B2C Internet Authentication Method using Statistical Keystroke Biometrics

Marino Tapiador[1], Juan A. Sigüenza[2]

[1] IBM Global Services and Universidad Autónoma de Madrid,
Escuela Politécnica Superior, Spain

[2] Universidad Autónoma de Madrid,
Escuela Politécnica Superior, Spain

Abstract. This paper is focused on biometric user authentication on the Internet using keystroke dynamics. This work describes the advantages of using keystroke biometrics in B2C Internet applications, and also it shows the problems and techniques related to sampling the typing pattern in the Internet. This research work is presented applied to a real experimental system that implements a keystroke dynamics login mechanism based on interkey and holdkey typing times. Several experiments are described and as a result of these experiments, a statistical pattern-recognition model based on mixed typing times and the average normalization technique is presented in this work. Internet open platform technologies are used.

1 Introduction

In the work presented in this paper, we have developed a system based on behavioral biometrics (see [1]) to provide authentication in a secure Internet application: a business-to-customer (B2C) e-commerce system. The system uses the keystroke dynamics to learn which is the user's behavior when he/she types a sequence of fixed characters in the keyboard. Several works on keystroke dynamics authentication can be found, but not considering the specific characteristics of typing in Internet applications: for instance, the preliminary work of Gaines and Lisowski [2], the password experiments of Monrose and Rubin [3], or the authentication studies of Obaidat and Sadoun [4].

Keyboard is a common hardware that comes in useful to authenticate a user in the Internet because nowadays is the common input device in each terminal in the network. This is a key point for e-commerce systems which main objective is to get a big number of customers i.e. in B2C systems. It also implies no extra-hardware needed, that also increases the number of potential users of the system in the Internet. This kind of biometric control in a web application already has a minimal cost because it only needs new software, not extra-hardware.

Tapiador M. and A. Sigüenza J. (2004).
B2C Internet Authentication Method using Statistical Keystroke Biometrics.
In *Proceedings of the 2nd International Workshop on Security in Information Systems*, pages 73-78
DOI: 10.5220/0002661100730078

2 Method

2.1 Data Collection Mechanism for Typing Samples

A ‘sample’ in keystroke biometrics can be information related to: typing difficulty, interkey times (latencies), holdkey times (durations) or others like the number of keys involved on the character generation, keystroke overlaps, etc. Our work is focused on measuring typing times i.e. latencies and durations. In the Internet the problem is to do this sampling process in a platform-independent manner i.e. independently of the CPU frequency where the user is typing.

The prototype developed in this work is based on machine cycles in order to be able to fetch the holdkey times and get maximum sampling precision. Machine cycles can be measured using low-level programming languages or assembler. However this kind of languages depend on the hardware platform –kind of computer- been used, so that it is not a valid approach for a B2C Internet application. This type of application is going to run in the Internet where a lot of different hardware platforms are connected. This requirement implies to use an open-platform language to capture the keystroke biometric data: Java language. With this high-level language we can measure the machine cycles in an indirect way, using machine pseudocycles using an event-based technique where a counter-thread is continuously increasing the value of a variable i.e. this value is the number of machine cycles multiplied by a constant factor equal to the number of machine instructions involved in the adding loop implemented in Java language.

The machine pseudocycles are also dependent on the computer frequency where the user is typing. The idea is we also need to identify the user independently which computer is been used by him/her. In the Internet the user could be using our application from different terminals at different times. Therefore, in order to assure a pattern-recognition algorithm is able to identify the genuine user between different hardware platforms, the biometric samples need to be normalized to be CPU frequency-independent. In this case, a normalization technique by the average value is used. It means the average time of the typed sequence is used as reference time to normalize all the times in the sequence.

2.2 Statistical Model for Typing Recognition

The prototype is a typical three-tier Internet architecture. In our experiments the keystroke pattern-recognition prototype consist of a database with biometric information that is checked with a server program (CGI in C++) when a user does a login into a simulated web system i.e. an HTML page with a Java applet.

When the user tries to access to the website, the system sends a form (the Java applet) where the user must type his/her userId and password. At the same time the system is registering this information, it is also registering biometric data related to the keystroke dynamics of the user, and this information is sent to the server through Internet. The server receives the biometric data and runs a CGI program with the logic related to the pattern-recognition algorithms to try to identify or authenticate to the

user by his/her typing rhythm. The CGI compares the information with the information stored into a database of files in the hard disk. This files come from the user registration process in the system (training).

A statistical model provides the pattern-recognition mechanism, and it consists of a parameter-based estimation. We can consider each holdkey or interkey time as the result of a random experiment. That result is a normalized -by the average- time value t in $\Omega = (0, +\infty)$. Let be the hypothesis "each user types with some regularity rate", we can expect the time values generated by the user will be concentrated around some average value, and with small deviations from it. These deviations are minimum when the user is very regular. That behavior is well-modeled by a gaussian function i.e. a normal density $N(\mu,\sigma^2)$ with μ average and σ^2 variance. The analytic expression of this function is well known:

$$N(\mu,\sigma^2) \rightarrow f(x) = (1/\sqrt{2\ \sigma^2\pi})\ \exp(\ (-1/2\ \sigma^2) \cdot (x-\mu)^2\ ) \qquad (1)$$

The f(x) corresponds to a normal distribution but we don't know its exact parameters (μ,σ^2). The technique of maximum likelihood shows that the best estimators are:

$$\mu = (1/n)\ \Sigma\ x$$
$$\sigma^2 = (1/n)\ \Sigma\ (x - \mu)^2 \qquad (2)$$

Thus with this technique we get a punctual density estimation following the time distribution for a specific typing. Let be a T_i time which estimated distribution is $N_i(\mu,\sigma^2)$, the probability of getting that T_i is defined by the density $f_i(T_i\ ;\mu,\sigma^2)$. This T_i point probability is used as a scoring function for the T_i point by the system.

During the training phase (ten samples) the user generates the samples composing a template that consist of the sample estimators for the average μ and variance σ^2 for each time interval in the characters sequence. In production time, when a new T_i is generated in a sequence, its probability to occur is calculated using a maximum likelihood estimation by the sample estimators stored into the cited template. Therefore that probability P_i for T_i is obtained based on its distribution estimation $N_i(\mu,\sigma^2)$ calculated during the training phase i.e. $f_i(T_i\ ;\mu,\sigma^2) = P_i$. If P_i is too low it means is highly probability the user has not generated the time, so he/she is a potential fake. The opposite implies the user is probably the genuine user. In order to get this evaluation in a [0,1] rate, it is normalized the density function by its maximum value and it generates this scoring function:

$$S(x) = (1\ /\ S_{max}) \cdot f_i(x;\ \mu,\sigma^2)\ ,$$
$$S(x) = (\sqrt{2\ \sigma^2\pi}) \cdot (1/\sqrt{2\ \sigma^2\pi})\ \exp(\ (-1/2\ \sigma^2) \cdot (x-\mu)^2\ )\ , \qquad (3)$$
$$S(x) = \exp(\ (-1/2\ \sigma^2) \cdot (x-\mu)^2\ )$$

During the training phase the sample estimators included in the statistical template are calculated each time using this procedure:

(1st) Average, sample estimator:

$$\mu_1 = (1 / k) \cdot (n \cdot \mu_0 + X_k), k = n + 1; \ \mu_0 = (1 / n) \Sigma X_i, \ i = 1,2, \ldots n \quad (4)$$

(2nd) Variance, sample estimator:

$$\sigma_1^2 = (\ (1 / (n + 1)) \Sigma X_i^2) - \mu_1^2, i = 1,2, \ldots n + 1;$$
$$\sigma_0^2 = (1 / n) \Sigma (X_i - \mu_0)^2 \ = ((1 / n) \Sigma X_i^2) - \mu_0^2, i = 1,2, \ldots n \quad (5)$$

Where μ_1 is known because it was previously calculated, and the other term can be also calculated due it depends on ΣX_i^2 and occurs that:

$$\sigma_0^2 = (1 / n) \Sigma X_i^2 - \mu_0^2 \rightarrow (\sigma_0^2 + \mu_0^2) \cdot n = \Sigma X_i^2, i = 1,2, \ldots n; \text{ and thus:}$$
$$\sigma_1^2 = (((\sigma_0^2 + \mu_0^2) \cdot n + X_k^2) / k \ ) - \mu_1^2, k = n + 1 \quad (6)$$

So this equations are used to calculate again the sample estimators for the average and the variance each time a new sample is received during the training process, and starting from an initial sample corresponding to the user registration into the system. That first sample is used for the initial averages, and the initial variances are set to zero. This statistical model is used in the same way for interkey times and for holdkey times, just considering the different sequence lengths.

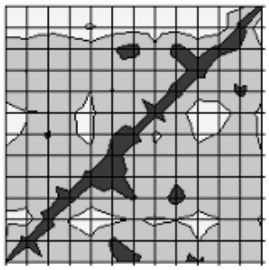
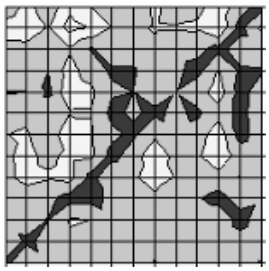
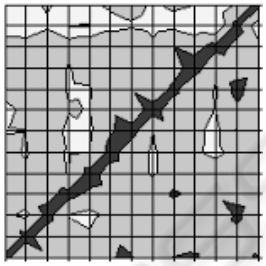
3 Experimental results

In the experiment the samples are composed of interkey times only, or holdkey times only, or both of them i.e. mixed times. The statistical recognition model previously described was used with one simple userId/password sequence ("autonoma" / "internet"). Previous work in this area showed a better recognition rate using simple character sequences (see [5]). The main objective was to compare interkey times versus holdkey times, and mixed times, and to measure the recognition rates. The experiment was performed with a group of men and women with ages between 21 and 48 years old. Netscape and MS-IExplorer v4+ were used under Windows platform.

The Table 1 resumes the results of the experiment and allows comparing the performance of using the three time approaches. The table shows that comparatively the recognition performance is better for mixed times. The figures in Table 1 show the real user typing versus all the possible users included him/her, therefore each figure should present a 'diagonal effect' i.e. a diagonal of maximum score values: diagonal points are where the real user is the supposed user. The best performance is associated to the mixed times figure in the sense of it has a more clearly defined 'diagonal effect'.

Table 1. Experiment results of the three kind of times.

■ 0,9-1,2 ■ 0,6-0,9 □ 0,3-0,6 □ 0-0,3 = Scoring. (X-axis: real user. Y-axis: supposed user.)

Statistical Model		
Interkey times	Holdkey times	Mixed times
		

Thus, considering the winner mixed times only, the Table 2 illustrates with more detail the scoring rates (%) achieved by this recognition model. The real users typing are represented by the 'X' items and the supposed users or templates used are the 'Y' items. The 'diagonal effect' can be clearly observed and only one error is presented in cell Y12-X9 i.e. a case where the maximum recognition value is obtained with a user different to the genuine user.

Table 2. Experiment results of mixed times.

%	***Y1***	***Y2***	***Y3***	***Y4***	***Y5***	***Y6***	***Y7***	***Y8***	***Y9***	***Y10***	***Y11***	***Y12***	***Y13***
X1	62	38	53	46	47	55	56	37	56	58	47	54	56
X2	41	70	52	46	44	62	57	47	54	49	49	57	48
X3	35	30	55	33	28	37	39	34	40	33	36	53	37
X4	52	48	53	65	56	54	58	46	61	53	56	54	52
X5	48	44	49	41	71	57	53	40	52	48	53	46	47
X6	52	47	59	45	47	80	61	44	61	52	41	58	53
X7	42	42	65	34	39	62	71	42	57	48	36	65	47
X8	36	43	44	42	39	57	48	66	51	46	39	56	42
X9	46	42	62	38	39	62	61	49	66	51	39	69	50
X10	53	47	62	40	51	61	59	45	57	70	46	54	56
X11	54	52	54	51	59	56	61	46	62	56	69	57	46
X12	30	27	42	22	21	39	43	37	39	33	25	76	32
X13	36	31	39	28	34	39	43	35	39	37	23	40	62

4 Conclusions

Considering the final results resumed in Table 1, the accuracy obtained is 99% for mixed times versus a 97% for interkey times and a 94% for holdkey times. Thus, it is observed that mixed times work better than each other separately. In terms of False Accept Rate (FAR) and False Rejection Rate (FRR) the system showed a 0.6% of FAR and a 7% of FRR (for details on FRR/FAR see the work of Wayman [6]).

The experiments suggest that keystroke biometric devices can be developed for the Internet using the average normalization technique to separate the typing samples from the computer platform used to generate them. These keystroke biometric devices can use open-platform tools like Java Applets or CGI programs. Multi-thread Java programming techniques can already be used to capture the user typing rhythm without intrusive impact i.e. not using 'heavy' components as e.g. ActiveX which need special security permissions in order to access to low-level features of the machine. Nowadays, the systems alerts to the user in order to close other applications running during the login process in the website because the program could get varying amounts of processor time, due the counts depend heavily on the other processes on the machine. Future work could be focused on solving this issue.

The presented system is good for the B2C e-commerce model where we want to reach a broad market i.e. it is good because it is a way to increase the security related to the authentication process without requiring special biometric hardware devices like fingerprint readers and so. The prototypes developed in our work have a low cost in hardware and software in comparison to other traditional biometric devices (fingerprint, iris-scan, etc.) and 'brute force' attacks are useless against them because they should generate also the interkey/holdkey times in each password typing (sample).

References

1. Bolle,R., Jain,A., Pankanti,S. *Biometrics. Personal Identification in Networked Society.* Kluwer Academic Publishers. (1999)
2. Gaines,R., Lisowski,W., Press,S., Shapiro,N. Authentication by keystroke timing: some preliminary results. In *Rand Report,* R-256, NSF, Rand Corp., Santa Monica, CA. (1980)
3. Monrose,F., Rubin,A.D., Keystroke dynamics as a biometric for authentication. *Elsevier Science Direct, 16, (no.4):351-359*. (2000)
4. Obaidat,M.S., Sadoun,B. Verification of computer users using keystroke dynamics. *IEEE Trans. on Systems, Man and Cybernetics. Vol.27, No.2*, 261-269. (1997)
5. Tapiador,M., Sigüenza,J. Fuzzy Keystroke Biometrics on web security. In *AutoID'99 Proceedings: workshop on Automated Identification Advances Technologies*, Summit, New Jersey, USA, October 1999, IEEE Robotics and Automation Society, 133-136. (1999)
6. Wayman, J.L. Technical Testing and Evaluation of Biometric Identification Devices. In *Biometrics. Personal Identification in Networked Society.* 345-368. Kluwer Academic Publishers. (1999)