# Key-typing Biometrics Using Japanese Input Method Characteristics

Kunimi Hirano[1] and Alberto Palacios Pawlovsky[2]

(2010年2月20日　受理)

*Abstract*–Advances in technology makes now possible the use of biometrics in many portable devices as cellular phones and laptop computers. Typing biometrics is also one way of differentiating the characteristics of a user and has the advantage that it does not need a specialized input device. In this work, we show how the use of the characteristics of typing in Japanese could help in improving keystroke dynamics in this language. We show that we can lower the equal error rate (EER) of a system to almost 1/4 of the one of an implementation that does not use those characteristics.

*Keywords*–Biometrics, identification, keystroke, Japanese, input method.

## I. Introduction

The increasing availability of cheaper and powerful microprocessors of high integration has also fostered the proliferation of multiple and sophisticated sensors and other devices. These in turn have made possible the use of biometrics in many portable products. We have biometrics related input devices in cellular phones, notebooks and even in USB memories. Processing technologies have also advanced and now we have digital cameras with face detection technologies. They would be able, probably in a few years, to even recognize faces. In many portable devices such as cellular phones and portable computers security relies still in the input of a user name and a password or only a password (PWD). Even in touch-based devices that input is possible through a virtual keyboard. With the proliferation of the internet and services based on it there has been an increase in the number of works focusing in keystroke-based biometrics. Recently ones focus on the analysis and methods for the analysis of keystroke data[1],[2],[3] and its application for secure sites[4]. We have been working also on the processing of keystroke data and methods to diminish the number of false authentications. In a previous work we showed a method to improve the reliability of a system dividing the users according to the method of determining the confidence margin of their data [5]. In our group we also developed three methods to improve the accuracy of authentication [6].

In section II we give a brief introduction to the

[1]Department of Electronics and Information Engineering, [2]Department of Robotics Engineering, Faculty of Engineering, Toin University of Yokohama, 1614 Kurogane-cho, Aoba-ku, Yokohama, 225-8502

system we have used in this work. In section Ⅲ we explain the characteristics particular to the use of a Japanese password in the authentication process. In section Ⅳ we show the accuracy results with different implementations. In section V we give some conclusions and discuss some topics that need further work.

## Ⅱ. Keystroke-based identification

The main blocks of our keystroke-based authentication system are shown in Figure 1. A user requiring an input to the system must enter a password that will then be compared to a pool of templates of data of authorized users. If the password and input characteristics match one of the templates under some requirements, the user will be allowed to enter the system. If the input does not fulfill those requirements the user will not be authorized to enter.

The flow diagram of the verification process is shown in Figure 2. The user is required an identification number (ID) and if it is valid the system will require the input of a password (PWD) that is common to all the users. Once the required password is entered in the system, it will check if the input method to enter the password correspond to one or more templates in the system. If it does not find a corresponding template, the user will not be allowed to enter the system. If the input method matches one or more templates of authorized users, the system will then check if the

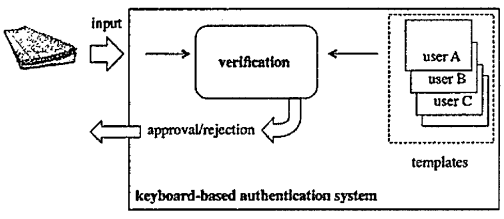


Fig.1. Block diagram of our authentication system

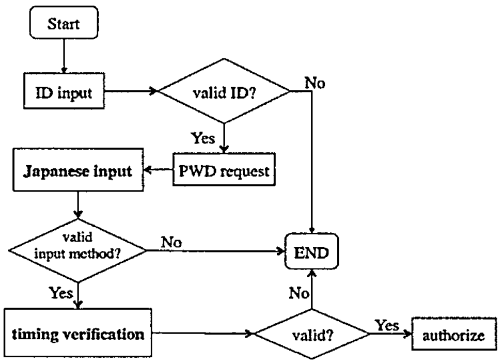


Fig.2. Flow diagram of the verification process

timing characteristics of the input fits the requirements impose by the system. If the input fulfills them the user will be authorized. Otherwise, it will not be able to enter the system.

As explained above, our system use a common password or all its users. It is a phrase in Japanese that allows us to identify several users by its method of entering this phrase using a keyboard. Details regarding these topics are given in the following section.

## Ⅲ. Japanese password based authentication

Japanese language uses four alphabets. One of them (kanji: Chinese ideograms) is derived from Chinese and is the most frequently used in printed form. One more alphabet (hiragana) is used to express Japanese sounds and also in writing at the initial education level. Another one (katakana) is used to express words borrowed from foreign languages and that do not have an equivalent Japanese word. The last alphabet (romaji) is the one used in western countries. Usually, this alphabet is used when entering words using a keyboard. The way of entering words in Japanese is not unique. Figure 3 shows some examples of sounds in the two systems we can use to enter words. As could be seen from the sounds given as examples we will

| Romanization System | し | ち | つ | ふ | じ |
|---|---|---|---|---|---|
| Kunrei | si | ti | tu | hu | zi |
| Hepburn | shi | chi | tsu | fu | ji |

Fig. 3. Romanization types: some examples

have 32 different ways of entering them (5 sounds with 2 possible ways of entering each of them : $2^5$). Both are usually supported for all operating systems and applications that allow the entering of written data in Japanese.

We have implemented a system that uses these differences to identify a valid user. We use in our system only one password common to all users. Figure 4 shows this password and the ways each word can be entered (note that the third word has the same transcription in both systems but one more (unclassified) additional way of entering it).

| Password | 富士の地下室 | | | | | | |
|---|---|---|---|---|---|---|---|
| PWD basic words | ふ | じ | の | ち | か | し | つ |
| Kunrei | hu | zi | no | ti | ka | si | tu |
| Hepburn | fu | ji | | chi | (ca) | shi | tsu |

Fig. 4. The Japanese password used in our system

This password has in total 64 different ways of being entered ($2 \times 2 \times 1 \times 2 \times 2 \times 2 \times 2$). We classify the users by their way of entering this password. Any one trying to get access will also be classified according to his/her way of entering it. If the way it was entered does not match with any of the registered users the access will be denied without further processing. If it match with one registered way of entering the password, the time in entering it will be used to process it. The timing data will be compared with the templates of the users that have the same way of entering the password. The comparison is made using the time between keys, the time the keys remain pressed and the total time in entering the password. Figure 5 shows an abbreviated example that indicates the time values used in the validation process.

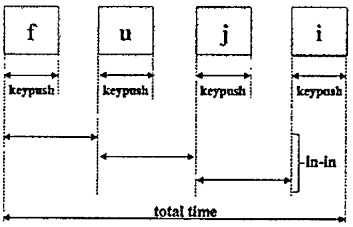


Fig. 5. Time values used in the validation process

## IV. Experimental Results

In keystroke biometrics we usually create one template for each user. A standard template contains the average values for each data, its standard deviation and the confidence weight of each item used in the validation process. An example of the calculation of the confidence weight is shown in Table I. These confidence weights are different for different users.

TABLE I Example of Confidence Weights of keypush Timings

| data | mean value (us) $\bar{t}$ | $1/\bar{t}$ | confidence weight |
|---|---|---|---|
| f | 107 | 0.0093 | 29.904% |
| u | 109 | 0.0092 | 29.582% |
| j | 190 | 0.0053 | 17.042% |
| i | 137 | 0.0073 | 23.472% |
| | | $\Sigma = 0.0311 = 100\%$ | $\Sigma = 100\%$ |

In Table II we show the confidence weights of three users of a hypothetical system.

TABLE II Confidence Weights' Example

| data | user A | user B | user C |
|---|---|---|---|
| f | 30% | 28% | 15% |
| u | 30% | 12% | 30% |
| j | 17% | 30% | 30% |
| i | 23% | 30% | 25% |

We also define a confidence interval. It is defined by a lower limit given by the average value minus one standard deviation, and an upper limit given by the average value plus one standard deviation (see Figure 6). The confidence weights are used with the

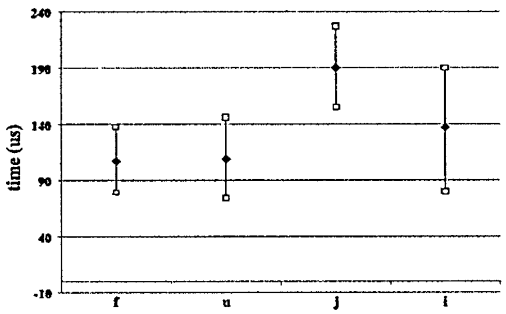


Fig. 6. Examples of confidence intervals

confidence intervals to identify a user. If the data entered by someone trying to gain access to the system lay in the corresponding confidence interval, the corresponding confidence weight will be added to a confidence level to determine if the user is accepted or not as a valid user. In Table III we give an example that shows that an input is **in** or **out** of the confidence interval for each user of Table II.

TABLE III Confidence Level Calculation Example

| data | user A | user B | user C |
|---|---|---|---|
| f | in | out | in |
| u | out | in | out |
| j | in | out | in |
| i | out | in | in |
| confidence level | 47% | 42% | 70% |

Looking at the confidence levels of this example we would say that the input probably corresponds to user C (the one with the highest confidence level). But, its approval also depends on the acceptance threshold set in the identification system. If it requires an 80% confidence level, the input will not be accepted as a valid one. We run several experiments with four and nine users. We also used several different settings to validate their data. All they have in common that they use the confidence intervals, confidence weights and the confidence level describe above. Details of each experiment are given in the following subsections with their corresponding results.

*A. Using only **keypush** timings: Lv0*

This four-user implementation did not use the characteristics of the Japanese language, only the time the keys remain pressed. We called this setting level 0 (we call it in what follows Lv0). The FAR (False Acceptance Rate) and FRR (False Rejection Rate) values obtained with this setting are shown in Figure 7. The point at which the rate of both accept and reject errors are equal is called the EER (Equal Error Rate) or CER (Crossover Error Rate). The Lower this rate, the more accurate the identification system is considered to be. With Lv0 we obtained an EER of 11.8% (at a threshold of 44.1%).

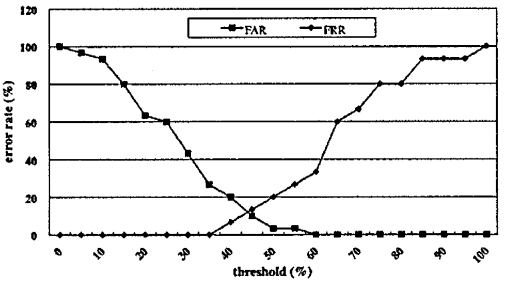


Fig. 7. FAR and FRR plotting using Lv0

*B. Lv0 + Japanese input characteristics: Lv1*

We also experimented adding to the Lv0 (four-user) system the capacity of recognizing different Japanese input methods (we call it in what follows Lv1). With it we obtained the FAR and FRR shown in Figure 8.

As could be seen from this figure, we were

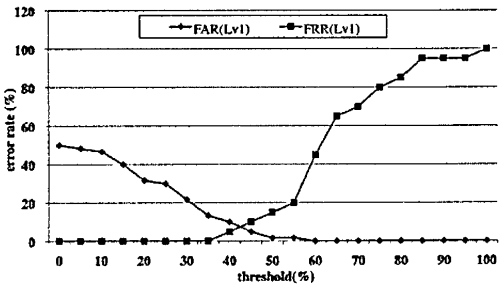


Fig.8. FAR and FRR plotting using Lv1

able to lower the FAR values (the FRR values remained the same). The lowering of the values of FAR let us obtain an EER of 7.5% at a threshold of 42.5%. The inclusion of the capacity of recognizing users according to their Japanese input method allowed us to improve the accuracy of our system by almost 36%.

*C. Using only **keypush** and **in-in** timings: Lv21*

We also experimented adding to Lv0 the processing of the timing values between keys (in-in in Fig. 5). This system improves on the values of Lv0 lowering EER from 11.8% to 10% at a threshold of 47.5% (see Figure 9). The values shown are for a four-user system.

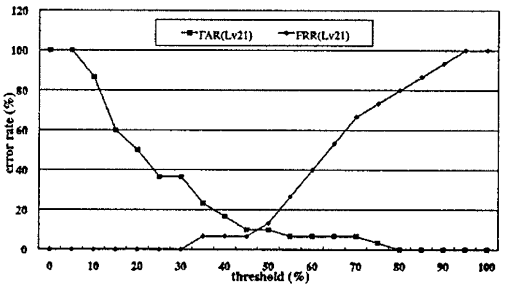


Fig.9. FAR and FRR plotting using Lv21

*D. Lv21 + Japanese input characteristics: Lv22*

We also implemented a system that uses the time the keys remain pushed, the timing between keys, and the capacity of recognizing users by their method of inputting Japanese.

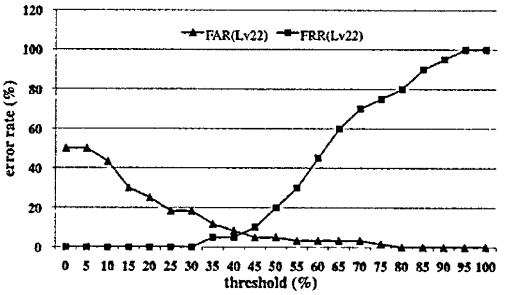


Fig.10. FAR and FRR plotting using Lv22

The FAR and FRR values for a four-user system are shown in Figure 10. The EER for this implementation was of 7% at a threshold of 42%.

*E. Lv22 + **total time** : Lv3*

We also implemented a system that adds to Lv22 the **total time** of inputting the password to the system (see Fig. 5). In a four-user implementation it gave an ERR of 6.3% at a threshold of 41.3%. In a nine-user implementation it gave an ERR of 3.5% at a threshold of 36.5%. The FAR and FRR values of this last implementation are shown in Figure 11.

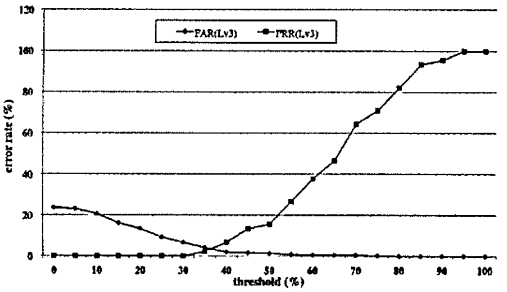


Fig.11. FAR and FRR plotting using a 9-user Lv3

The increase in the number of users also increased the number of ways to input the common password of our system and improved the ERR lowering it. With several ways of entering the password (several different templates) and with a small number of user per template, we can expect a very low EER. However, if the users of the system use a few ways of entering the password the EER that we can expect for a simple implementation (like Lv1) is high.

Table IV shows the EERs for all the implementations we tested in this work.

## V. Conclusions

We described a keystroke identification system that uses the way of inputting

TABLE IV EER of All Our Implementations

| 4-user | 9-user | EER | threshold |
|---|---|---|---|
| Lv0 | - | 11.8% | 44.1% |
| Lv1 | - | 7.5% | 42.5% |
| Lv21 | - | 10% | 47.5% |
| Lv22 | - | 7% | 42% |
| Lv3 | - | 6.3% | 41.3% |
| - | Lv3 | 3.5% | 36.5% |

Japanese sounds to improve the accuracy in identifying the users of it. Combining different timings and the characteristics of inputting Japanese sounds gave us in the best case an EER of 3.5%. This value is close to the 2% that the NIST [7] considers as a minimum necessary to make a biometrics-based identification method viable for its use in a security system. This is also close to the 2.5% reported in [8]. The approach described in this paper has the advantage of improving the security of a system using a Japanese password but its use is limited to one country and language. However, the idea could be useful for other languages too. The approach taken let us improve the FAR. One possible way of lowering more the EER could be in lowering the FRR and moving the threshold to the right (increasing it). In this work we only tried one of the three methods proposed in [6], so the application of the other two and their combination is also a topic for further study.

*Acknowledgments*

The authors wish to express their gratitude to the members of our laboratory and those people outside it that gently provided the data for testing the different implementations described in this paper.

References

[1] L. K. Maisuria, Cheng Soon Ong, and Wen Kin Lai "A comparison of artificial neural networks and cluster analysis for typing biometrics authentication," Proceedings of The International Joint Conference on Neural Networks (IJCNN99), Vol. 5, pp. 3295-3299, July 1999.

[2] W. Chen and W. Chang, "Applying hidden Markov models to keystroke pattern analysis for password verification," Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration (IRI 2004), pp. 467-474, November 2004.

[3] W. Chang, "Improving hidden Markov models with a similarity histogram for typing pattern biometrics," Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (IRI 2005), pp. 487-493, August 2005.

[4] S. T. de Magalhaes, K. Revett, and H. M. D. Santos, "Password secured sites - stepping forward with keystroke dynamics," International Conference on Next Generation Web Services Practices (NWeSP 2005), pp. 6, August 2005.

[5] Alberto Palacios Pawlovsky and Noboru Hirabayashi, "A Study on Keystroke-based Identification," Research Bulletin of the Toin University of Yokohama, Vol. 16, pp. 105-111, Yokohama, Japan, June 2007.

[6] Sumio Takahashi, "Keystroke biometrics: Methods to Improve Authentication Accuracy," B. Sc. thesis, Toin University of Yokohama, Faculty of Engineering, Dept. of Electronics and Information Engineering, Kanagawa, Yokohama, Japan, March 2009 (in Japanese).

[7] NIST (USA National Institute of Standards and Technology), Information Technology Laboratory (ITL) site: http://www.nist.gov/itl/

[8] Aleksander Svelokken Andersen, "Biometric Authentication and Identification using Keystroke Dynamics with Alert Levels," Master thesis, Oslo University College, May 23, 2007.