Recognizing Individual Typing Patterns

Michał Choraś[1] and Piotr Mroczkowski[2]

[1] Image Processing Group, Institute of Telecommunications
University of Technology & Life Sciences
S. Kaliskiego 7, 85-791 Bydgoszcz
chorasm@utp.edu.pl
[2] Hewlett Packard Polska, Global Delivery Poland Center
ul. Szturmowa 2a, University Business Center, Warsaw, Poland
piotr.mroczkowski@hp.com

Abstract. In the article three methods of extracting individual typing patterns are proposed and tested. Moreover, we present satisfactory experimental results confirming that these typing patterns may be used as biometrics for human identification, especially in web-based applications (e.g. password hardening).

1 Introduction

Individual typing patterns recognition systems analyze the way a user types at a terminal by monitoring the keyboard events. In such recognition systems, several things can be analyzed: time between key-pressed and key-released events, break between two different keystrokes, duration for digraphs and trigraphs and many more. In other words not what is typed, but how it is typed is important.

These characteristics of typing patterns are considered to be a good sign of identity and therefore may be used as biometrics for human identification and for enhancing web security in client-server applications [1][2][3].

Keystroke verification techniques can be divided into two categories: static and continuous. Static verification approaches analyze keyboard dynamics only at specific times, for example during the logon process. Static techniques are considered as providing a higher level of security than a simple password-based verification system [1]. The main drawback of such an approach is the lack of continuous monitoring, which could detect a substitution of the user after the initial verification. Nevertheless, the combination of the static approach with password authentication was proposed in several papers [4] and it is considered as being able to provide a sufficient level of security for the majority of applications. Our web identification system is based on such a combination.

Continuous verification, on the contrary, monitors the user's typing behavior through the whole period of interaction [1]. It means that even after a successful login, the typing patterns of a person are constantly analyzed and when they do not mach user's profile access is blocked. This method is obviously more reliable but, on the other hand, the verification algorithms as well as the implementation process itself, are much more complex.

J. Martí et al. (Eds.): IbPRIA 2007, Part II, LNCS 4478, pp. 323–330, 2007.

One of the first studies on keyboard biometrics was carried out by Gaines et al.[5]. Seven secretaries took part in the experiment in which they were asked to retype the same three paragraphs on two different occasions in a period of four months. Keystroke latency timings were collected and analyzed for a limited number of digraphs and observations were based on those digraph values that occurred more than 10 times [6].

Similar experiments were performed by Leggett with 17 programmers [4]. In the 15 last years, much research on keystroke analysis has been done (e.g., Joyce and Gupta [7], Bleha et al. [8], Leggett et al. [4], Brown and Rogers [9], Bergadano et al. [10], and Monrose and Rubin [1][6]).

Several proposed solutions got U.S. patents (for instance Brown and Rogers [11]). Some neural network approaches (e.g., Yu and Cho [12]) have also been undertaken in the last few years. More recently, several papers where keystroke biometrics, in conjunction with the login-id password pair access control technique, were proposed (e.g., Tapiador and Sigenza [13]). Some commercial implementations are also available ('Biopassword', a software tool for Windows platform commercialized by Net Nanny Inc. [14]).

2 Typing Patterns Characteristics

In the proposed and implemented individual typing pattern recognition system three independent methods of the identity verification are performed every time a user attempts to log in.

First and second method is based on the calculation of the degree of disorder of digraphs and trigraphs respectively. The last one compares typing paths stored in the database against a typing path created at the time of logon process. Hereby we present background of our methods.

2.1 Digraphs and Trigraphs

Digraph is defined as two keys typed one after the other. In our case the duration of a digraph is measured between the press event of the first key and release event of the second key.

Trigraph is defined as three keys typed one after the other. The duration of trigraph is measured between pressing event of the first key and release of the third key.

2.2 Degree of Disorder

Having two sets of key latencies of the same $Login-Password$ pair, it is possible to measure their "similarity". One way to calculate that is the degree of disorder (do) technique [10].

Let us define vector V of N elements and vector V', which includes the same N elements, but ordered in a different way. The degree of disorder in vector V can be defined as the sum of the distances between the position of each element

in V with respect to its counterpart vector V'. If all the elements in both vectors are in the same position, the disorder equals 0.

Maximum disorder occurs when elements in vector V are in the reverse order to the model vector V'. Maximum disorder (do_{max}) is given by:

$$do_{\max} = \frac{|V|^2}{2} \tag{1}$$

where $|V|$ is the length of V and it is even or by:

$$do_{\max} = \frac{(|V|^2 - 1)}{2} \tag{2}$$

where $|V|$ is length of V and it is odd.

In order to get the normalized degree of disorder (do_{nor}) of a vector of N elements, we divide do by the value of the maximum disorder. After normalization, the degree of disorder falls between 0 (V and V' have the same order) and 1 (V is in reverse order to V').

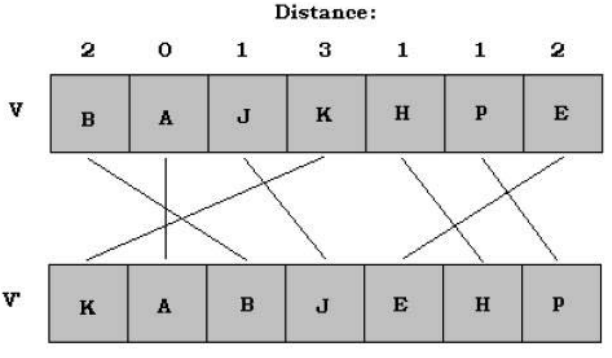

Fig. 1. The distances between the position of each element in V with respect to V'

For the vector V in Figure 1 the disorder can be calculated as:

$$do = (2 + 0 + 1 + 3 + 1 + 1 + 2) = 10 \tag{3}$$

where do_{max} equals:

$$do_{\max} = \frac{(|V|^2 - 1)}{2} = \frac{7^2 - 1}{2} = \frac{48}{2} = 24 \tag{4}$$

In order to normalize the disorder, we perform:

$$do_{nor} = \frac{do}{do_{\max}} = \frac{10}{24} = 0,4167 \tag{5}$$

For a more exhaustive introduction to degree of disorder see [10].

2.3 Typing Paths

Typing paths can be described as a set of key code/key event pairs stored in order of occurrence. If some short sequence of chars is being retyped by a user several times (which is the case with the "Login - Password" mode), the analysis of such paths is likely to show some typical characteristics of a user's behavior:

- moments where keys overlap (second key is pressed before the release of the first one)
- the position of the key pressed in the case of duplicate keys (digits, SHIFT's, etc.)

3 Experimental Setup and Results

In our experiments 18 volunteers participated in testing the proposed keystroke pattern recognition methods. Typing skills varied slightly among them - the majority of the group type on PC keyboard every day. Every volunteer had assigned unique login-id and password. The full name of particular individual was used as her/his login-id, since it is one of the most frequently typed phrase for most of people. In our experiments we calculated standard biometrics recognition parameters, namely False Rejection Rate (FRR) and False Acceptance Rate (FAR) for each of the users. We set the systems for different thresholds: 0.25, 0.3, 0.35 and 0.4.

Table 1. FRR results for the combined feature vector (for all the methods)

user	Combined FRR
user1	7.6923
user2	2.5000
user3	0.0000
user4	41.3043
user5	55.5556
user6	6.2500
user7	41.6667
user8	32.0000
user9	0.0000
user10	15.0000
user11	22.7273
user12	33.3333
user13	36.8421
user14	43.7500
user15	36.3636
user16	9.5238
user17	7.6923
user18	15.7895

Table 2. FAR results for digraphs and trigraphs for the 0.25 threshold

user	Digraph FAR	Trigraph FAR
user1	0.0000	15.3846
user2	0.0000	0.0000
user6	0.0000	17.5439
user8	0.0000	0.0000
user9	0.0000	12.5000
user10	0.0000	1.9231
user14	1.2346	28.3951
user15	0.0000	9.0909
user17	0.0000	0.0000
user18	0.0000	0.0000

Table 3. FAR results for digraphs and trigraphs for the 0.3 threshold

user	Digraph FAR	Trigraph FAR
user1	1.9231	34.6154
user2	0.0000	15.3846
user6	0.0000	47.3684
user8	0.0000	1.6949
user9	0.0000	50.0000
user10	0.0000	7.6923
user14	9.8765	38.2716
user15	0.0000	45.4545
user17	0.0000	0.0000
user18	9.0909	18.1818

In the first stage every participant performed 15 attempts of log in-password authentication that were evaluated by the system in order to calculate the model vector of digraphs and trigraphs as well as to collect the typing paths.

After that users performed several another logon attempts as valid users (FRR tests) and few attempts as impostors (**trying to log on somebody's else account knowing login and password** - FAR tests).

Each user performed 20 logon attempts as valid user. The combined FRR results are presented in Table 1. Unfortunately, usually after several successful attempts most of the users wanted to find out how the system behaves in case of sudden change of typing patterns and they 'test' the system trying to type in extremely different way then they used to. This behavior of users is inevitable in real-life applications and it definitely affected the FRR performance of the system.

In the second part of experiments a participant was asked to act as impostor. She/he was trying to logon on somebody else account. In order to increase the number of logon attacks per single account, we randomly selected 10 out of 18 existing accounts to be attacked. This decision was motivated by the fact that the number of participants (and thus samples) was limited (users were not

Table 4. FAR results for digraphs and trigraphs for the 0.35 threshold

user	Digraph FAR	Trigraph FAR
user1	5.7962	48.0769
user2	7.6923	61.5385
user6	7.0175	66.6667
user8	5.0847	3.3898
user9	12.5000	68.7500
user10	9.6154	19.2308
user14	24.6914	59.2593
user15	0.0000	54.5455
user17	0.0000	0.0000
user18	27.2727	45.4545

Table 5. FAR results for digraphs and trigraphs for the 0.4 threshold

user	Digraph FAR	Trigraph FAR
user1	19.2308	50.0000
user2	46.1538	69.2308
user6	12.2807	71.9298
user8	15.2542	11.8644
user9	18.7500	81.2500
user10	26.9231	36.5385
user14	33.3333	67.9012
user15	0.0000	63.6364
user17	0.0000	10.0000
user18	54.5455	63.6364

willing to spend hours trying to hack somebody's else account). Bigger number of attacks per single account will picture more clearly the FAR, so smaller number of accounts to hack was the only reasonable solution.

The results showing FAR for each of the threshold for digraph and trigraph method are shown in the Tables 2-5. The results for typing path method and for all the methods combined together are shown in the Table 6.

In any web implementation of typing patterns recognition (e.g. password hardening), FAR is more important than FRR and therefore we think our results are satisfactory. Nevertheless some minor changes to our client-server implementation could decrease FRR, which would make the system more user-friendly. It is hard to determine which of the developed and implemented method gives the best performance for all users. The best solution is to make the logon algorithm adaptive. The algorithm should check which method gives the best performance for given user in order to give it the biggest weight while taking the access/no access decision.

In case of non-adaptive implementation the best results were observed for thresholds: 0,25 for trigraphs and 0,3 for digraphs. The threshold for digraphs

Table 6. FAR results for the typing paths method and the final FAR results for all the combined methods

user	Typing Path FAR	Combined FAR
user1	0.0000	**1.9230**
user2	7.6923	**0.0000**
user6	0.0000	**0.0000**
user8	3.3898	**0.0000**
user9	0.0000	**0.0000**
user10	1.9231	**0.0000**
user14	0.0000	**8.1649**
user15	9.0909	**0.0000**
user17	0.0000	**0.0000**
user18	0.0000	**0.0000**

and trigraphs should not be equal. It should be higher for digraphs and lower for trigraphs.

It is also noticeable that longer char sets (trigraphs) have more stable statistics for a legitimate user (the standard deviation of particular trigraph's durations is small, and thus the distance calculated from the degree of disorder is smaller), but on the other hand they are easier to forge.

Typing patterns characteristics are sensitive to the emotional and physical state of the person who is verified. Very poor typing skills are another factor which can affect the process of authentication. The good thing is that our methods of individual typing patterns extraction are very likely to achieve a high level of acceptance among ordinary users.

Moreover, unlike other biometric or security systems, which usually require additional hardware and thus are expensive to implement, typing patterns recognition system is almost for free - the only hardware required is the keyboard [1].

4 Conclusion

In the article we presented and tested methods of recognizing individual typing patterns. We also proved that biometrics system based on such extracted typing patterns is capable of identifying humans and increasing security in web applications where logging-in is the necessity for the clients (e-banking).

The combined values of FRR varied from 0% to 55% (Table 1) and the values of FAR **were equal to %0 for all but 2 users** (Table 6). For the 2 users is was possible for the impostor to logon with their password and biometrics characteristics with the probability 1.9% and 8.2%, respectively.

This means that the presented methods are effective and could be implemented to increase web security in applications where logging-in is the necessity for the clients.

References

1. Monrose, F., Rubin, A.: Keystroke Dynamics as a Biometric for Authentication. Future Generation Computer Systems 16(4), 351–359 (2000)
2. Obaidat, M.S., Sadoun, B.: Keystroke Dynamics Based Authentication. In: Jain, A.K., Bolle, R., Pankanti, S. (eds.) Biometrics: Personal Identification in Networked Society. (1998)
3. Obaidat, M.S., Sadoun, B.: Verification of Computer Users Using Keystroke Dynamics. IEEE Trans. Syst., Man, Cybern.-Part B. 24(2), 261–269 (1997)
4. Leggett, G., Williams, J., Usnick, M.: Dynamic Identity Verification via Keystroke Characteristics. International Journal of Man.-Machine Studies 35(6), 859–870 (1991)
5. Gaines, R., Lisowski, W., Press, S., Shapiro, N.: Authentication by Keystroke Timing: some preliminary results, Rand Report R-256-NSF. Rand Corporation (1980)
6. Monrose, F., Rubin, A.: Authentication via Keystroke Dynamics, Conference on Computer and Communications Security, pp. 48–56 (1997)
7. Joyce, R., Gupta, G.: User authorization based on keystroke latencies. Communications of ACM 33(2), 168–176 (1990)
8. Bleha, S., Slivinsky, C., Hussein, B.: Computer-access security systems using keystroke dynamics. IEEE Trans. on Patt. Anal. Mach. Int. 12(12), 1217–1222 (1990)
9. Brown, M., Rogers, S.J.: User identification via keystroke characteristics of typed names using neural networks. International Journal of Man.-Machine Studies 39, 999–1014 (1993)
10. Bergadano, F., Gunetti, D., Picardi, C.: User Authentication through Keystroke Dynamics. ACM Transactions on Information and System Security 5(4), 367–397 (2002)
11. Brown, M., Rogers, S.J.: Method and apparatus for verification of a computer user's identification, based on keystroke characteristics, Patent Number 5,557,686, U.S. Patent and Trademark Office, Washington, DC (September 1996)
12. Yu, E., Cho, S.: Biometrics-based Password Identity Verification: Some Practical Issues and Solutions, XVth Triennial Congress of the International Ergonomics Association (IEA), Seoul, Korea (August 24-29, 2003)
13. Tapiador, M., Sigüenza, J.A.: Fuzzy Keystroke Biometrics On Web Security. In: Proc. of AutoID (1999)
14. `http://www.biopassword.com`