Paper

Uniqueness and Reproducibility of Traffic Signatures

Kazumasa Oida

Department of Computer Science and Engineering, Fukuoka Institute of Technology, Fukuoka, Japan

Abstract—Usable user authentication is an important research topic. The traffic signature-based approach is a new authentication technology that identifies the devices used by online users based on traffic signatures, where the traffic signature is a statistic of the video stream delivered by the authentication server to the user device. This approach has two advantages. First, users need not do any operations regarding the device identification. Second, users need not be sensitive to the privacy loss and computer theft. In this paper, an author evaluates the uniqueness and reproducibility of the signature by introducing a function that quantifies the distance between two signatures. Through number of experiments is demonstrated that the process interference approach has the advantage of generating new signatures that are sufficiently distinguishable from one another.

Keywords—user authentication, traffic signature, HTTP streaming, packet capture, variance plot.

1. Introduction

User authentication is mostly based on passwords. A password-hacking exercise, however, demonstrated that a large number of passwords can easily be cracked [1]. Accordingly, users who place high value on their accounts should adopt a more robust authentication strategy. Since the United States Federal Financial Institutions Examination Council officially recommended the use of multi-factor authentication in 2005 [2], various authentication technologies have been proposed, where multi-factor authentication requires a prover to provide more than one distinct factors to a verifier and there are three distinct authenticating factors: what you have (e.g. house keys), what you know (e.g. passwords), and what you are (e.g. fingerprints) [3], [4].

Current what you have authentication schemes add an additional hardware device to a desktop/laptop PC. Such a device is, for example, a security token, smartphone, or trusted platform module (TPM). Unfortunately, they are not widely used today because they are complex, lead to a loss of privacy, reduce control of the computer, or need to be protected against device theft [3]. Number of authors argue that users' capabilities and understanding should be factored into the design of security technologies [5], [6].

In [7] is proposed another what you have authentication scheme, which identifies the machines users are operating to access their accounts. Presented in [7] approach is based on video traffic analysis. The authentication server delivers a video stream and the user device records packet arrival times to calculate a traffic signature. The server then verifies whether the obtained signature agrees with the one obtained before or the one registered previously. Recently, some online banks request users to show their countersigns, when users try to sign-in using devices that are different from those they used to use. The difference can be detected using the client environment carried by the HTTP protocol, which includes an IP address, a browser type, etc. This scheme roughly distinguishes user computer platforms, whereas author's approach, shown in [7], precisely distinguishes them based on their unique signatures. As long as user machines are correctly identified, users do not need to be aware of anything about the machine identification since signatures are calculated and verified without intervention from users.

Contrarily, the current three major what-you-have authentication technologies, which deliver codes via the security token, email (or SMS), or an app running on a portable device (e.g. smartphone) [8], direct users to do some operations, such as starting the device/app and typing in the code. In addition, as opposed to previous authentication schemes, in which a single device/app generates codes based on some algorithm, the traffic signature is formed through the interactions among numbers of elements, which include not only hardware and software components of the user platform but also the server, video coding techniques, communication protocols that affect statistics of video traffic [9]. Since the interactions are not simple, it is difficult to infer the signature even if detail specifications of the user and server machines are given.

Meanwhile, a TPM chip into which unique RSA keys have been burnt can strictly identify the user machine [10]. This PKI-based approach strongly connects a user device to its owner, so that owners must pay careful attention to a privacy loss and unit theft. Some mechanisms that minimize the risk when stolen are a priori integrated (e.g. platform integrity). Unfortunately, these extra attention and mechanisms may cause the usability problems. Contrarily, in author's approach, users do not have to be sensitive to these risks since the signatures appear only in the authentication process and there is no personal identity-related information on the user device.

The PKI-based approach is also costly. As discussed in [5], authentication solutions must be accessible to all online users not just in terms of knowledge and effort but also in terms of cost. The authors in [5] quote that older users, who have much to gain from online participation, might be unable or unwilling to own a smartphone, which is the second

factor of choice in many authentication solutions currently deployed or planned. Presented approach demands video delivery. The HTTP-based streaming technologies are used because they are inexpensive and widely used today [11]. They avoid NAT and firewall traversal issues and provide cost effectiveness since there is not need dedicated streaming servers for video delivery.

The previous author's work [7] introduced the traffic signature and discussed its sensitivity to the user machine. This paper evaluates the signature through numbers of experiments and clarifies its uniqueness and reproducibility. Section 2 introduces the traffic signature and briefly outlines the result in [7]. Section 3 defines the distance between two signatures, based on which uniqueness and reproducibility are discussed in Section 4. Section 5 investigates which components of the user machine or interactions among them dominantly participate in forming the signature. The findings in this section are effective not only in enhancing reproducibility, but also in allowing users to have different signatures even if their machines consist of the same hardware components. Finally, Section 6 presents the conclusions.

2. Traffic Signature

First decay rate is defined, which is derived from the variance plot [12]. Decay rate $\beta(m)$ indicates how fast traffic variability declines at time scale m. It depends on various factors (e.g., computer hardware and software implementation, protocols, propagation delays, and bandwidth) and the dominant factors vary with m. The following describes how to calculate $\beta(m)$. Let X_k denote the number of arriving packets during the k-th time interval of length δ, where $\delta = 10^{-5}$ s. The m aggregated series $\{X_k^{(m)}\}$ are obtained by dividing time series $\{X_k\}$ into blocks of length m and averaging the series over each block as

$$X_\ell^{(m)} = \frac{1}{m} \sum_{i=\ell m-m+1}^{\ell m} X_i, \ \ell = 1,2,\ldots,\lfloor N/m \rfloor, \qquad (1)$$

where m is a positive integer, N is the size of series $\{X_k\}$, and $\lfloor x \rfloor$ is the largest integer that does not exceed x. The sample variance of $\{X_k^{(m)}\}$ is given by

$$V^{(m)} = \frac{1}{\lfloor N/m \rfloor - 1} \sum_{k=1}^{\lfloor N/m \rfloor} (X_k^{(m)} - \bar{X})^2, \qquad (2)$$

where $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$. Hereafter it is assumed that aggregation levels m_i, $i = 0,1,\ldots$, take real numbers. The decay rate at level m_i is defined as

$$\beta(m_i) = \log\left(V^{(\lfloor m_{i+1} \rfloor)} / V^{(\lfloor m_i \rfloor)}\right), \qquad (3)$$

where $m_0 = 1$ and $m_{i+1} > m_i$. Throughout the paper, $\log \frac{m_{i+1}}{m_i} = \log \frac{N}{50} \cdot \frac{1}{21}$ for all i, and $N = 6 \cdot 10^6$. The traffic signature is twenty decay rates $\beta_1, \beta_2, \ldots, \beta_{20}$, where β_i is used to indicate $\beta(m_i)$ for simplicity.

2.1. Experimental System

Unless otherwise mentioned, all results in this paper are obtained using the experimental system in Fig. 1. In the figure, the client PC (C-PC) accesses France 24 live (IP = 213.205.104.131), a news channel based in France, using the Internet Explorer Flash Player add-on. This online news is delivered at a constant rate (448 Kb/s) using the TCP protocol. In Fig. 1, all packets destined for C-PC are copied to the attacker's PC (A-PC) by the port-mirroring hub, so that not only C-PC but also A-PC collects video packets from the server with WinDump [13] to calculate signatures. Hereinafter, signatures calculated on C-PC (resp. A-PC) are referred to as user (resp. attacker) signatures. The news channel France 24 live was used because there are 30 routers between the news server and C-PC and the round-trip time is approximately 283 ms. Such a long-distance communication generates highly variable video traffic. Practical signatures should be stable in this case.

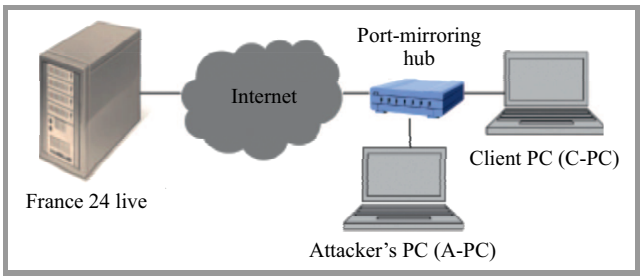


Fig. 1. The port-mirroring hub copies all packets destined for C-PC to A-PC.

2.2. Previous Results

In this subsection the results presented in [7] are briefly introduced. Four machines in Table 1 are used as C-PC in Fig. 1. Although they are all Windows machines, their software and hardware components are somewhat different. Figures 2 and 3 show their attacker and user signatures, respectively. Throughout the paper, ten samples are obtained for each signature $\{\beta_i\}_{1\le i\le 20}$ to see the stability of each decay rate β_i. From the figures, decay rates $\{\beta_i\}_{1\le i\le 16}$ are mostly stable, while $\{\beta_i\}_{17\le i\le 20}$ are not. This is mainly because the number of samples $X_k^{(\lfloor m_i \rfloor)}$ decreases as an increase in i. It can be seen that attacker signatures (Fig. 2) are all similar. In contrast, distinct differences exist between any two user signatures in Fig. 3.

Table 1
Four Windows machines *a-d* used in the experiment

C-PC	Model	Purchase	OS
a	XPS420	Jun. 2008	Vista, 32 bits
b	XPS435T	Jul. 2010	Windows 7, 64 bits
c	XPS9100	Jul. 2011	Windows 7, 64 bits
d	Inspiron	Jan. 2012	Windows 7, 64 bits

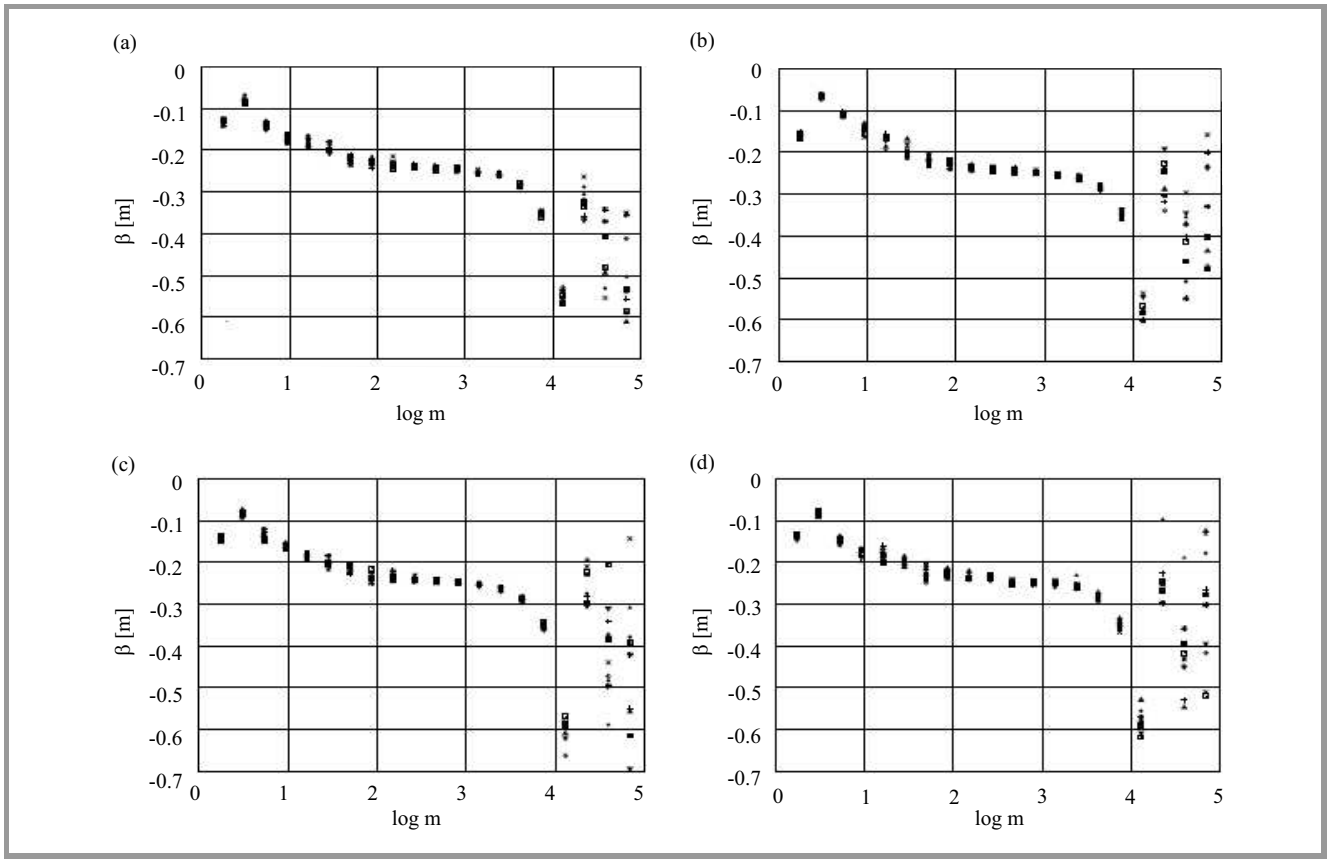


Fig. 2. Attacker signatures (measured on A-PC): (a), (b), (c), and (d) correspond to machines *a*, *b*, *c*, and *d* from Table 1, respectively.

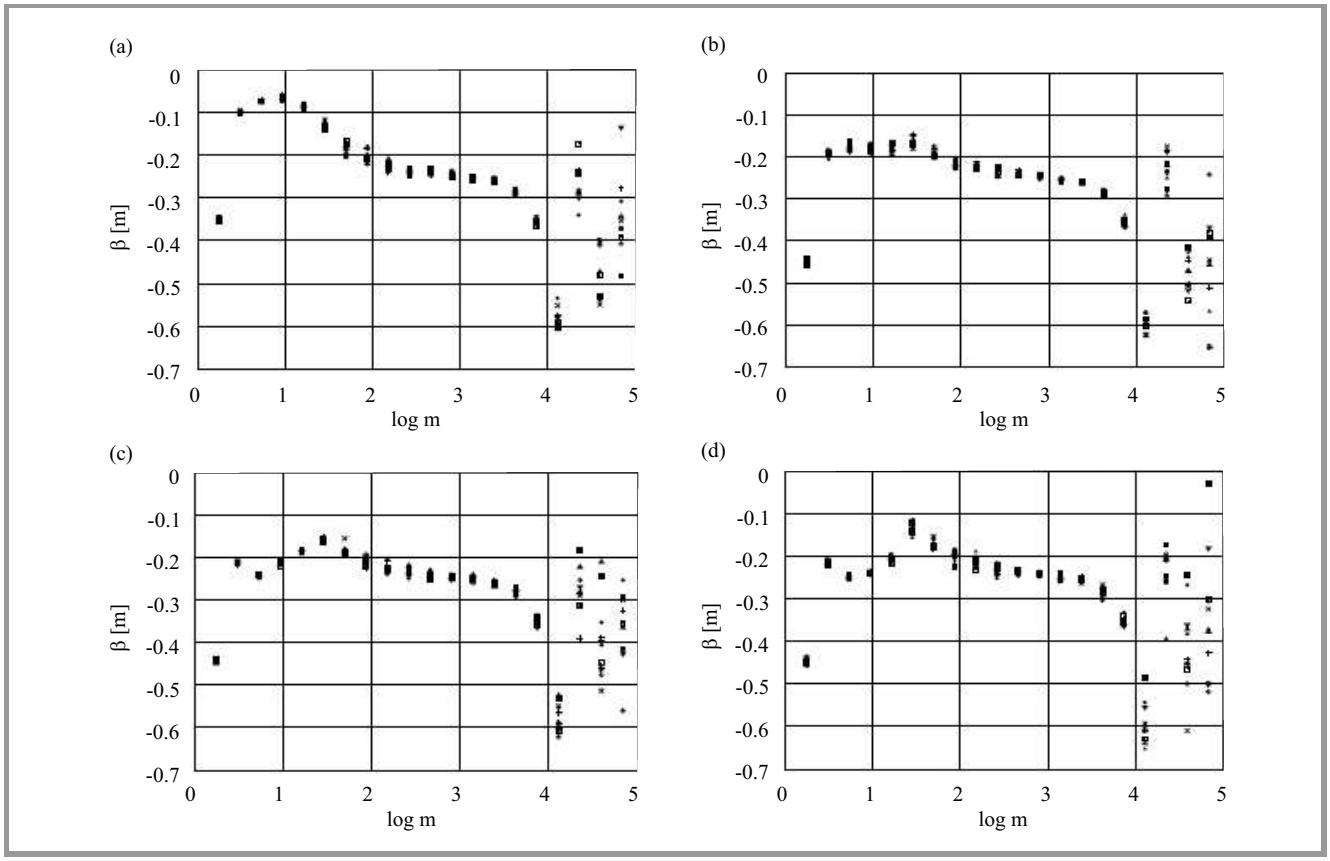


Fig. 3. User signatures (measured on C-PC): (a), (b), (c), and (d) correspond to machines *a*, *b*, *c*, and *d* from Table 1, respectively.

Next the difference between user and attacker signatures is shown. From Figs. 2 and 3, user signatures are different from attacker signatures typically at levels m satisfying $\log(m) < 2$. Note that $m = 10^2$ corresponds to the time scale of one millisecond since $10^2\delta = 10^{-3}$ s. The port-mirroring hub never affects variances at this large time scale. The following is author's explanation for this phenomenon. The difference occurs because A-PC performs only packet collection, while C-PC performs both packet collection and packet processing. On C-PC, the two jobs are executed in parallel on every packet arrival. The two jobs interfere with each other and this interference makes the execution time to obtain the current time fluctuate. In brief, the difference is due to inaccurate packet arrival timestamps caused by resource (memory, CPU, etc.) competition between two jobs. Hence, signatures in Fig. 3 differ only at small time scales. Furthermore, the TCP protocol intensifies the competition because packets tend to arrive in batches when the protocol is used. Since the interference is influenced by various factors (e.g. I/O controllers, device drivers, and job scheduling), it is conjectured that different machine models in Table 1 generated different signatures.

3. Signature Distance

Previous work does not quantify the difference between two signatures [7]. This section first defines the distance between them. Let $\{\beta_i^a\}$ and $\{\beta_i^b\}$ denote user signatures of machines a and b, respectively. They are distinguishable if there exists at least one integer i at which $|\beta_i^a - \beta_i^b|$ is sufficiently large. Therefore, the research is focused on the distance between the i-th decay rates β_i^a and β_i^b.

Figure 4 shows the histogram of 100 β_1 samples and the normal Q-Q plot, which compares the 100 β_1 samples with the theoretical normal distribution. From the figure, β_1 has a distribution close to the normal distribution. The normality tests were performed using 100 β_i samples. If $1 \le i \le 14$, both the Shapiro-Wilk and Anderson-Darling tests do not reject the null hypothesis stating that the β_i samples are normally distributed when the significance level α is 0.01. Therefore, this paper uses only $\{\beta_i\}_{1\le i\le 14}$ for authentication and assumes that β_i^a and β_i^b for $1 \le i \le 14$ are independent and each has a normal distribution.

Let μ_i^a and σ_i^a be the sample mean and standard deviation obtained from ten β_i^a samples. If $\mu_i^a > \mu_i^b$, the distance between β_i^a and β_i^b must be a decrease function of $\Pr(\beta_i^a < \beta_i^b)$, the probability that a sample of β_i^a is smaller than that of β_i^b. Let $F(x;\mu,\sigma^2)$ be the cumulative distribution function (CDF) of normal distribution $N(\mu,\sigma^2)$. If $\mu_i^a > \mu_i^b$, the probability $\Pr(\beta_i^a < \beta_i^b)$ is given by

$$\Pr(\beta_i^a < \beta_i^b) = F(0; |\mu_i^a - \mu_i^b|, (\sigma_i^a)^2 + (\sigma_i^b)^2). \quad (4)$$

Meanwhile, if $\mu_i^a < \mu_i^b$, $\Pr(\beta_i^a > \beta_i^b)$ is equal to the right hand side of Eq. (4). Therefore, independent of whether

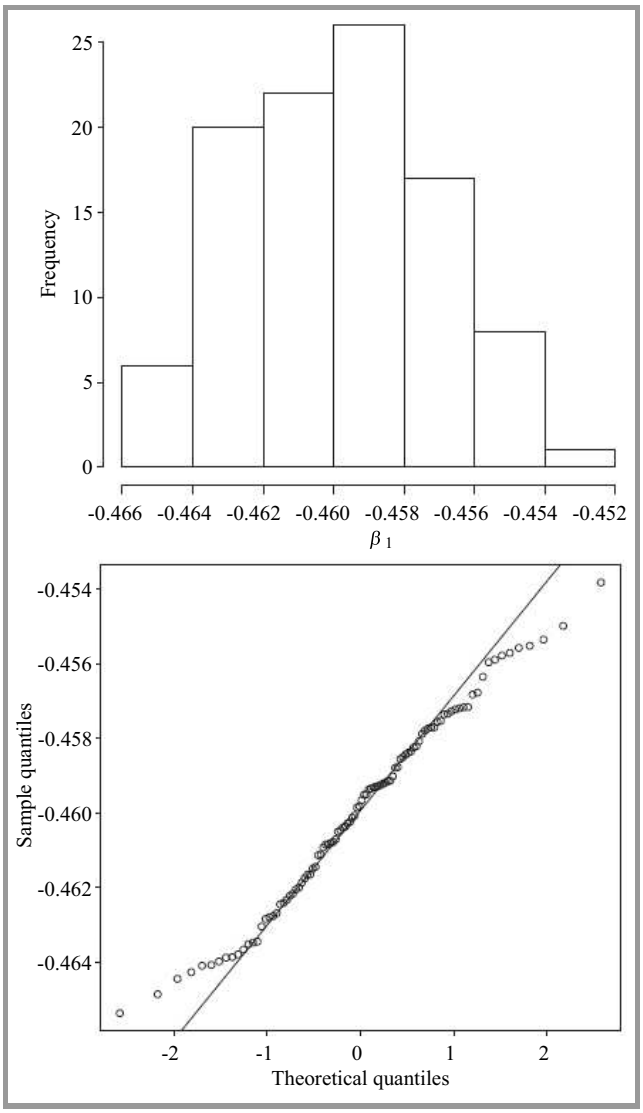


Fig. 4. The histogram and normal Q-Q plot obtained from 100 β_1 samples.

$\mu_i^a < \mu_i^b$ or not, $d_i(a,b)$, the distance between decay rates β_i^a and β_i^b, is defined as

$$d_i(a,b) = -\log F(0; |\mu_i^a - \mu_i^b|, (\sigma_i^a)^2 + (\sigma_i^b)^2). \quad (5)$$

Note that $d_i(a,b)$ is a decrease function of $\Pr(\beta_i^a < \beta_i^b)$ if $\mu_i^a > \mu_i^b$.

Also the $D(a,b)$, the distance between two signatures $\{\beta_i^a\}$ and $\{\beta_i^b\}$, is defined as

$$D(a,b) = |\{i \in \{1,2,\ldots,14\} | d_i(a,b) \ge L_i\}|. \quad (6)$$

Briefly, $D(a,b)$ is the number of integers i that satisfy $d_i(a,b) \ge L_i$, where threshold L_i, which is obtained later, determines whether the i-th decay rates of two signatures are the same or not. Two distance functions d and D have the following features: $d_i(a,b) = d_i(b,a)$, $D(a,b) = D(b,a)$, $d_i(a,b) \ge d_i(a,a) = -\log 0.5 (\approx 0.3)$, $D(a,b) \ge D(a,a) = 0$, and they do not satisfy the triangle inequality.

3.1. Signature Verification

Algorithm 1 shows the signature verification procedure A_s. In the algorithm, $\{\tilde{\beta}_i^b\}$ indicates the most recently obtained sample signature of machine b. Let G be the set of all user machines that request signature verification. Given $a(\in G)$ and $\{\beta_i\}$, procedure A_s returns "accept" if $\{\beta_i\}$ is considered as a signature of machine a; otherwise, it returns "reject". If accepted, $\{\tilde{\beta}_i^a\} = \{\beta_i\}$.

Algorithm 1: Signature verification procedure A_s.

```
    Require: For any b ∈ G_a, D(a,b) ≥ 1.
 1: procedure A_s(a, {β_i})
 2:     while G_a is not empty do
 3:         Select b ∈ G_a
 4:         G_a ← G_a \ {b}
 5:         for i = 1 to 14 do
 6:             if d_i(a,b) ≥ L_i then
 7:                 if (μ_i^a − μ_i^b)(β_i − β̃_i^b) ≤ 0 then
 8:                     return reject
 9:                 end if
10:             end if
11:         end for
12:     end while
13:     return accept
14: end procedure
```

The verification is performed by comparing the signature $\{\beta_i\}$ with signatures of other machines. Let G_a be the set of machines that are used for verifying a signature of machine a. The procedure works under the condition that for any $b \in G_a$, $\{\beta_i^a\}$ and $\{\beta_i^b\}$ are distinguishable, i.e.

$$D(a,b) \geq 1, \text{ for any } b \in G_a. \tag{7}$$

In the 7th line of Algorithm 1, an inequality

$$(\mu_i^a - \mu_i^b)(\beta_i - \tilde{\beta}_i^b) \leq 0 \tag{8}$$

implies that the magnitude relation between μ_i^a and μ_i^b is different from that between β_i and $\tilde{\beta}_i^b$. Note that P_{error}, the probability that inequality (8) holds, is

$$P_{error} = 10^{-d_i(a,b)}. \tag{9}$$

This seldom occurs if $d_i(a,b) \geq L_i$ (in the 6th line of Algorithm 1) holds for a large L_i. Thus, the procedure considers that $\{\beta_i\}$ is not a signature of machine a and returns reject. Procedure A_s returns accept after it makes t comparisons, where $t = \sum_{b \in G_a} D(a,b)$. Therefore, P_{forge}, the probability that an attacker successfully forges a signature that is accepted by the procedure, is

$$P_{forge} = 2^{-t} \tag{10}$$

if the attacker has no information about the signature of machine a. The forgery probability exponentially decreases with t.

Algorithm 1 needs statistics μ_i^a, μ_i^b, σ_i^a, and σ_i^b for calculating $D(a,b)$ and $d_i(a,b)$. It is recommended that μ_i^a, σ_i^a, and G_a should be updated by using the latest samples because these statistics may be affected by various software updates (e.g. Windows update).

3.2. Requirements

The traffic signature-based authentication has the same targets of challenge as biometric-based authentication, where biometric information (e.g., fingerprint, iris, etc.) is required to hold three requirements [14]:

- R1 – it is sufficiently different between any two users,
- R2 – it is reproducibly captured repeatedly,
- R3 – it is hard to be faked.

This paper focuses on R1 and R2. Before verifying whether traffic signatures satisfy R1 and R2, there is need to determine the values of $\{L_i\}$, which are criteria for determining whether the i-th decay rates of two signatures are different or not. This paper decomposes L_i into two parts as

$$L_i = L_s + \Delta L_i, \tag{11}$$

where $L_s(>0)$ is the distance required by security strength and $\Delta L_i(>0)$ denotes the fluctuation range of distance d_i caused by changes in CPU and network loads, etc. Requirement R1 demands that an integer i that satisfies $d_i(a,b) \geq L_i$ for any a and b should exist, and R2 insists that ΔL_i should be sufficiently small. In the next section, ΔL_i values are experimentally derived.

First the value of L_s such that the authentication system satisfies the following capability is determined. A user accesses the system every one minute and A_s returns reject once in a year on average. Assume that $\Delta L_i = 0$ and that for all $b \in G_a$ and for all i satisfying $d_i(a,b) \geq L_s$, $d_i(a,b) = L_s$. Then, the mean of the binomial distribution indicates that

$$60 \cdot 24 \cdot 365 \cdot \left(1 - (1 - 10^{-L_s})^t\right) = 1, \tag{12}$$

where $t = \sum_{b \in G_a} D(a,b)$. From Eqs. (9) and (10), both L_s and t should be enlarged as much as possible for minimizing both P_{error} and P_{forge}. From Eq. (12), the L_s and t at the same time can't be reduced. If $t = 20$, from Eq. (12), $L_s \approx 7$. In this case, from Eqs. (9) and (10), $P_{error} \approx 10^{-7}$ and $P_{forge} \approx 10^{-6}$ (i.e. the security strength corresponds to a six-digit code). Hereinafter, $L_s = 7$ is used.

4. Signature Analysis

4.1. Four Machines

This section discusses whether traffic signatures satisfy inequality (7) by using machine set $G = \{a,b,c,d\}$, which consists of four machines listed in Table 1, under the condition that fluctuation $\Delta L_i = 0$. Figure 5 exhibits distances d_i derived from user signatures of machines in G. It can be seen from Figs. 3 and 5 that d_i, $1 \leq i \leq 14$, correctly quantify signature differences. Let us see the distance between

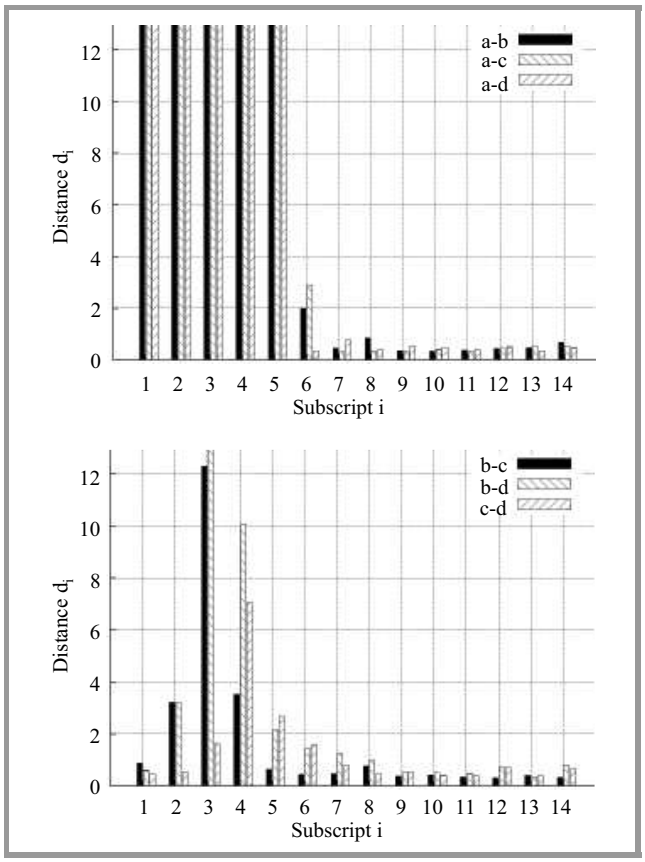


Fig. 5. Distances d_i for all machine pairs in Table 1 – "*a-b*" indicates $d_i(a,b)$.

signatures of machines a and b ("*a-b*" in Fig. 5). From the figure, $d_i(a,b) > L_i(=7)$ at $i = 1,2,3,4,5$. Therefore, $D(a,b) = 5$. Similarly, $D(a,c) = D(a,d) = 5$, $D(b,d) = 2$, and $D(b,c) = D(c,d) = 1$. Accordingly, (7) holds for all sets G_x, $x = a,b,c,d$ when $G_x = G \setminus \{x\}$. Figure 5 also shows that for any $x,y \neq a$, $D(a,y) \geq D(x,y)$. Namely, machine a creates the most characteristic signature. This may be because from Table 1, machine a is the oldest PC (therefore, it may be composed of many unique devices) and its Windows version is different from those of the others.

4.2. Fluctuation Range

Next the fluctuation range ΔL_i discussed in Subsection 3.2 is estimated. Let Δd_i be the decrease in the distance $d_i(x,y)$ caused by the increase in the load of machine x, i.e.

$$\Delta d_i = d_i(x,y) - d_i(\tilde{x},y), \tag{13}$$

where $\tilde{x}$ denotes machine x whose load has been raised. The author estimates the distribution of Δd_i through experiments with various machines x, and then determine ΔL_i such that Δd_i is not greater than ΔL_i with probability 0.95 ($\Pr(\Delta d_i \leq \Delta L_i) = 0.95$).

From Eq. (13), mean μ_i^y and variance $(\sigma_i^y)^2$ of machine y are necessary to obtain Δd_i. These values are independent of β_i^x and $\beta_i^{\tilde{x}}$. However, y should be as normal as possible. Therefore, one can assume that variance $(\sigma_i^y)^2$ is equal to the mean of variances $(\sigma_i^x)^2$ of various machines x, i.e.,

$$(\sigma_i^y)^2 = m\left((\sigma_i^x)^2\right), \tag{14}$$

where this paper uses $m(Z_i)$ and $s(Z_i)$ to indicate the mean and standard deviation of samples $\{Z_i\}$. On the other hand, μ_i^y can be determined by assuming that due to the load increase, the distance decreases to L_s, i.e.

$$d_i(\tilde{x},y) = L_s. \tag{15}$$

From Eqs. (13) and (15), we have

$$d_i(x,y) = \Delta d_i + L_s. \tag{16}$$

In short, the load increase lowers the distance from $L_s + \Delta d_i$ to L_s. Using Eqs. (5) and Eqs. (14), (15) and (16) are rewritten as

$$-\log F\left(0; |\mu_i^{\tilde{x}} - \mu_i^y|, (\sigma_i^{\tilde{x}})^2 + m(\sigma_i^x)^2\right) = L_s \tag{17}$$

$$-\log F\left(0; |\mu_i^x - \mu_i^y|, (\sigma_i^x)^2 + m(\sigma_i^x)^2\right) = \Delta d_i + L_s. \tag{18}$$

Given $\mu_i^{\tilde{x}}$ and $(\sigma_i^{\tilde{x}})^2 + m(\sigma_i^x)^2$, (17) has two solutions μ_i^y, so that (17) and (18) yield two Δd_i values for each x. The 24 Δd_i values are obtained using various desktop and laptop PCs x. For measuring signatures of $\tilde{x}$, the CPU, memory, and hard disk utilization rates are raised by playing a video (whose bitrate is 2.4 Mb/s) stored on the hard disk. The Shapiro-Wilk and Anderson-Darling tests do not reject the normality of 24 Δd_i samples for all $i = 1,2,\ldots,14$. Thus, this paper assumes that Δd_i has a normal distribution. Using mean $m(\Delta d_i)$ and standard deviation $s(\Delta d_i)$, ΔL_i is given by

$$\Delta L_i = m(\Delta d_i) + z_{0.95} s(\Delta d_i), \tag{19}$$

where $z_{0.95}$ satisfies $F(z_{0.95};0,1) = 0.95$; i.e., $\Pr(\Delta d_i \leq \Delta L_i) = 0.95$. Table 2 shows $m(\Delta d_i)$, $s(\Delta d_i)$, and ΔL_i derived from Δd_i samples. From the table, fluctuation ranges ΔL_i depend on i and are between 2.5 and 5. If ΔL_i values in Table 2 are used for calculating L_i in Eq. (11), we have $D(x,y) \geq 1$ for all $x,y \in G$ in Subsection 4.1 except for $D(c,d)$. Therefore, in equality (7) does not hold. In other words, the four machine models do not satisfy requirement R1 in Subsection 3.2. In Section 5, author improves signature uniqueness to fulfill the requirement.

Table 2
Means $m(\Delta d_i)$, standard deviations $s(\Delta d_i)$, and fluctuation ranges ΔL_i for $i = 1,2,\ldots,14$

i	1	2	3	4	5	6	7
$m(\Delta d_i)$	–0.1	–0.2	0.3	–0.1	0.3	0.2	0.1
$s(\Delta d_i)$	2.6	1.9	2.3	1.9	1.9	2.1	2.2
ΔL_i	4.2	2.9	4.1	2.9	3.3	3.7	3.7

i	8	9	10	11	12	13	14
$m(\Delta d_i)$	0.3	–0.1	0.0	0.9	1.3	–0.4	0.0
$s(\Delta d_i)$	2.4	1.9	2.1	2.5	2.3	1.7	2.3
ΔL_i	4.2	3.0	3.4	5.0	5.0	2.5	3.8

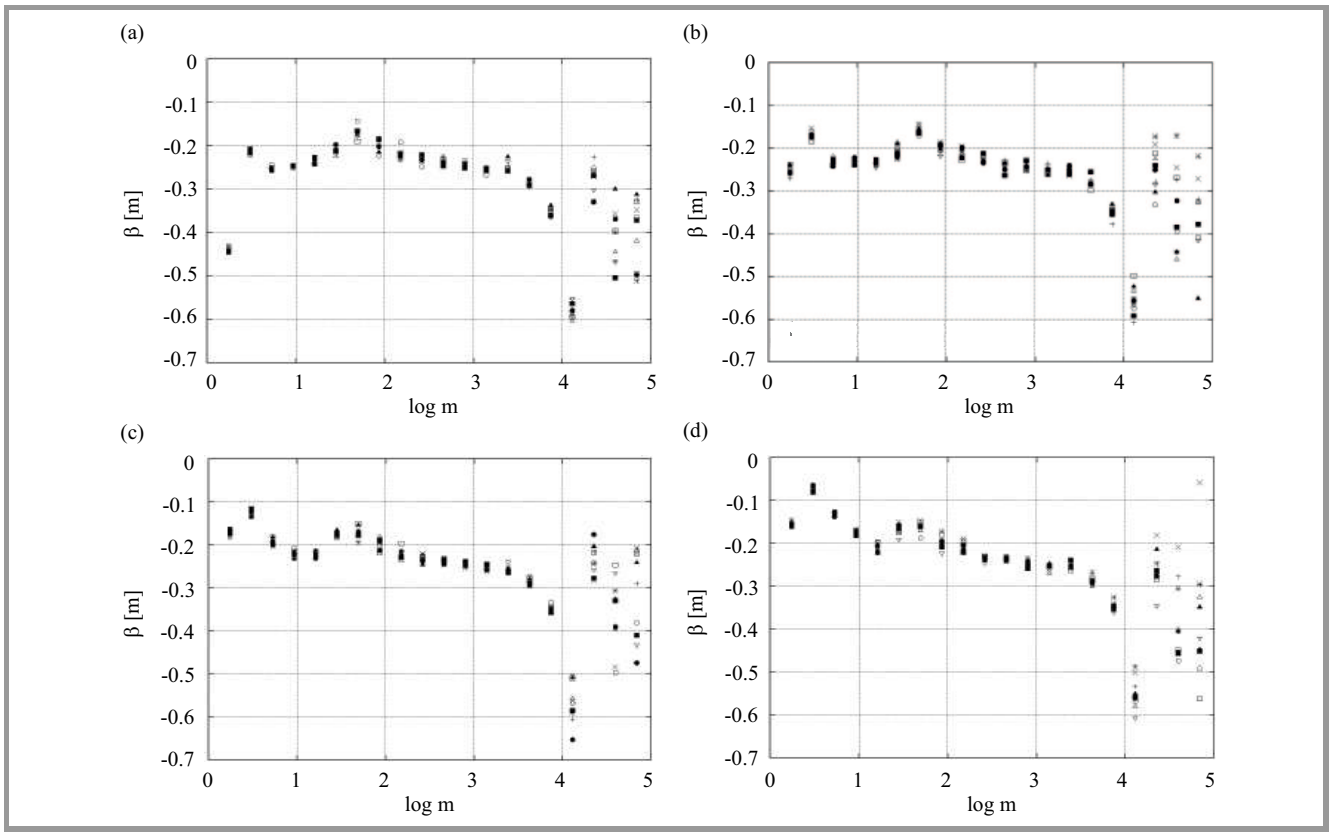


Fig. 6. User signatures for (a) g^1, (b) g^8, (c) g^{16}, and (d) g^{32}.

5. Uniqueness Improvement

So far the case where different machine models yield different signatures was considered. This result suggests that replacing some hardware or software components of C-PC might produce signatures that satisfy requirements R1 and R2. This section discusses the way how to change user signatures without adding a hardware device to C-PC. Note that as mentioned in the previous section, changing user signatures is often necessary for security.

5.1. Process Interference

One approach for uniqueness improvement is to increase the number of WinDump processes on C-PC. Let g^k denote machine g on which k WinDump processes are running. Figure 6 shows user signatures for g^1, g^8, g^{16}, and g^{32}. As shown in the figure, each number k creates a unique signature.

Figure 7 exhibits distances d_i between signatures of g^1 and g^k, $k > 1$. The figure demonstrates that the process interference-based approach generates many distinguishable signatures since $D(g^1, g^k) \geq 1$ for $k \in \{4, 8, 16, 32\}$. Figure 7 also provides the following attractive facts:

- distances $d_i(g^1, g^k)$ at $1 \leq i \leq 4$ increase with k,
- numbers i that satisfy $d_i(g^1, g^k) \geq L_i$ increase with k.

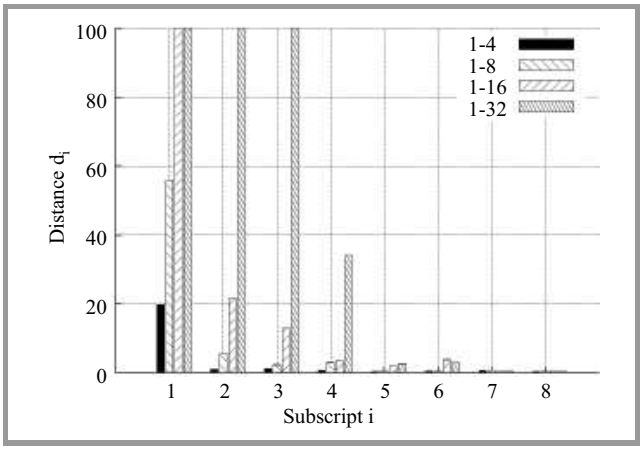


Fig. 7. The impact of the number of WinDump processes on distances d_i – "1-4" indicates $d_i(g^1, g^4)$.

The author conjectures that these phenomena arise due to process interference. Every time a packet arrives at C-PC, a packet processing process and k WinDump processes all start at once, so that they severely compete for CPU and buffer resources if k is large. Number k represents the degree of competition. As k increases, packet arrival timestamps become more inaccurate, and this inaccuracy result in the emergence of unique user signatures.

The process interference approach is neither CPU nor memory intensive. Therefore this approach can be considered

as a key technology for traffic signature-based user authentication.

5.2. Other Approaches

This subsection explores other possible approaches for producing unique signatures and investigates whether they are available under various hardware and software configurations. Experiments were made with 14 machines *e*-*r*, whose system information is listed in Table 3. The research is focused on Windows machines because most of personal desktop and laptop PCs use Windows OS. For comparison, Mac and Linux machines are included in the table. On Mac and Linux machines, tcpdump [15] runs instead of WinDump. Since their packet analysis mechanisms are different [16], experimental results may depend on which of them is used. The table also shows the maximum distances ($\max_i d_i$) between signatures measured before and after each of the following three operations:

Snaplen: The snapshot length of each packet collected by WinDump or tcpdump is increased to 4096 bytes (the default is 68 bytes). As a result of this, a larger data amount are stored in the hard disk.

Interfer: Eight WinDump (or tcpdump) processes are executed so that they severely compete for CPU and memory resources. Figure 6 is the result obtained by this operation with machine *g* in Table 3.

Load: The machine workload is raised by executing eight VLC media players [17], all of which play a video file on the hard disk.

In Table 3, symbol ◎ implies that the operation has an ability to create distinguishable signatures, and ○ indicates that distinguishable signatures may be obtained if the operation is adequately tuned (e.g. the snapshot length further increases). The table shows the following three results:

- the snapshot length-based approach may not yield long distances,
- the load-based approach is not suited to laptop PCs since the online news may abnormally terminate,
- the process interference-based approach is the most effective and stable approach.

However, this approach should be adequately tuned since $\max_i d_i$ depends on the machine configuration. Some machines require a large number of WinDump (or tcpdump) processes.

An advantage of the process interference-based approach is that a variety of recent and ongoing computer technologies keep producing unique and unpredictable signatures. Even if detail hardware and software specification of C-PC is given, obtaining user signatures through computation must be a difficult task. At the same time, however, new computer technologies make $\max_i d_i$ variable, so that the number of WinDump processes may need to be revised. The author considers the following recent technologies must have impacts on d_i:

Table 3
Three operations are performed to see whether they can yield distinguishable signatures under various machine configurations of C-PC. Symbols ◎ and ○ indicate $\max_i d_i \geq L_i$ and $2 < \max_i d_i < L_i$, respectively. Numbers in parentheses denote $\max_i d_i$

PC	OS	CPU	Snaplen	Interfer	Load
e	Vista	Q9450		○ (5)	
f	Win 7	i5-2400S		(2)	◎ (54)
g		i7-930		◎ (18)	◎ (51)
h		i7-960	○ (5)	◎ (104)	○ (3)
i		i3-2120	○ (6)	◎ (69)	○ (3)
j		i7-2600		◎ (62)	◎ (90)
k		i3-2130		○ (3)	◎ (37)
l	Win 8	i5-3350P	◎ (16)	◎ (56)	
m		i7-4770	◎ (15)	◎ (37)	◎ (13)
n (1)	Win 7	AMD	◎ (12)	◎ (66)	(4)
o (1)		Atom		◎ (38)	(4)
p (1)	Win 8	i5-3317U	◎ (24)	◎ (59)	
q	Linux	i7-3770K		(3)	
r	MAC	i7-2630		○ (5)	

Notes:
(1) *n*, *o*, *p* – laptop computers.
(2) $\max_i d_i = 2$ at 16 processes.
(3) $\max_i d_i = 2$ at 64 processes.
(4) The online news abnormally terminates.

Timestamp precision: WinPcap (a Windows library used by WinDump) by default obtains the timestamp through kernel function KeQueryPerformanceCounter(), which provides a time reference with microsecond precision. By modifying a registry key, timestamps are generated through faster i386 instruction RDTSC, which accesses TimeStamp Counter (TSC), whose precision is equivalent to the CPU frequency. RDTSC works only on Intel CPUs and is expected to provide nanosecond time resolution [18], [19].

Multiprocessing: A packet received by a NIC is stored in the NIC driver buffer. The timestamp of the packet is measured after the capture driver is invoked through a hardware interrupt. If there are pending interrupts, the driver is executed after all these interrupts are served. Therefore, the timestamp is significantly inaccurate if there are a large number of pending interrupts [18]. WinPcap works on symmetric multi-processing (SMP) machines. Multiple processors concurrently execute the same instance of the

capture driver, so that each processor handles a different packet stored in the NIC driver buffer. Accordingly, the delay of the driver execution depends on the number of processors [20].

Turbo Boost: Turbo Boost is a technology that enables the processor to run above its base operating frequency when workload on the processor calls for faster performance. The timestamp accuracy is affected by the technology since it dynamical changes processing capability.

5.3. Traffic Control

Some traffic control software tools effectively create unique signatures (probably because they frequently consult the current time). Traffic control is performed to reduce congestion, latency and packet loss by prioritizing, controlling, or reducing the network traffic. One of the traffic control tools is dummynet [21]. It emulates a network link that consists of a transmission link with fixed bandwidth B and propagation delay t_D and a finite FIFO queue with tail-drop. For link emulation, dummynet delays each packet i by $(\ell_i + Q_i)/B + t_D$, where ℓ_i is the length of packet i and Q_i is queue occupation when packet i was queued.

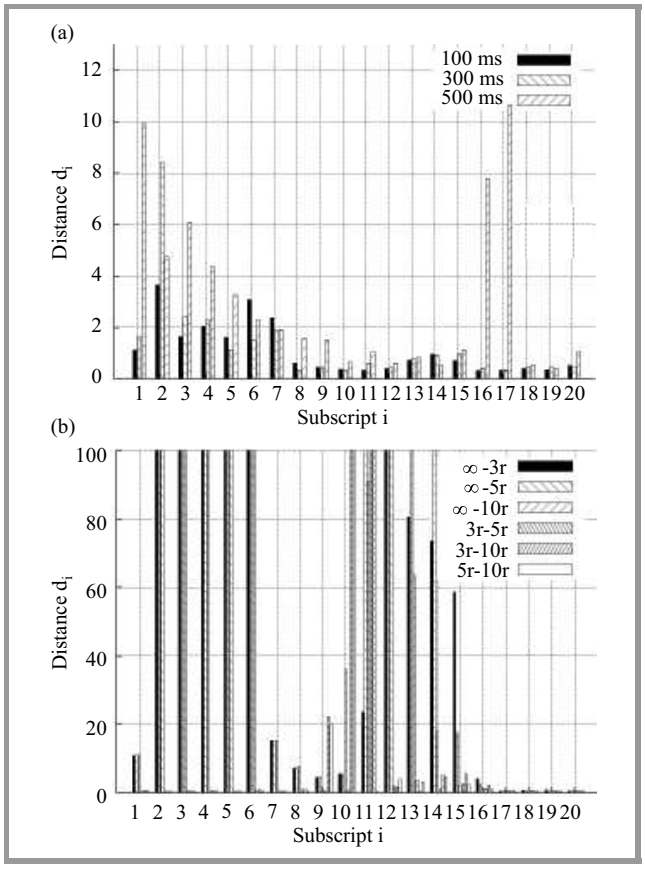


Fig. 8. The impact of: (a) propagation delay t_D and (b) bandwidth B on the distance. "100 ms" indicates the distance between two signatures measured at $t_D = 0$ and $t_D = 100$ ms. "$3r-5r$" indicates the distance between two signatures measured at $B = 3r$ and $B = 5r$.

Figure 8a shows distances between signatures measured at $t_D = 0$ and $t_D > 0$, where dummynet on C-PC delays every incoming and outgoing packets by t_D. Since video quality deteriorates greatly when $t_D = 500$ ms, the TCP window scale option [22], which allows larger windows to be used, is set to work when $t_D = 500$ ms. The figure demonstrates that $\max_i d_i$ is too small to distinguish signatures for all t_D values. On the other hand, as shown in Fig. 8b, dummynet bandwidth B is useful in raising $\max_i d_i$ greatly. From the figure, $\max_i d_i$ exceeds 100 at multiple values of i. By looking closely at Fig. 8b, it can be seen that in the case of "$\infty - 3r$", $\max_i d_i > 100$ at $i \in \{2,3,4,5,6,12\}$, where "$\infty - 3r$" denotes the distance between two signatures measured at $B = \infty$ (i.e. the bandwidth is unlimited) and $B = 3r$, where r is the average rate of the video stream. Note that perceived video quality is not degraded as long as B is at least three times greater than r. All approaches in Subsection 5.2. change user signature $\{\beta_i\}$ only at small time scales i (e.g. $i \leq 5$), whereas by changing bandwidth B, $\{\beta_i\}$ varies at large i (e.g. $9 \leq i \leq 14$). This is an important advantage of this approach.

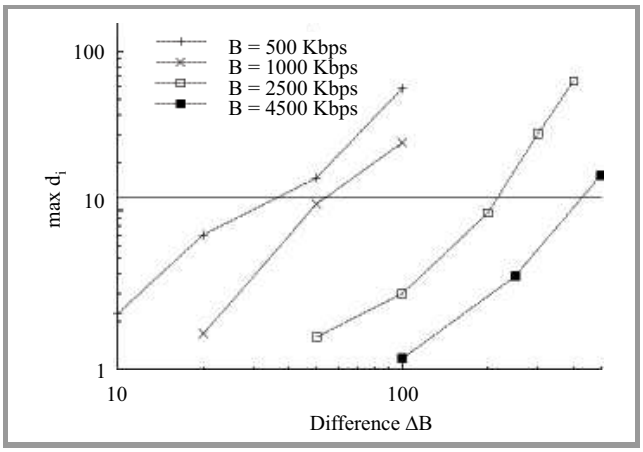


Fig. 9. A larger B requires a larger ΔB to satisfy $D \geq 1$. The solid line denotes $\max_{1 \leq i \leq 14} L_i$.

Figure 9 shows $\max_i d_i$ between two signatures obtained when the dummynet bandwidths are B and $B + \Delta B$ Kb/s. The solid line in the figure denotes $\max_{1 \leq i \leq 14} L_i (= 12)$, so that $D \geq 1$ if $\max_i d_i$ is above the line. From the figure, one can roughly estimate how many distinguishable signatures one can be obtained by changing the bandwidth, because the figure explains how $\max_i d_i$ increases with ΔB and how the minimum ΔB that satisfies $D \geq 1$ grows with B. For example, one can get roughly ten distinguishable signatures in the range of $500 \leq B \leq 1000$ Kb/s since from the figure, the smallest ΔB that satisfies $D \geq 1$ is approximately 50 Kb/s in the range.

6. Conclusions

For protecting users who place high value on their accounts, various what you have authentication technologies

have been proposed. However, they are not widely used today mainly because security hardware added to the user machines poses other new problems. Additional hardware is not necessary if the user machine itself is identified. In this paper, the feasibility of applying the traffic signature to the user machine identification has been discussed, where the signature is calculated from HTTP-based video traffic transmitted by the authentication server. This paper focused on uniqueness and reproducibility of the signature based on the distance function defined in this paper and obtained the following results:

Uniqueness was verified based on a criterion, which requires that the distance between any two signatures is not less than $L_s + \Delta L_i$, where $L_s = 7$ and ΔL_i is the fluctuation range of the i-th decay rate. The security strength corresponds to a six-digit code when $L_s = 7$. Although different machine configuration models tended to provide different signatures, these signatures did not always meet the criterion. However, the process interference approach, in which the number of executing packet-capture processes is used as a parameter for controlling the accuracy of packet arrival timestamps, was shown to be effective for producing signatures that meet the criterion.

Reproducibility was verified by calculating signatures from real Internet traffic delivered by France 24 live. Although the traffic traversed 30 routers and experienced a long propagation delay, signatures measured on various machines were stable especially over small time scales. Sample signatures showed that the fluctuation ranges ΔL_i were between 2.5 and 5. Therefore, $9.5 \leq L_s + \Delta L_i \leq 12$. When the machine load is highly increased by playing eight video files in parallel, five machines out of fourteen generated signatures that exceed $L_s + \Delta L_i$. Therefore, some machines need to reduce their loads before performing the authentication. However, dummynet, a traffic control tool, is expected to mitigate the impact of the load because dummynet generated many signatures whose distances from the original signature were significantly large (more than 100).

References

[1] R. P. Guidorizzi, "Security: Active authentication", *IT Professional*, vol. 15, no. 4, pp. 4–7, 2013.

[2] Federal Financial Institutions Examination Council, "Authentication in an internet banking environment", 2005 [Online]. Available: http://www.ffiec.gov/pdf/authentication_guidance.pdf

[3] M. Jakobsson, R. Chow, and J. Molina, "Authentication – are we doing well enough? [guest editors' introduction]", *IEEE Secur. & Priv.*, vol. 10, no. 1, pp. 19–21, 2012.

[4] D. DeFigueiredo, "The case for mobile two-factor authentication", *IEEE Securi. & Priv.*, vol. 9, no. 5, pp. 81–85, 2011.

[5] M. Sasse and C. C. Palmer, "Protecting you" [guest editors' introduction], *IEEE Secur. & Priv.*, vol. 12, no. 1, pp. 11–13, 2014.

[6] C. Herley, "More is not the answer", *IEEE Secur. & Priv.*, vol. 12, no. 1, pp. 14–19, 2014.

[7] K. Oida, "A traffic signature sensitive to client machines", in *Proc. Int. Conf. Adv. Comp. Inform. Technol. ACIT'13*, Kuala Lumpur, Malaysia, 2013.

[8] E. De Cristofaro, H. Du, J. Freudiger, and G. Norcie, "Two-factor or not two-factor? A comparative usability study of two-factor authentication", Computing Research Repository, 2013.

[9] K. Oida, "Video traffic attributes for end host identification", *Int. J. Comp. Commun. Engin.*, vol. 1, no. 4, pp. 396–401, 2012.

[10] Trusted Computing Group, "Trusted platform module (TPM) summary", July 2009 [Online]. Available: http://www.trustedcomputinggroup.org/resources/trusted_platform_module_tpm_summary

[11] O. Oyman and S. Singh, "Quality of experience for http adaptive streaming services", *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 20–27, 2012.

[12] J. Beran, *Statistics for Long-Memory Processes*. Chapman & Hall/CRC Press, 1994, vol. 61.

[13] WinDump [Online]. Available: http://www.winpcap.org/windump/

[14] T. Weigold, T. Kramp, and M. Baentsch, "Remote client authentication", *IEEE Secur. & Priv.*, vol. 6, no. 4, pp. 0036–43, 2008.

[15] tcpdump [Online]. Available: http://www.tcpdump.org/

[16] F. Risso and L. Degioanni, "An architecture for high performance network analysis", in *Proc. 6th IEEE Symp. Comp. & Commun. ISCC 2001*, Hammamet, Tunisia, 2001, pp. 686–693.

[17] VLC media player [Online]. Available: http://www.videolan.org/vlc/

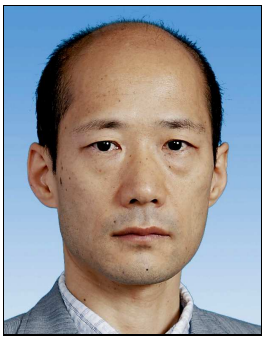
[18] L. Degioanni, M. Baldi, F. Risso, and G. Varenni, "Profiling and optimization of software-based network-analysis applications", in *Proc. 15th IEEE Symp. Comp. Architec. & High Perform. Comput. SBAC-PAD'03*, São Paulo, Brazil, 2003, pp. 226–234.

[19] P. Orosz and T. Skopko, "Performance evaluation of a high precision software-based timestamping solution for network monitoring", *Int. J. Adv. Softw.*, vol. 4, no. 1 and 2, pp. 181–188, 2011.

[20] G. Varenni, M. Baldi, L. Degioanni, and F. Risso, "Optimizing packet capture on symmetric multiprocessing machines", in *Proc. 15th IEEE Symp. Comp. Architec. & High Perform. Comput. SBAC-PAD'03*, São Paulo, Brazil, 2003, pp. 108–115.

[21] M. Carbone and L. Rizzo, "Dummynet revisited", *ACM SIGCOMM Comp. Commun. Rev.*, vol. 40, no. 2, pp. 12–20, 2010.

[22] V. Jacobson *et al.*, "TCP extensions for high performance", IETF, RFC 1323, May 1992 [Online]. Available: www.ietf.org/rfc/rfc1323.txt



Kazumasa Oida received the Bachelor of Information Science, Master of Engineering, and Doctor of Informatics degrees from the University of Tsukuba in 1983, Hokkaido University in 1985, and Kyoto University in 2002, respectively. He is currently a Professor in the Department of Computer Science and Engineering, Fukuoka Institute of Technology, Japan. His main interests include analysis and modeling of packet traffic and the origin of adaptive behavior in complex dynamic systems.
E-mail: oida@fit.ac.jp
Department of Computer Science and Engineering
Fukuoka Institute of Technology
Fukuoka, 811-0295 Japan