International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:1, 2007

Sensing Pressure for Authentication System Using Keystroke Dynamics

Hidetoshi Nonaka, and Masahito Kurihara

***Abstract*—**In this paper, an authentication system using keystroke dynamics is presented. We introduced pressure sensing for the improvement of the accuracy of measurement and durability against intrusion using key-logger, and so on, however additional instrument is needed. As the result, it has been found that the pressure sensing is also effective for estimation of real moment of keystroke.

***Keywords*—**Biometric authentication, Keystroke dynamics, Pressure sensing, Time-frequency analysis.

I. INTRODUCTION

THERE is an increasing need to authenticate and identify individuals in real-time and automated process for system security and robust safeguards. Biometric authentication is now popular for various applications: banking; airport security; forensics; computer login; and so on. It includes fingerprint, palm-print, iris scan, retinal scan, speech, face, vein pattern, x-ray dental image, and signature. They have some advantages over other methods that there is no keeping one's memory as password or no holding identification object as ID card. Biometric authentication technologies are classified into two categories according to what is measured: physical characteristics; and behavioral characteristics. Authentication by keystroke dynamics can be also classified as a behavioral biometric authentication technique in the meaning that it is free from oblivion or missing [1], [2].

Keystroke dynamics is measured with dwell time and flight time in keyboard typing. By retaining distinctive rhythm or cadence of legitimate user as the profile, computer intrusions or malicious accesses can be detected and rejected. Some systems refer to "Key-Press" events, and others fetch

"Key-Up" events, consequently, these systems have remarkable merit that no additional hardware is required. On the other hand, they suffer from two disadvantages as trade-off.

Firstly, the timing of keyboard event involves variable delay. In the case of USB-HID keyboard, the keystroke data are sent to PC only once in every 8 ms. Moreover, in some operating systems, each fetch of keyboard event delays up to 30 ms. These delays are not negligible by considering that instantaneous typing speed will exceed 20 strokes/s. The real timing of keystroke can be only estimated statistically without additional sensing instruments.

Secondary, authentication by only keystroke dynamics is vulnerable to key-loggers. Key-loggers are originally developed for evaluation or improvement of user interfaces. They collect not only the order of keys but also the timing of strokes. Various systems are distributed as commercial software/hardware and open source applications. They are lawful systems as backup tools for legitimate inputs against disk-crash, as watchdog tools for unauthorized accesses, as monitoring tools for observation of children, and so on. But some misapplications for intrusive accesses or thefts are reported, for example, interception by bypassing hardware key-logger between keyboard and PC in public places as internet café.

In this paper, we propose a measuring system of keystroke dynamics with sensing keying pressure. It has been reported that sensing pressure with tapping rhythm itself is effective for authentication [3]. Furthermore, real timing of keystroke can be measured by waveform matching of time series from pressure sensors. Both of the accuracy of measurement and durability against intrusion are improved in compensation for additional attachment of pressure sensors.

II. SYSTEM ? ONFIGURATION

Fig. 1 shows the configuration of keystroke pressure measurement system. Two pressure sensors (piezo-registive films: FlexiForce by Nitta) are attached between the undersurface of keyboard and cork blocks. The one is on the opposite place of "W" key, and the other "O" key. The shape of each sensor is circle with 10 mm in diameter and 0.1 mm in thickness. The data from each sensor are sent to A/D converter through band pass filter and amplifier to be converted to 10-bit digital data in 1.0000 kHz rate. The difference from previous data in 8-bit are preserved temporarily in a ring buffer of serial memory. The size of the ring buffer is 1,048,576-bit, then it can retain the measured data of last 65,536 ms. According to a request via USB from PC, previous unsent data in the ring buffer are transferred to PC.

Hidetoshi Nonaka is with Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060 0814, Japan
(e-mail: nonaka@main.ist.hokudai.ac.jp).

Masahito Kurihara is with Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060 0814, Japan
(e-mail: kurihara@main.ist.hokudai.ac.jp).

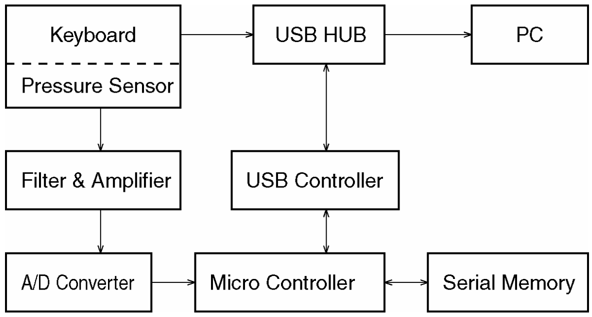


Fig. 1. Hardware configuration of keystroke pressure measurement system.

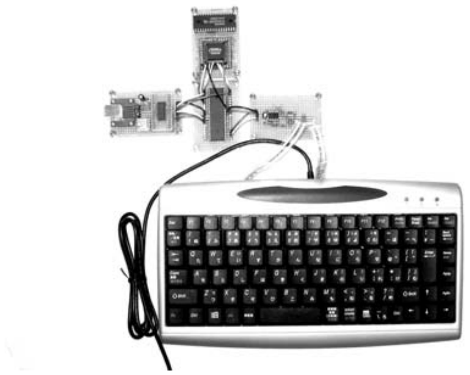


Fig. 2. Prototype of keystroke pressure measurement system with a keyboard.

The transmission rate is 12 Mbps (USB full speed). Each request is issued by PC with either keystroke event or timer interrupt in about 60 Hz, then the size of data is less than 16 bytes.

Fig. 2 shows a prototype of keystroke pressure measurement system with membrane-type USB keyboard. The waveform also depends on keyboards, because each keystroke does not reach the sensor directly but propagates through the key, mechanics and chassis of keyboard. There are various kinds of keyboards: mechanical keyboards, ones with micro-switches, ones with conductive rubbers, or membrane keyboards. Therefore, the solution of appropriate attachment of sensor is not trivial. We have tested various materials for sensor attachment: natural rubber, silicon rubber, wood, metal, urethane gel for shock absorption, vibration-proof material, acrylic resin, polyurethane, and so on. For the present, we have adopted cork blocks by reason of stability, noiselessness, and applicability to various kinds of keyboards. The alignment of sensors with cork blocks is depicted in Fig. 3.

Fig. 4 shows an example of keystroke indicated in the monitoring window. In this example, a user typed "stroke". In the upper panel, the left side of each triangle represents the timing of "Key-Down" event of corresponding key, and the opposite vertex represents the timing of "Key-Up" event. Some overlaps are observed between Key-Down and Key-Up events: "s-t" and "k-e". Middle panel and lower panel are the waveforms of right and left pressure sensors respectively. As is shown in these panels, "o" and "k" are dominant in the right waveform, on the other hand, "s", "t", "r" and "k" are dominant in left one. Moreover, a train of fluctuation is observed previous to each key event. The latency sometimes reaches 50 ms. The next section is devoted to explain the method for estimating real moment of keystroke with the timing of events and the waveform of pressure sensors.

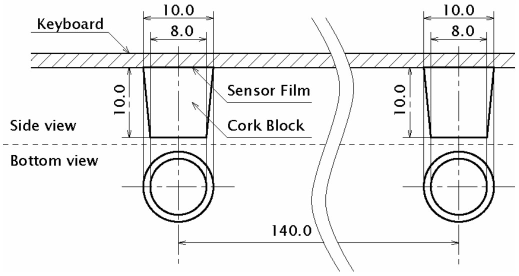
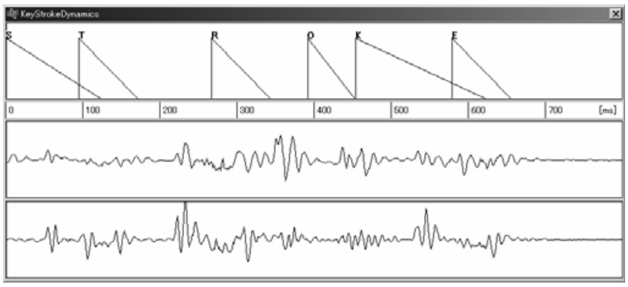


Fig. 3. Alignment of pressure sensors.



□ Monitoring window and an example of keystroke: "stroke."

III. Estimation of Real Moment of Keystroke

For time series $\tilde{v}_{0,t},\ t = 0,\cdots,2^J - 1 \equiv n-1$, we define transforms as:

$$
\begin{aligned}
\tilde{w}_{1,t} &= \tfrac{1}{2}(\tilde{v}_{0,t+1(\mathrm{mod}\,n)} - \tilde{v}_{0,t}),\\
\tilde{v}_{1,t} &= \tfrac{1}{2}(\tilde{v}_{0,t+1(\mathrm{mod}\,n)} + \tilde{v}_{0,t}),\\
\tilde{w}_{2,t} &= \tfrac{1}{2\sqrt{2}}(\tilde{v}_{1,t+3(\mathrm{mod}\,n)} - \tilde{v}_{1,t}),\\
\tilde{v}_{2,t} &= \tfrac{1}{2\sqrt{2}}(\tilde{v}_{1,t+3(\mathrm{mod}\,n)} + \tilde{v}_{1,t})\\
&\vdots\\
\tilde{w}_{j,t} &= \tfrac{1}{2^{j/2}\sqrt{2}}(\tilde{v}_{j-1,t+2^j-1(\mathrm{mod}\,n)} - \tilde{v}_{j-1,t}),\\
\tilde{v}_{j,t} &= \tfrac{1}{2^{j/2}\sqrt{2}}(\tilde{v}_{j-1,t+2^j-1(\mathrm{mod}\,n)} + \tilde{v}_{j-1,t})\\
j &= 1,2,\cdots,J
\end{aligned}
\tag{1}
$$

The original waveform from pressure sensor corresponds to $\tilde{v}_{0,t}$, and $\tilde{w}_{i,t}$, $\tilde{v}_{i,t}$ correspond to the time-frequency decomposition of the original waveform. We can rewrite (1) in matrices form:

$$\begin{aligned} \tilde{W}_1 &= \tilde{H}_0\tilde{V}_0,\ \tilde{V}_1 = \tilde{G}_0\tilde{V}_0 \\ \tilde{W}_2 &= \tilde{H}_1\tilde{V}_1,\ \tilde{V}_2 = \tilde{G}_1\tilde{V}_1 \quad , \\ &\vdots \end{aligned} \tag{2}$$

where

$$\begin{aligned} \tilde{W}_j &\equiv [\tilde{w}_{j,0}, \tilde{w}_{j,1}, \cdots, \tilde{w}_{j,n-1}]^T \\ \tilde{V}_j &\equiv [\tilde{v}_{j,0}, \tilde{v}_{j,1}, \cdots, \tilde{v}_{j,n-1}]^T , \end{aligned}$$

and

$$\tilde{H}_0 \equiv \frac{1}{2}\begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} \tilde{v}_{0,0} \\ \tilde{v}_{0,1} \\ \tilde{v}_{0,2} \\ \vdots \\ \vdots \\ \tilde{v}_{0,n-3} \\ \tilde{v}_{0,n-2} \\ \tilde{v}_{0,n-1} \end{bmatrix}$$

$$\tilde{G}_0 \equiv \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{v}_{0,0} \\ \tilde{v}_{0,1} \\ \tilde{v}_{0,2} \\ \vdots \\ \vdots \\ \tilde{v}_{0,n-3} \\ \tilde{v}_{0,n-2} \\ \tilde{v}_{0,n-1} \end{bmatrix}$$

$$\tilde{H}_1 \equiv \frac{1}{2\sqrt{2}}\begin{bmatrix} -1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} \tilde{v}_{1,0} \\ \tilde{v}_{1,1} \\ \tilde{v}_{1,2} \\ \vdots \\ \vdots \\ \tilde{v}_{1,n-3} \\ \tilde{v}_{1,n-2} \\ \tilde{v}_{1,n-1} \end{bmatrix}$$

$$\tilde{G}_1 \equiv \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \tilde{v}_{1,0} \\ \tilde{v}_{1,1} \\ \tilde{v}_{1,2} \\ \vdots \\ \vdots \\ \tilde{v}_{1,n-3} \\ \tilde{v}_{1,n-2} \\ \tilde{v}_{1,n-1} \end{bmatrix}$$

The set of transforms $\tilde{W}_1, \tilde{W}_2, \cdots, \tilde{W}_J$ and $\tilde{V}_J$ from $\tilde{V}_0$ is equivalent to maximal overlap discrete Haar wavelet transform (MOHWT). It is known that energy is decomposed properly as

$$\|\ \tilde{V}_0\ \|^2 = \sum_{j=1}^{J} \|\ \tilde{W}_j\ \|^2 + \|\ \tilde{V}_J\ \|^2 \ , \tag{3}$$

where $\|\ \tilde{V}_0\ \| \equiv \sum_{t=0}^{n-1} \tilde{V}_t^2$, and so on.

Furthermore, V_0 can be reconstructed and partially reconstructed using inverse transformation, though it is not ortho-normal but only orthogonal:

$$\tilde{V}_0 = \sum_{j=1}^{J} \tilde{H}_j{}^T \tilde{W}_j + \tilde{G}_J{}^T \tilde{V}_J \ \text{(Full reconstruction)} \tag{4}$$

$$\bar{V}_0 = \sum_{\{j\}\subset[1,\,J]} \tilde{H}_j{}^T \tilde{W}_j + \tilde{G}_j{}^T \tilde{V}_j \ \text{(Partial reconstruction)} \tag{5}$$

The definition of wavelets and scaling functions are different from those of conventional MOHWT. It is only for the convenience of successive computation of transform in real time. The computation of transform and inverse transform require $O(n \log_2 n)$ multiplications [4].

Partially reconstructed waveform is compared with the reference patterns by Successive DP Matching Method (S-DP) [5]. A brief description of S-DP is as follows.

Let

$$\begin{aligned} \tilde{V} &= \cdots, \tilde{v}_{0,i}, \cdots \tilde{v}_{0,1}, \tilde{v}_{0,0} \\ X &= \cdots, x_j, \cdots x_1, x_0 \end{aligned} \tag{6}$$

be two discrete time series, where X is one of reference pattern. The time-normalized distance D is defined as

$$D(X,Y) = \min_{\substack{i=i(k) \\ j=j(k)}} \left[\sum_{k=0}^{-K} d(i,j) \right], \tag{7}$$

where $K > 0$ is a constant and $d(i,j)$ is the distance between $\tilde{v}_{0,i}$ and x_j . Warping function $i(k)$ and $j(k)$ are subject to following conditions:

$$\begin{aligned} i(k-1) &\leq i(k) \\ j(k-1) &\leq j(k) \\ i(k-1)+j(k-1) &\neq i(k)+j(k) \\ |\, i(k)-j(k)\,| &< r \end{aligned} \qquad (8)$$

for a constant $r > 0$. The minimization problem (7) can be solved by dynamic programming:

$$\begin{aligned} &D_0 = d(0,0) \\ &D_{k-1} = d(i(k),j(k)) + D_k \quad k = 0,-1,-2,\cdots,-K+1 \qquad (9) \\ &D = D_{-k}\,, \end{aligned}$$

where the pair $(i(k), j(k))$ is chosen from following three:

$$\{(i(k)-1, j(k)),\ (i(k), j(k)-1),\ (i(k)-1, j(k)-1)\}$$

such as to minimize $d(i(k), j(k))$ under the condition:

$$|\, i(k)-j(k)\,| < r$$

The estimation and matching process is summarized in Fig. 5. When keystroke event is detected, the last waveform of 32 ms is decomposed by (2). If it is not accord to the reference spectrum (eigen-frequency of the keyboard and attachment of pressure sensors), the time origin is shifted back and the decomposition process is executed iteratively. When it accords the reference spectrum, the estimated timing of keystroke is obtained. The waveform is partially reconstructed by (5), normalized with their amplitude, and compared with the reference patterns using successive DP matching. Reference patterns are obtained a priori for each user by background measurement of keystrokes.

Table 1 and Fig. 6 show an example of estimation of real moment of keystroke. As is shown in the figure, the real timing precedes the event timing in 22 ms and 50 ms. Fig. 7 shows the latency between keystroke event fetched by operating system and estimated real moment of keystroke by 1,000 times trials. The average is 27.81 ms and the value varies between 5 ms and 55 ms. It is obviously significant for the application to authentication using keystroke dynamics.

```
int estimation(void){
  //Return the estimated keystroke time in ms
  fetch-key-event();
  int t = 0;
  while(t > -T){
    Calculate $\tilde{W}_j$ and $\tilde{V}_j$ by(2);
    if($\tilde{W}_j$ ,$\tilde{V}_j$ accord the reference spectrum){
      Reconstruct $\overline{V}_0$ by(5);
      Normalize $\overline{V}_0$ with amplitude;
      if(S-DP-Matching($\overline{V}_0$)) return t;
    }
    t --;
  }
  return t;
}
```

Fig. 5. Pseudo code of estimation and matching process (simplified).

Table 1. A result of estimating the real moment of keystroke.

	Event	Estimation	Latency	Magnitude
Key-Down	154 [ms]	104 [ms]	50 [ms]	232
Key-Up	248 [ms]	226 [ms]	22 [ms]	95

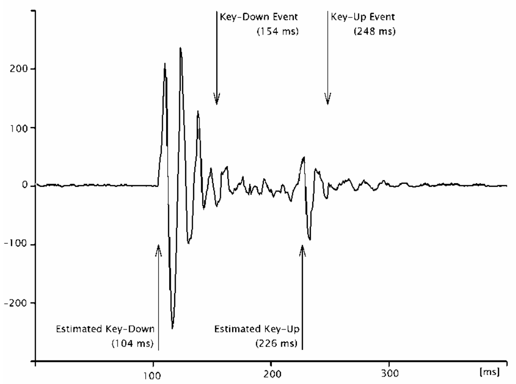


Fig. 6. Waveform of measured data and estimation of real moment of keystroke.

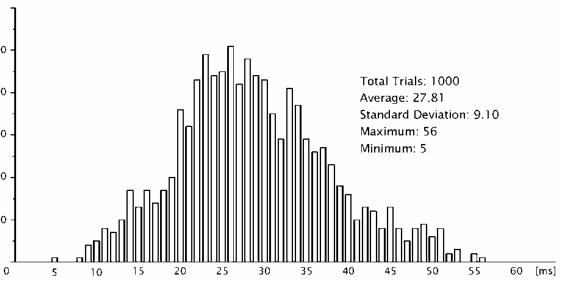


Fig. 7. Latency between keystroke event and estimated keystroke.

IV. Conclusions

We proposed an authentication system using keystroke dynamics with pressure sensing. The primary motivation of introducing the pressure sensing is the improvement of the durability of authentication against key-logger, but it is found that the pressure sensing is also effective for estimating the real moment of keystroke. At present, our system has been tested for a few participants. The evaluation of performance and capability for authentication with diverse users is under way.

References

[1] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi, "User authentication through keystroke dynamics," *ACM Transaction on Information and system security* (TISSEC), **5**, 4, pp. 367-397, 2002.

[2] BioNet Systems/ Net Nanny/ Stanford Research Institute, "BioPassword," http://www.biopassword.com.

[3] Neil Henderson, Neil White, Raymond Veldhuis, Pieter Hartel, and Kees Slump, "Sensing pressure for authentication," *Proceedings of 3rd IEEE Benelux Signal Processing Symposium* (SPS-2002), 2002.

[4] Hidetoshi Nonaka, and Masahito Kurihara, "Time-frequency decomposition in gesture recognition system using accelerometer," *Knowledge-Based Intelligent Information and Engineering Systems*, LNAI **3212** (8th Int. Conf. KES2004), pp. 1072-1078, 2004.

[5] Hidetoshi Nonaka, and Masahito Kurikara, "Anticipatory matching method for query-based head gesture identification, *International Journal of Computing Anticipatory Systems*, **13**, 2004 (in print).