [12, 13]. To protect data in storage, *i.e.* Data at Rest (DaR), DLP systems need the capability to understand various file formats so that contents of files can be extracted and examined. The file decoding involved in this is considerably complicated due to various file structure designs (*e.g.*, sequential, inverted, index-sequential, *etc.*) and different character encodings (*e.g.*, Unicode, ASCII, Big5, GB2312, *etc.*). For example, parsing a Microsoft Excel document requires a precise understanding of how the file expresses and separates each data value. However, according to the binary file format for Office published by Microsoft [14], the structure of the Excel 97-2003 file format is described in a document spanning 1,183 pages. It is thus too long and complicated to implement a corresponding parser for DLP systems.

Because file formats can be proprietary (*e.g.*, Microsoft Office documents), open-source (*e.g.*, HTML, Office Open XML, CSS), or even unpublished, the implementation of parsers is challenging and burdensome. Sometimes, files need to be reverse engineered when the file format is unspecified [15]. To protect DaR, current DLP systems

focus on the number of file formats that they can decode. For instance, RSA has announced that its DLP suite can parse over 300 file formats, whereas McAfee has claimed that its DLP solution can parse more than 390 file formats. Solutions developed by Websense can decode more than 400 file formats. Thus, all data security companies aim to cover as many file formats as they can, and thus suffer an ever-increasing burden of implementation.

The rapid growth of modern applications makes the situation worse. Emerging applications may use new file formats to target different domains for purposes of usage, thus creating new challenges for DLP systems in file format decoding. As a result, new unsupported file formats may cause data breaches. To avoid this perennial challenge, it is widely anticipated that a solution exists to determine the content inside files without parsing them.

### 1.2 The Data Creator Issue

As a file may contain contents created by different users, another critical issue in current DLP systems, in addition to file content inspection, is recognizing the identity of the data creators of a file. Certain types of data must be carefully handled, especially data created by senior officials in an organization, *e.g.*, the supervisor, section manager, general manager, *etc.* Such files may contain data, such as strategic organizational policies, that is more sensitive than those created by ordinary employees. Organizations thus need to know the level as well as the means of protection of such data.

The problem of identifying the data creator can be solved by using common plagiarism detection methods [16, 17]. However, such methods focus on identifying the plagiarized content from a given source (*e.g.*, research papers or books) or determining the intentional modification of words or sentence structure without changing the content [16]. These solutions involve file-level cross-reference identification and cannot identify the specific author of material in a file, especially when there are multiple authors of the same file. The ability to identify the data creator of a file has two main advantages: (1) the ability to investigate and assign responsibility, and (2) the capability of refining the sensitivity level of the file according to the identity of its data creators.

Current DLP solutions lack data creator identification ability, *i.e.* they can only recognize the creator of a file through its metadata, but cannot identify an author who contributes data to the file (the data creator). Consequently, the level of confidentiality of a file cannot be determined with high granularity, and thus current DLP solutions fail to prevent unauthorized accesses to critical data [18]. Therefore, the risk of malicious or accidental leakage of data increases.

### 1.3 Our Contribution

In order to solve the above-mentioned issues in data security, we propose in this paper an active behavior-based DLP model. This model is regarded as a compatible system with current state-of-the-art content-based model (*e.g.*, DLP solutions of Websense, RSA, McAfee, *etc.*). It assists the content-based model to track and analyze a user's keystroke behavior while he/she types text in a file and uses this information to refine the actual content that user has entered. To further identify contents generated by different

data creators, it uses keystroke and mouse movement behavior to learn about and profile data creators. With the assistance of the proposed model, current content-based model will benefit from the following advantages:

1. Sensitive content can be discovered without decoding a file, hence eliminating the need to build a new file parser for each new format.
2. Data creators can be identified by analyzing their keystroke as well as mouse movement behavior, due to which the sensitivity level of the file in question can be assessed according to the identities of the creators.
3. The level of confidentiality of a file can be determined immediately after it has been created, thus reducing the time between data breach and incident detection.

The rest of this paper is structured as follows. We present in Section 2 the details of our proposed model and its framework implementation. In Sections 3 and 4, we describe a keystroke and mouse movement behavior learning methods respectively that can be used to identify data creators. Section 5 demonstrates the practical experiments of the proposed model. Section 6 presents a comparison between the proposed model and other content-based DLP models. We offer our conclusions in Section 7.

## 2. THE PROPOSED MODEL

To analyze user's keystroke and mouse movement behavior, the proposed model needs to track the strokes on his keyboard and the movement of his mouse without influencing user's work. Although such tracking can assume various forms – *e.g.*, software, hardware or even external monitoring (such as acoustic analysis or electromagnetic emissions) – the model is implemented as a software agent that resides on user's desktop. An organization can mandate its employees to install this agent and require them to run the software in the background of operating system every time they use a computer. Accordingly, the agent can record and analysis the user's keystroke and mouse movement behavior. To obtain user training data, the organization can also mandate its employees to provide their keystroke and mouse behavior data within a fix period of time. Such data can then be used to create that user's behavior model for identification.

Moreover, if an employee uses intentional delay or speed up to interrupt the keystroke and mouse movement behavior collection, and causes the training data to be useless for data creator identification later, the company still can discover such abnormal behavior and take some internal investigations.

In the flowing, we first provide an overview of our proposed DLP model, and then show how to implement the framework using existing technologies.

### 2.1 Overview

Since a parser will extract a text stream – say streamA – from a file usually generated from user input, it is reasonable to seek to identify the data creator by tracking and analyzing the user's keystroke behavior. This is done so that the newly extracted text stream – say streamB – is identical to streamA with a very high probability. StreamB can

then be verified using any of the aforementioned data matching techniques to determine its sensitivity.

In order to obtain streamB and determine the identity of its creators, the main idea of our proposed DLP model is to create a Secure Keystream Analyzer (SKA). This can provide active file content analysis as follows (also see Fig. 1):

1. When a user (*e.g.*, Bob) starts an application (*e.g.*, Microsoft Office Excel), the SKA will hook the keyboard and Mouse application programming interfaces (APIs).
2. When Bob typing texts to the application, SKA records his keystroke and mouse movement behavior. Once the file (say newfileB) is saved, the SKA will start to analyze the record as follows.
3. As Bob's keystroke (*e.g.*, typingB) may include a number of typing errors and useless keystrokes, the SKA will analyze and eliminate such system keys (*e.g.*, Escape, Menu, Pause/Break, and PrintScreen/SysRq) and function keys (*e.g.*, F1, F2) in order to extract a refined streamB. Moreover, the SKA also uses machine learning method to identify the data creators by verifying the keystroke and mouse movement behavior.
4. Following this, a file named log\_newfileB which contains analyzed result, streamB, and the identity of the data creators, are sent to a DLP system for sensitive data analysis.

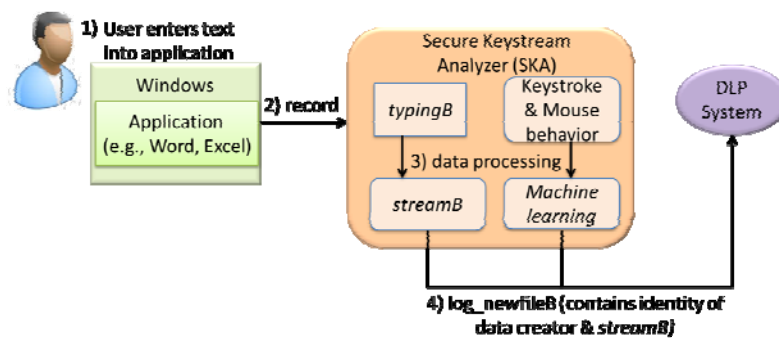


Fig. 1. The concept of the proposed DLP model.

The proposed SKA is equivalent to a text analyzer for files, is independent of existing DLP systems but is compatible with them. Therefore, once a DLP system determines streamB contains sensitive data, the DLP can (i) tag newfileB as either “clean” or “sensitive” according to the receiving path; (ii) classify the sensitive data into different types, such as name, address, SSN, passport number, *etc.*, (iii) refine the sensitivity level of the file according to the identity of its data creators, and (iv) calculate and send the fingerprint (*e.g.*, the MD5 value) of newfileB to a security gateway, which will use the fingerprint to prevent the sensitive file from leaving the organization's network.

## 2.2 Framework Implementation

The following sub-section describes how to implement the SKA and how the SKA can collaborate with a DLP system.

### 2.2.1 Secure keystream analyzer (SKA)

For implementation, the SKA hooks the keyboard APIs to track and profile user behavior. Fig. 2 (a) shows Bob entering some contact information into Microsoft Excel, whereas Fig. 2 (b) shows the raw input data, `typingB`, recorded by the SKA.

To track Bob's mouse movement behavior as he types `typingB`, the SKA also installs a mouse hook to record mouse movement behavior pixel-by-pixel on the screen. Since mouse movement behavior contains valuable patterns that can be used to identify the user [19], the SKA combines such information with the keystroke behavior pattern to identify data creators.

	A	B	C
1	Alex	Alex0131@gmail.com	
2	Bob	Bob307@hotmail.com	
3	Cate	Caty@yahoo.com	
4	Eva	Eva1151@yahoo.com	
5	David	David21@yahoo.com	
6	John	John1101@gmail.com	
7	Mark	MarkLi@hotmail.com	

(a) Bob's input.

```
Alex [TAB] A k e x [BackSpace] [BackSpace] [BackSpace] l e x 0 1 3 1
@g m a i l . c o m [Enter] B o b [TAB] B o b 3 0 7 @ h o t m a i l . c o m
[Enter] C a t e [Space] C a t y [BackSpace] [BackSpace] [BackSpace]
[BackSpace] [BackSpace] [TAB] C a t y @ y a h o o . c o m [Enter] E v a
[Enter] [TAB] E v a 1 1 5 1 @ y a h o o . c o m [Enter] D a v i d [Space]
[BackSpace] [TAB] D a v i d 2 1 @ y a h o o . c o m [Enter] J o h n [TAB] J
o h n 1 1 0 1 @ g m a i l . c o m [Enter] M a r k [TAB] M a r k [BackSpace]
[BackSpace] r k L i @ h o t m a i l . c o m [Enter]
```

(b) The `typingB` recorded by SKA.

Fig. 2. Keystroke tracking.

Once the SKA detects that Bob has saved his typing into a file `newfileB.xls` (through the `CreateFile` and `WriteFile` APIs), `typingB` is analyzed accordingly. Since there are a number of control keys (*e.g.*, `[TAB]`, `[BackSpace]`, `[Space]`, `[Enter]`, *etc.*) in `typingB`, the SKA preserves all alphabetical, numeric, and punctuation keys, and translates `[TAB]`, `[Space]`, and `[Enter]` as a space (between words). It ignores function keys and system keys because these are useless in refining `streamB`. Finally, the SKA counts the number of `[BackSpace]` (or `[Delete]`) strokes following a word to infer the output of Bob's typing, and stores the results of the analysis of `streamB` as a file `log_newfileB` in XML format (see Fig. 3).

It is easy to see from Fig. 3 that `log_newfileB.xml` contains the same result as Bob entered in Microsoft Excel. As a result, a DLP system with an XML parser can discover that `newfileB.xls` contains a number of sensitive data, such as personnel names and email addresses, through the analysis file `log_newfileB.xml`.

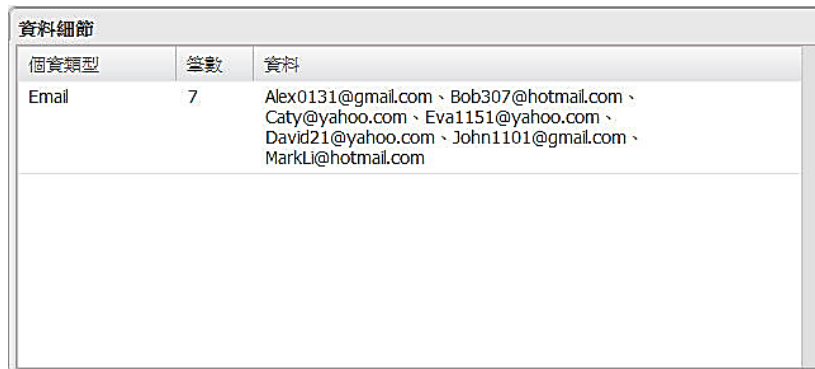
```
Alex Alex0131@gmail.com Bob Bob307@hotmail.com Cate
Caty@yahoo.com Eva Eva1151@yahoo.com David David21@yahoo .
com John John1101@gmail.com Mark MarkLi@hotmail.com
```

Fig. 3. `StreamB` in `log_newfileB.xml`.

### 2.2.2 DLP system

We use the PrivacyID tool developed by amXecure [10] in our proposed model. PrivacyID is an ingenious DLP tool that is implemented as an agent and is deployed in the client's computer. Our model also includes a Central Management Server (CMS). It supports 23 file formats (XML, XLS(X), DOC(X), PPT(X), EPUB, ODS, *etc.*), seven types of sensitive data types (Chinese name, email, credit card number, phone number, *etc.*), and performs well to detect sensitive content in files. Once the agent discovers sensitive content, it sends the content to the CMS. The system administrator can thus keep track of the distribution of sensitive files throughout the company.

To show that the result of the analysis of `log_newfileB.xml` is equivalent to the original `newfileB.xls`, the PrivacyID agent is triggered. The agent performs sensitive content scanning after authenticating the user. When we log into the CMS following the completion of the process, the details of `log_newFileB.xml` can be found. It is easy to see that the email addresses found in `log_newFileB.xml` are the same as those entered by Bob in the original `newfileB.xls` (see Fig. 4).



個資類型	筆數	資料
Email	7	Alex0131@gmail.com、Bob307@hotmail.com、Cathy@yahoo.com、Eva1151@yahoo.com、David21@yahoo.com、John1101@gmail.com、MarkLi@hotmail.com

Fig. 4. The detailed results of CMS.

## 3. KEYSTROKE BEHAVIOR LEARNING

Because keystrokes contain many interesting user typing behaviors, and the typing characteristics of each user (*e.g.*, keyboard typing frequency, typing habits, *etc.*) are different, they can be used to verify user identity [20, 21, 26]. Then, the sensitivity of the file can be fine-tuned automatically with the identity of the data creator, and enhance its security. For instance, if a manager types sensitive data, it is reasonable to assume that the sensitivity of such data is higher than data typed by general staff. Accordingly, a DLP system can monitor such data carefully.

To achieve this goal, the SKA described in the previous section needs to collect the time cost of character typing by user. Then, a machine learning algorithm is used to create the typing model for that user. An outline of the experiment is provided through the following basic steps:

- Recording and Extracting. The following corpus were typed in experiments:

- “*Min-Hwa Law; 0954125789; A239481567; mhlaw@ gmail.com; 4234-4800-5437-2283*”
- “*Mr. Law will arrive in 80 min; his phone number is 0954-234-437; email address is Min-Hwa948@gmail.com. Please contact him.*”
- “*Mr. Wang will arrive tomorrow with flight No. MH480; his phone number is 09371-25283; and the email address is wangming4289@gmail.com. Please contact him.*”

The time logs were collected in terms of character and computer system time. Each character was categorized in 11 types of character location area, and the switching time costs were extracted from the movement between 11 character types in milliseconds.

- **Training and Testing:** Machine learning techniques are applied for classifying users. The goal is to find the classifier and its parameters to achieve optimal accuracy.

### 3.1 Recording and Extracting

Because user behavior for keyboard operation and data input skills might vary per user, the results for the time cost of keying sensitive characters, symbols, and numbers, and for toggling character types are different. By recording the time cost of switching between character types, the typing frequency of each user is obtained and can be used to create the user’s typing model via machine learning algorithms. Subsequently, the model can be used to classify different data creators based on their typing frequency. All possible input characters were categorized into 11 types. Table 1 lists the character type ID and its corresponding characters.

**Table 1. Character type ID and characters.**

Character Type ID	Characters
1	<i>qazwsxedc</i>
2	<i>rfvtgbyhn</i>
3	<i>ujmikolp</i>
4	<i>[]\';,./</i>
5	<i>1234567890</i>
6	<i>QAZWSXEDC</i>
7	<i>RFVTGBYHN</i>
8	<i>UJMIKOLP</i>
9	<i>{ }   : ' " ?</i>
10	<i>! @ # \$ % &amp; * ( ) _ + ' - =</i>
11	<i>space</i>

The switching time cost of character type  $T$  was extracted from the character time cost, where  $T_{i,j}$  is the switching time cost for inputting characters from the  $i$ -th type to the  $j$ -th type. A total of 121 features from the switching time cost were extracted for each instance. In general, 60 characters were collected in an instance. The mean switching time cost was calculated from the same features. A total of 589 instances of ten users were tested and recorded in three experiments; the information for the experiment instances is listed in Table 2.



**Table 2. The information of experiment instances.**

Experiment	Char. No.	Instance No.	No. of Test Times	Test User No.	Total Instance No.
1	73	3	10	10	300
2	119	7	3	9	189
3	149	10	1	10	100

Characters extracted in experiment instance

1: 1~60, 11~70, 21~73

2: 1~60, 11~70, 21~80, 31~90, 41~100, 51~110, 61~119

3: 1~60, 11~70, 21~80, 31~90, 41~100, 51~110, 61~120, 71~130, 81~140, 91~149

**Table 3. A data sample of character time cost for an instance.**

Character time cost
300; 190; 872; 711; 1121; 311; 460; 581; 611; 340; 481; 271; 640; 411; 511; ...
Character encode type
8; 3; 2; 10; 7; 1; 1; 11; 8; 1; 1; 4; 11; 5; 5; 5; 5; 5; 5; 5; 5; 4; ...
Instance features
$T_{8,3}=190; T_{3,2}=872; T_{2,10}=711; T_{10,7}=1121; T_{7,1}=311; T_{1,1}=460; \dots$

Table 3 provides a data sample of character and switching time costs for an instance between different types.

### 3.2 Training and Testing

The character time cost data introduced in the previous section can be applied in various learning algorithms [28]. Support vector machines (SVMs) [22] were originally designed for binary classification. C. Hsu constructed a multi-class classifier by combining several binary classifiers [23]. Teh *et al.* [29] surveyed the research of keystroke dynamics biometrics. In classification method, machine learning is widely used in the pattern recognition domain. SVM generates the smallest possible region that encircles the majority of feature data related to a particular class. SVM maps the input vector into a high-dimensional feature space via the kernel function. As a result, the separating function is able to create more complex boundaries and to better determine which side of feature space a new pattern belongs. SVM is claimed to have a competitive performance as compared to neural network and yet less computational intense [30].

The effectiveness of SVM depends on the kernel selection, the kernel parameters, and the soft margin parameter  $C$ . The Gaussian radial basis function (RBF)  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$  is used for maximal margin hyperplanes. The kernel parameters  $\gamma$  and the cost parameter  $C$  need to estimate the best prediction. The LIBSVM tool [24], which is a method well established for SVMs, was included in our test environment. The tuning of  $\gamma$  and  $C$  are selected by a grid search with exponentially growing sequences.

### 3.3 Experiments

Cross validation was used to identify good parameters so that the classifier could accurately predict unknown data and prevent the overfitting problem [25]. In  $v$ -fold

cross-validation, the training set was divided into  $v$  subsets of equal size. One subset was tested sequentially using the classifier trained on the remaining  $v-1$  subsets. Leave- $p$ -out cross-validation is a way to increase the proportion of test set,  $p$  subsets were used to test.

Our experiment was based on standard method (50% training set and 50% test set). To mitigate over-fitting issue, portion (10%) of training set was used for parameter tuning pretest. The parameter was tuned before using cross-validation to evaluate the test set. 10-fold data subsets were generated randomly. We left 5 fold subsets out for cross-validation testing, 4 fold subsets were used to build SVM classifier, and 1 fold subset was used to pretest for parameter tuning. The ten-fold data subsets were randomly recombined for training, tuning, and testing on a rotation estimation of ten runs. A confusion matrix is a specific table layout that allows performance visualization for a classification system. The confusion matrix table was generated and the accuracy was evaluated for every estimate subset and testing subset.

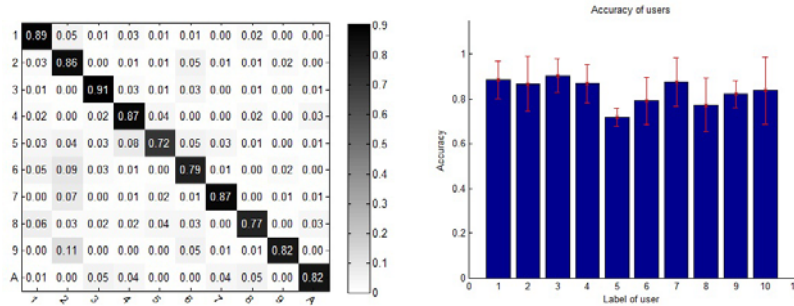


Fig. 5. The average confusion results of ten runs and the average identification accuracy for each user. ( $C = 46.6$ ,  $\gamma = 0.000000146$ ).

Fig. 5 shows the confusion matrix results of the ten-run SVM test and the average identification accuracy for each user. In left figure, the matrix diagonal corresponds to correct classification assignments. Outside of matrix diagonal, each raw value represented the ratio of false negative to other user, and each column value represented the ratio of false positive to the user. Right figure showed the error bar of standard deviations at each user in ten-run test.

As proof of concept, keystroke profiling not only can inspect sensitive content without decoding the file format, but it can also effectively determine the identity of the data creator. It is easy to see that the sensitivity level of such data can be automatically fine-tuned with the corresponding identity significance, and accordingly, enhance the inspection capability of the DLP system.

Because a user might use a mouse as an auxiliary input device when creating data, by integrating mouse movement behavior, the accuracy of the data creator identification can be improved further. Section 4 demonstrates the learning method for mouse movement behavior.

#### 4. MOUSE MOVEMENT BEHAVIOR LEARNING

Because user behavior for mouse operation and movement might vary per user, the

results for the flying time costs and mouse movement acceleration are different. By recording the flying position and time of a mouse movement, the flying time costs of the moving distance with direction and its acceleration profile are obtained for each user, and they can be used to create the user's typing model via machine learning algorithms. Subsequently, the model can be used to classify different data creators based on their typing frequency.

#### 4.1 Recording and Extracting

Mouse movement record datasets were collected from an extended period (e.g., 4 hrs) of regular mouse operation. A movement record dataset was collect as follows:

$$P = \{P_n(x, y, t): x, y, t \in R, n > 0\} \quad (1)$$

where  $x$  and  $y$  are the horizontal and vertical position of the mouse in the screen, and  $t$  is the current universal time value. The recording time interval was 0.08 s. Users might randomly move the mouse, but most actions have an end target. For each movement between consequent points, the movement vector  $\vec{P}_n = P_n(x, y) - P_{n-1}(x, y)$ , the direction angle  $\theta$  of the movement vector can be calculated from Eqs. (2) and (3).

$$\Delta y = P_n(y) - P_{n-1}(y), \Delta x = P_n(x) - P_{n-1}(x) \quad (2)$$

$$\theta = \tan^{-1}(\Delta y / \Delta x) \quad (3)$$

Based on the values of  $\theta$ ,  $\Delta y$ , and  $\Delta x$ , eight movement direction identities are defined in Table 4.

**Table 4. A data sample of character time cost for an instance.**

Move Direction Identity ( $R$ )	$\theta$	$\Delta x, \Delta y$
1	$-\pi/8 \leq \theta < \pi/8$	$\Delta x \geq 0$
2	$\pi/8 \leq \theta < 3\pi/8$	$\Delta x \geq 0$
3	$\theta \geq 3\pi/8 \parallel \theta < -3\pi/8$	$\Delta y \geq 0$
4	$-3\pi/8 \leq \theta < -\pi/8$	$\Delta y \geq 0$
5	$-\pi/8 \leq \theta < \pi/8$	$\Delta x < 0$
6	$\pi/8 \leq \theta < 3\pi/8$	$\Delta x < 0$
7	$\theta \geq 3\pi/8 \parallel \theta < -3\pi/8$	$\Delta y < 0$
8	$-3\pi/8 \leq \theta < -\pi/8$	$\Delta y < 0$

$$\text{The move distance of } \vec{P}_n \text{ is defined as } d_n = \Delta \sqrt{\Delta x^2 + \Delta y^2}. \quad (4)$$

The Effectiveness of the Mouse Movement Segment (**EMMS**) is defined to extract a meaningful move segment.

**EMMS** = a subset of  $P$  that consists of  $k$  consecutive data records ( $P_n$ ) with the same  $R_n$ , and that satisfies the conditions

$$\forall d_n \geq \delta_1 \text{ and } k > \delta_2. \quad (5)$$

where  $\delta_1$  is the threshold for effective mouse move distance, *e.g.*, four pixels, at which to ignore tiny movement, and  $\delta_2$  is the threshold of consecutive data records.

The overall movement distance  $D$  and the flying time cost  $\Delta t$  of *EMMS* can be derived from the start point  $P_{start}$  to the end point  $P_{end}$  of *EMMS*. A value of 10 degrees of distance level was used to represent the range of movement. The maximum degree value is 10, which means that the movement distance is more than 90% of the diagonal distance of the screen resolution.

Moreover, acceleration can be the moving forward force of user behavior. The acceleration information  $a_n$  was used as a further feature to represent the behavior of each *EMMS*,

$$a_n = \frac{v_n - v_{n-1}}{\Delta t_n} = \frac{d_n/\Delta t_n - d_{n-1}/\Delta t_{n-1}}{\Delta t_n}, n > 2 \quad (6)$$

where  $d_n$  is the move distance from recorded point  $P_{n-1}$  to  $P_n$ ,  $\Delta t_n$  is the movement time cost from recorded point  $P_{n-1}$  to  $P_n$ , and  $v_n$  is the moving velocity at  $P_n$ . A three-degree polynomial curve that fits  $A$  was used to smooth  $a_n$  distribution.

$$A = f(a_n), n = 3 \text{ to the data number of } EMMS \quad (7)$$

Further, the features  $A_{max}$ ,  $A_{min}$ , and  $T_{A=0}$  can be decided from  $A$ :  $A_{max}$  is the maximum value of acceleration,  $A_{min}$  is the minimum value of acceleration, and  $T_{A=0}$  is the zero acceleration time point from maximum to minimum. A value interval of 5 deg was used to describe the user acceleration profile. The  $A_{max}$  and  $A_{min}$  degrees are relative to the global maximum and minimum acceleration value of  $A$ , and  $T_{A=0}$  is relative to the overall flying time cost  $\Delta t$  of the *EMMS*.

An instance might consist of multiple *EMMS* (*e.g.*, 100) to adequately describe mouse behavior. The mouse behaviors were collected and combined to the total of 95 features as follows:

- Average flying time cost for  $j$ th direction and  $k$ th degree of distance, where  $j = 1-8$  and  $k = 1-10$ .
- The ratio  $rA_{max}(l)$  for the degree of acceleration  $A_{max}$  profile, where  $rA_{max}(l)$  is the count of  $l$  degree over total *EMMS* count and  $l = 1-5$ .
- The ratio  $rA_{min}(l)$  for the degree of acceleration  $A_{min}$  profile, where  $rA_{min}(l)$  is the count of  $l$  degree over total *EMMS* count and  $l = 1-5$ .
- The ratio  $rT_{A=0}(l)$  for the degree of  $T_{A=0}$  profile, where  $rT_{A=0}(l)$  is the count of  $l$  degree over total *EMMS* count and  $l = 1-5$ .

## 4.2 Experiments

A total of ten users participated in this experiment with an individual computer. A total of 4 hrs of mouse operation behavior were recorded approximately every 0.08s, *i.e.*, 180,000 data points were logged. *EMMS* were extracted using the method described in Section 4.1. The number of extracted *EMMS* for each user is listed in Table 5.

**Table 5. The number of extracted EMMS of each user ID.**

User ID	Number of EMMS
1	1481
2	2264
3	4345
4	4490
5	6281
6	2521
7	2528
8	3855
9	678
10	2375

The *EMMS* dataset was assigned randomly to the 40 instances. The dataset with 95 features of each instance was generated. The methods described in Section 3.2 were used for learning and classification. The Leave-5-Out 10-fold cross-validation method was used for evaluation. A total of 40% of the instances were trained, 10% of the instances were used to tune the parameters, and the remaining 50% of the instances were tested. The overall test accuracy is 80.25% with ten-run experiments.

Fig. 6 shows the confusion matrix results of the ten-run SVM test and the average identification accuracy for each user. The matrix diagonal corresponds to the correct classification assignments, and the total accuracy was calculated through the sum of the matrix diagonal value divided by the testing sample number.

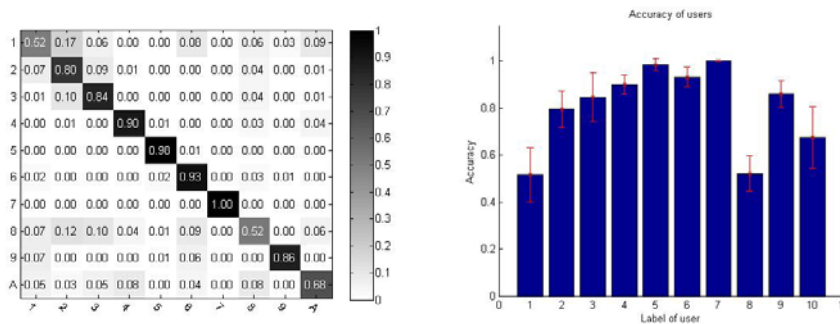


Fig. 6. The average confusion results of ten runs and the average identification accuracy for each user. ( $C = 5$ ,  $\gamma = 0.925$ ).

## 5. PRACTICAL EXPERIMENTS

### 5.1 Free Style Keystroke Experiment

To extend the corpus style in experiments, further essays were introduced in key-stroke experiment: [Sep. 2014, Yahoo.com]

- New phablets from Apple, Samsung could revive mobile market.
- The Microsoft Engine That Nailed The World Cup Is Predicting Every NFL Game.
- The AL East-leading Baltimore Orioles have hit the most home runs in baseball, and it's not even close.
- This Baby's Reaction to Hearing for the First Time Is Guaranteed to Make You Smile.
- Tomorrow – Partly cloudy with afternoon showers or thunderstorms.
- Taiwan's first budget airline said Monday it was scheduled to launch its maiden flight later this month.
- Google Is Working on a Chip That Lets Machines Think Like Humans.
- Hackers break into server for Obamacare website: U.S. officials.
- Ebola Could Reach the U.S. By the End of This Month.
- Photos from the celebrities were stolen individually, the company said.

A total of 100 instances of five users were retested and recorded in twice keystroke experiments. 50% of data was joined into previous training set and rebuild the classifier by the method described in Section 3, and tested for remaining 50% test set. The average accuracy is 84.2% in new test set.

## 5.2 Combination Keystroke and Mouse Movement Profiling Learning

Based on the keystroke feature data set with 610 instances from ten users, the mouse movement *EMMS* data set was assigned randomly to 61 instances and combined to form a new user behavior feature data set. The mouse behavior feature value required normalization to meet the keystroke behavior feature value range, *e.g.*, the mouse behavior feature value had to be multiplied by 1,000.

The methods described in Section 3.2 were used for learning and classification, and the Leave-5-Out *10-fold* cross-validation was used for evaluation. Various methods were used to evaluate our retrieval system. Table 6 lists the results from the users in the first-run SVM test, where *TP* no. = the number of true positives, *FN* no. = the number of false negatives, *FP* no. = the number of false positives, *TN* no. = the number of true negatives, accuracy  $A = (TP + TN) / (TP + FN + FP + TN)$ , precision  $P = TP / (TP + FP)$ , recall  $R = TP / (TP + FN)$ , and *F*-measure  $F = 2PR / (P + R)$ .

**Table 6. Results from various user measures in the first run SVM test.**

Used ID	TP no.	FN no.	FP no.	TN no.	A	P	R	F
1	30	1	5	267	0.98	0.86	0.97	0.91
2	28	2	2	271	0.99	0.93	0.93	0.93
3	30	0	6	267	0.98	0.83	1.00	0.91
4	29	1	1	272	0.99	0.97	0.97	0.97
5	28	2	0	273	0.99	1.00	0.93	0.97
6	28	3	2	270	0.98	0.93	0.90	0.92
7	30	0	0	273	1.00	1.00	1.00	1.00
8	27	3	2	271	0.98	0.93	0.9	0.92
9	29	2	1	271	0.99	0.97	0.94	0.95
10	25	5	0	273	0.98	1.00	0.83	0.91

**Table 7. Further measure results in ten-run SVM test.**

$n^{\text{th}}$ Run	Micro Precision	Micro Recall	Macro Precision	Macro Recall	Macro $F$
1	0.937	0.937	0.942	0.937	0.938
2	0.925	0.925	0.928	0.925	0.923
3	0.931	0.931	0.937	0.932	0.931
4	0.931	0.931	0.937	0.931	0.931
5	0.938	0.938	0.942	0.938	0.938
6	0.895	0.895	0.911	0.895	0.896
7	0.947	0.947	0.950	0.947	0.947
8	0.914	0.914	0.926	0.914	0.914
9	0.905	0.905	0.908	0.905	0.904
10	0.941	0.941	0.944	0.941	0.941
average	0.926	0.926	0.933	0.927	0.926

Table 7 lists the **further measure results** of the ten-run SVM test, where

$$\text{Micro Precision} = \frac{\sum_{i=1}^9 TP_i}{(\sum_{i=1}^9 TP_i + \sum_{i=1}^9 FP_i)},$$

$$\text{Micro Recall} = \frac{\sum_{i=1}^9 TP_i}{(\sum_{i=1}^9 TP_i + \sum_{i=1}^9 FN_i)},$$

$$\text{Micro Specificity} = \frac{\sum_{i=1}^9 TN_i}{(\sum_{i=1}^9 TN_i + \sum_{i=1}^9 FP_i)},$$

$$\text{Micro Negative predictive value} = \frac{\sum_{i=1}^9 TN_i}{(\sum_{i=1}^9 TN_i + \sum_{i=1}^9 FN_i)},$$

$$\text{Macro Precision} = \frac{\sum_{i=1}^9 P_i}{9},$$

$$\text{Macro Precision} = \frac{\sum_{i=1}^9 R_i}{9},$$

$$\text{Macro F} = \frac{\sum_{i=1}^9 F_i}{9}.$$

The average accuracy of the ten-run experiments is **92.64%**. Fig. 7 shows the confusion matrix results of the ten-run SVM test and the average identification accuracy for

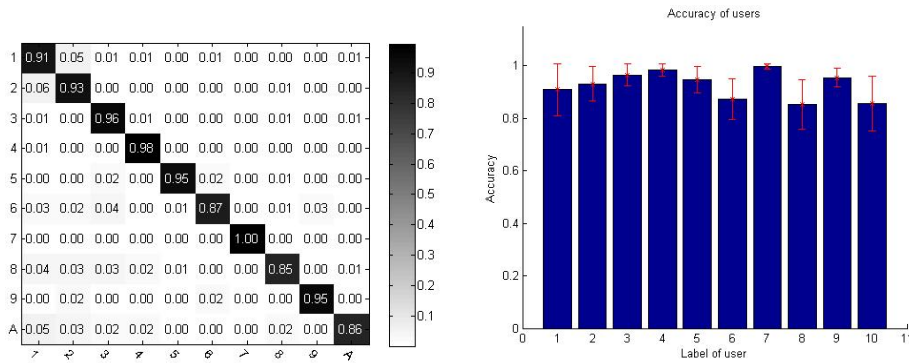


Fig. 7. The average confusion results of ten runs and the average identification accuracy for each user. ( $C = 17$ ,  $\gamma = 0.0000000945$ ).

each user. The matrix diagonal corresponds to the correct classification assignments, and the total accuracy was calculated through the sum of the matrix diagonal value divided by the testing sample number.

Table 8 lists the average accuracy results for all experiments. The results show that the combination of keystroke and mouse movement algorithms provides significant progress in keystroke and mouse movement behavior identification. The proposed approach offers a competitive keystroke and mouse behavior identification solution.

**Table 8. The various average accuracy results for the experiment.**

Exp. ID	Average Accuracy Train Set (%)	Average Accuracy Tuning Set (%)	Average Accuracy Test Set (%)	Feature Data Set Description
1	100	86.6	83.13	Keystroke features
2	100	78.5	80.25	95 mouse movement features
3	100	91.29	92.64	Combination keystroke and mouse movement features

## 6. COMPARISON

Table 9 illustrates the comparison between content-based and behavior-based DLP model. It is easy to see that the proposed behavior-based DLP model is able to obtain text streams within a file without parsing it. The main idea is to record a user keystrokes online and then converts them to text streams. Therefore, it requires more resources to perform the online analysis. On the contrary, the content-based DLP model is required to implement the file format manually. Since such method supports offline processing, it can be used to perform content inspection when computer is idle. As it does not hook the APIs and analysis keystrokes timely, the required computing resources are less than the behavior-based DLP model.

**Table 9. Comparison between content-based and behavior-based DLP model.**

Compared Items	Content-based DLP model	Behavior-based DLP model
File format parsing	Yes, required to implement file parser manually	No, the model is able to obtain text stream without file format parsing
Content inspection	Offline processing, can be applied when computer is idle	Online processing, more computing resources are required
Data creator identify	Can only identify one data creator via metadata	Can identify multiple data creators of a file via machine learning
Confidentiality detection	Determine only when content inspection starts	Timely, determine immediately after a file is created
APIs hook	No	Yes



On the other hand, current content-based DLP model can only determine the data creator via metadata. As metadata does not record every data creator identity inside, the content-based DLP model cannot provide more data beyond it. By using machine learning method to analysis user keystroke and mouse movement behavior, the proposed behavior-based model solves the problem. It can identify multiple data creator of a file without relying metadata. Compare with content-based DLP model, the proposed behavior-based DLP model can determine the confidentiality of a file (via the converted text streams and the identity of file creator) timely after it was created. Therefore, when cooperate with a security gateway, it can prevent the sensitive file from leaving the organization's network.

To summarize, although the behavior-based DLP model requires more computing resource than content-based DLP model, it does not require file parser and is able to identify multiple data creators. Moreover, the behavior-based DLP model can provide a robust data protection because it can determine the confidentiality of a file timely.

## 7. CONCLUSIONS

Based on keystroke profiling, we proposed in this paper an active behavior-based DLP model to eliminate the need for current commercial DLP systems to parse different file formats in order to detect confidential data. Furthermore, our model makes possible the identification of the data creator by recording and analyzing his/her keystroke and mouse movement behavior. This combination approach provides high data visibility, helps determine the identity of the data creators, and is compatible with current DLP systems.

The framework implementation and the experimental results show that the proposed model performs well with prevalent technologies. To the best of our knowledge, this is a novel method to lessen the burden on DLP systems to develop new file format parsers. At the same time, it provides existing DLP systems with the capability to determine the identity of data creators with an accuracy of 92.64%.

There still exist the following issues that need to be resolved: (1) a user behavior may vary dependent on different keyboard and mouse hardware, incremental learning should takes place whenever new behavior instances emerge; (2) the SKA cannot detect sensitive text that is not typed linearly, *e.g.*, text copied from elsewhere or generated from some form through the auto-fill feature [27]. The resolution of these issues requires further research. Nevertheless, we believe that our proposed DLP model and behavior identification are important innovations in the DLP domain.

## REFERENCES

1. A. Shabtai, Y. Elovici, and L. Rokach, *A Survey of Data Leakage Detection and Prevention Solutions*, Springer, Berlin, 2012.
2. V. Stamati-Koromina, C. Ilioudis, R. E. Overill, C. K. Georgiadis, and D. Stamatis, "Insider threats in corporate environments: a case study for data leakage prevention," in *Proceedings of the 5th Balkan Conference in Informatics*, 2012, pp. 271-274.

3. Websense, Inc., "Unified data loss prevention for gateways, endpoints and discovery," <http://www.websense.com/assets/datasheets/datasheet-data-security-suite-en.pdf>, 2013.
4. Symantec, Inc., "Internet security threat report, 2011 trends," <http://www.symantec.com/threatreport/>, 2012.
5. Verizon Communications, "2012 data breach investigations report," <http://www.verizonenterprise.com/DBIR/2012/>, 2012.
6. DataLossDB, Open Security Foundation, "Data loss statistics," <http://datalossdb.org/statistics>, 2013
7. E. Ouellet, "Magic quadrant for content-aware data loss prevention," Technical Report No. G00224160, Gartner, Inc., 2013.
8. RSA, The Security Division of EMC Corporation, "RSA data loss prevention suite," <http://www.rsa.com/products/DLP/sb/9104n DLPSTn SBn 0311.pdf>, 2010.
9. McAfee, "McAfee total protection for data loss prevention," Technical Report, <http://www.mcafee.com/au/resources/solution-briefs/sb-total-protection-for-dlp.pdf>, McAfee, Inc., 2012.
10. amXecure, "PrivacyID," <http://www.amxecure.com/index.php/zh/siem/453-privacyid>, 2013.
11. P. A. Networks, "Preventing data leaks at the firewall," <http://www.paloaltonetworks.com/literature/whitepapers/>, 2008.
12. S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2003, pp. 76-85.
13. H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security and Privacy*, Vol. 8, 2010, pp. 24-31.
14. M. Cooperation, "Office binary file formats (for Word, Excel and PowerPoint)," <http://download.microsoft.com/download/2/4/8/24862317-78F0-4C4B-B355-C7B2C1D997DB/OfficeFileFormatsProtocols.zip>, 2008.
15. Wikipedia, "File format," [http://en.wikipedia.org/wiki/File\\_format](http://en.wikipedia.org/wiki/File_format), 2013.
16. M. S. Pera and Y. K. Ng, "Simpad: A word-similarity sentence-based plagiarism detection tool on web documents," *Web Intelligence and Agent Systems*, Vol. 9, 2011, pp. 27-41.
17. S. Hariharan, "Automatic plagiarism detection using similarity analysis," *The International Arab Journal of Information Technology*, Vol. 9, 2012, pp. 322-326.
18. C. C. Lin, C. C. Chang, and D. Liang, "A new non-intrusive authentication approach for data protection based on mouse dynamics," in *Proceedings of International Symposium on Biometrics and Security Technologies*, 2012, pp. 9-14.
19. N. Zheng, A. Paloski, and H. N. Wang, "An efficient user verification system via mouse movements," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 2011, pp. 139-150.
20. K. Revett, F. Gorunescu, M. Gorunescu, M. Ene, S. T. de Magalhães, and H. M. D. Santos, "A machine learning approach to keystroke dynamics based user authentication," *International Journal of Electronic Security and Digital Forensics*, Vol. 1, 2007, pp. 55-70.
21. Y. Zhao, "Learning user keystroke patterns for authentication," *World Academy of Science, Engineering and Technology*, Vol. 14, 2008, pp. 739-744.

22. C. Cortes and V. Vapnik, "Support-vector network," *Machine Learning*, Vol. 20, 1995, pp. 273-297.
23. C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, Vol. 13, 2002, pp. 415-425.
24. C. C. Chang, "LIBSVM, a library for support vector machines," <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2012.
25. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1986.
26. I. Traore, I. Woungang, S. Obaidat, Y. Nakkabi, and I. Lai, "Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments," in *Proceedings of International Conference on Digital Human*, 2012, pp. 138-145.
27. T. Wuchner and A. Pretschner, "Data loss prevention based on data-driven usage control," in *Proceedings of IEEE 23rd International Symposium on Software Reliability Engineering*, 2012, pp. 151-160.
28. S. P. Banerjee and D. L. Woodard, "Biometric authentication and identification using keystroke dynamics: A survey," *Journal of Pattern Recognition Research*, 2012, pp. 116-139.
29. P. S. Teh, A. B. J. Teoh, and S. Yue, "A survey of keystroke dynamics biometrics," *The Scientific World Journal*, Vol. 2013, Article ID 408280, 2013.
30. E. Yu and S. Cho, "Keystroke dynamics identity verification – its problems and practical solutions," *Computers and Security*, Vol. 23, 2004, pp. 428-440.

### ACKNOWLEDGMENT

This work was partially supported by Ministry of Science and Technology, National Taiwan University and Intel Corporation under Grants MOST102-2911-I-002-001, NSC 102-2221-E-011-056 and NTU103R7501, as well as by the Ministry of Economic Affairs, Taiwan, under Grant 103-EC-17-A-21-0823.



**Jain-Shing Wu (吳建興)** is a Ph.D. student in Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology, Taiwan. He has been working at Institute for Information Industry (III) since 1993. He currently works for CyberTrust Technology Institute at III as division director. He has served as a principal investigator of ICT security technology development programs since 1996. His research focuses on data security, malware behavior analysis, security big data analytics, and vulnerability assessment.



**Chih-Ta Lin (林志達)** is a Ph.D. student in Department of Electrical Engineering at National Taiwan University of Science and Technology, Taiwan. He received his Master degree in Chemical Engineering from Taiwan University in 1989. He currently works for CyberTrust Technology Institute at Institute for Information Industry. He also serves as a principal investigator of security analytics technology development program. His research focuses on malware behavior analysis, security big data analytics, security statistical learning, data mining and information retrieval.



**Yuh-Jye Lee (李育杰)** received the Ph.D. degree in Computer Science from the University of Wisconsin-Madison in 2001. He is an Associate Professor of Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology. He also serves as a Principal Investigator at the Intel-NTU Connected Context Computing Center. His research is primarily rooted in optimization theory and spans a range of areas including network and information security, machine learning, data mining, big data, numerical optimization and operations research.



**Song-Kong Chong (鍾松剛)** received the M.S. in Graduate Institute of Networking and Communication Engineering from Chaoyang University of Technology, Taichung, Taiwan, in 2004. He received the Ph.D. degrees in Computer Science and Information Engineering from National Cheng Kung University in 2011. He is currently working for UBIC Taiwan, Inc. as a Senior Operation Eengineer. His research interests include cryptography, information security, quantum cryptography and digital forensic.