

Keystroke Dynamics – stepping forward in authentication

Sérgio Tenreiro de Magalhães¹, Kenneth Revett² and Henrique M. D. Santos¹

¹ Universidade do Minho

Department of Information Systems

Campus de Azurem

4800-058 Guimaraes, Portugal

{psmagalhaes, hsantos} @dsi.uminho.pt

² University of Westminster

Harrow School of Computer Science

London, UK HA1 3TP

revettk@westminster.ac.uk

Abstract

Authenticating a user is a complex problem in which we depend to provide some level of trust to the Security Proceedings of the Information System and to the Information System itself. Biometric technologies have shown that they are able to provide high rates of accuracy but they also raise many ethical issues. On the other side of the problem, many times authentication must be made without the collaboration of the one being authenticated. Keystroke Dynamics is a biometrical technology that can be used in stealth mode and that doesn't present any issues to what relates to being intrusive, when used in collaborative mode. Many results have been published but there isn't any integrative technologies yet.

1. Introduction

The use of biometric technologies to increase the security of a system has become a widely discussed subject. While governments and corporations are pressing for a wider integration of these technologies with common security systems (like passports or identity cards), human rights associations are concerned with the ethical and social implications of its use. This situation creates a challenge to find biometric algorithms that are less intrusive, easier to use and more accurate.

The precision of a biometric technology is measured by its False Acceptance Rate (FAR), that measures the permeability of the algorithm to attacks, by its False Rejection Rate (FRR), that measures the resistance of the algorithm to accept a legitimate user, and by its Crossover Error Rate (CER), the point of interception of the FAR curve with the FRR curve that indicates the level of usability of the technology. For a biometric technology to be usable on a stand-alone base, its CER must be under 1%. As an algorithm gets more demanding, its FAR gets lower and its FRR gets higher; usually the administrator of the system can define a threshold and decide what will be the average FAR and FRR of the applied algorithm, according to the need for security – dependent of the risk evaluation and of the value of what is protected; also the threshold can be, in theory, defined by an Intrusion Detection System (software designed to identify situations of attack to the system).

Establishing the error rates of a biometric technology is a complex problem. Studies have been made to normalize their evaluation, but the fact is that the results are strongly dependent of the number of individuals involved in the process and, what is worst, of who is chosen. This means that, even with a large amount of data collected, the results can be very different if we change the group evaluated. This happens because it's very difficult to obtain a sample representative of the population, once we do not know how to characterize the population. A good example of this disparity are the results of the Fingerprint Verification Competition (FVC) 2004, where the best CER

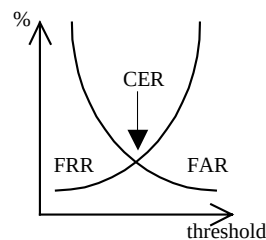


Fig.1 - CER

achieved had a value of 2,07% [Maio, 2004], compared with the results of the FVC 2002, where the best CER achieved was 0,19% [Maltoni et al, 2003]. Some international companies were present in both contests and the only justification for the disparity of the results is the difference in the sample data used to test the algorithms. All things considered, the only way to evaluate biometric algorithms is comparing them using the same data and, then, we can say that this technology is more accurate than that. The results also vary according to the use: a system used to identify an individual is less accurate than a system used to authenticate it (in this case the user presents himself to the system and it has only to confirm his identity)

Biometric technologies are usually classified as behavioral (e.g. voice recognition) or physical (e.g. retinal recognition), according to the classification of the characteristics evaluated. But they can also be classified as collaborative, if they demand that the user knows of its existence and participate in the process, or as stealth technologies, if they can be used without the knowledge of the one that is being authenticated or identified [Magalhães, 2003].

Keystroke dynamics is a behavioral biometric technology that can be used with the collaboration of the user or in stealth mode and that allows an increased level of security, both in authentication and in identification, almost without any extra effort once it works by analyzing the patterns existing on the way that the user types (a password, a passphrase or general text) on a keyboard. These algorithms can adjust their parameters to adapt themselves to evolutions on the typing patterns of the user.

2. Keystroke Dynamics – previous work

As in many other problems, there have been two different approaches to the challenge of finding an algorithm for keystroke dynamics that minimizes the CER: machine-learning and deterministic algorithms.

Among the solutions based on the machine learning we can find the work presented by Chen [3] which achieved a cross-over error rate (CER) less than 1% and a 0% false acceptance rate (FAR). Ord and Furnell [4] also tested this technology, with a 14 people group, to study the viability of applying it to the simple use of PINs (Personal Identification Numbers) typed on a numeric-pad. Unfortunately the results suggest that, for a large-scale use, the technology is not feasible.

Deterministic algorithms have been applied to keystroke dynamics since the late 70's. In 1980 Gaines [5] presented a report of his work to study the typing patterns of seven professional typists. The small number of volunteers and the fact that the algorithm is deduced from their data and not tested in other people later, results in a lower confidence on the FAR and FRR values presented. But the method used to establish a pattern was a breakthrough: a study of the time spent to type the same two letters (digraph), when together in the text. Since then, many algorithms based on Algebra and on Probability and Statistics have been presented. Joyce Gupta presented in 1990 [6] an algorithm to calculate a value that represents the distance between acquired keystroke latency times and correspondent times previously stored. In 1997 Monroe and Rubin use the Euclidean Distance and probabilistic calculations based on the assumption that the latency times for one-digraph exhibits a Normal Distribution [7]. Later, in 2000, they also present an algorithm for identification, based on the similarity models of Bayes [8], and in 2001 they present an algorithm that uses polynomials and vector spaces to generate complex passwords from a simple one, using the keystroke pattern [9].

The algorithms cited are a small example of the many approaches used to find adequate keystroke dynamics algorithms with a convenient CER. Many others could also be referred, all with different evaluation methods, different number of users involved (usually a limited number of users), different number of keystrokes required to enroll the system and different number of repetitive operations required to authenticate and/or identify the user. This diversity in the algorithm parameters and in the evaluation method makes the task of comparing their results a very difficult one. Furthermore, there is, in this subject, no concept of what is a representative data sample. The same algorithm presents different results when tested with different volunteer groups. The only way to compare two algorithms is to test it against the same group.

Envisaging wide scale applications, like web-based applications (where this method is not executable now) one must consider the results only if the test user group's size is considerably large. In this application domain one must remember that the computational effort necessary to execute the algorithm is a critical factor.

Nevertheless, and for the record, according to Peacock [10] regarding keystroke dynamics, they reported FAR values from 0% to over 50%, the FRR varies from more than 25% to less than 1% and the numbers of users involved is usually between ten and one hundred.

3. An lightweight algorithm

The algorithm presented in [2] is lightweight both on the enrolment and on the authentication stage. The enrolment process, made by the user once on the first use of the service, consists on typing the users usual password, or passphrase, twelve times. The data is stored and the average, the median and the standard deviation of the times for each digraph is calculated and stored along with the average, the median and the standard deviation for the total time spent on each password/passphrase.

The authentication process is, in the user's point of view, equal to any password method. He/she only has to introduce his/her password like he/she usually does. For each keystroke the algorithm will measure the time latency, defined as *TLP*, and compare it with the one stored. The comparison result will be a hit if and only if

$$\text{Lowest}(\text{Average}; \text{median}) * \left(0,95 - \frac{SD_{\text{deviation}}}{\text{Average}}\right) \leq TLP \text{ and}$$

$$TLP \leq \text{Higher}(\text{Average}; \text{median}) * \left(1,05 + \frac{SD_{\text{deviation}}}{\text{Average}}\right).$$

This same calculation is repeated for all the password/passphrase keystrokes, and the results are stored in a Boolean array.

Then a sum *A* is calculated. The not hit values do not contribute to the sum *A*. The value 1 is added if the previous value is not a hit (or if it is the first value) and a value 1,5 is added if the previous value is a hit. The final value of *A* will decide if the authentication process succeeded, or not, according to the threshold defined by the system administrator. For instance, if the threshold is set on 70%, users will only be authenticated to the system if the value *A* obtained from a given attempt is over 70% of the highest possible value, which is given by: $(\text{number\_of\_characters} - 1) * 1.5 + 1$. Finally, if and only if, the login attempt is accepted, the oldest values stored for the latencies are substituted by the corresponding values collected in this successful attempt. This last procedure will allow the data stored to evolve with the user.

The accuracy of the biometric systems is measured by the False Acceptance Rate – FAR, that measures the percentage of illegitimate attempts to login that are successful – and by the False Rejection Rate – FRR, that measures the percentage of legitimate user's denied login attempts. The Crossover Error Rate, or Equal Error Rate, is the value of these rates when they are equal, for a determinate threshold (figure 1). The lowest the FAR, more accurate is the technology.

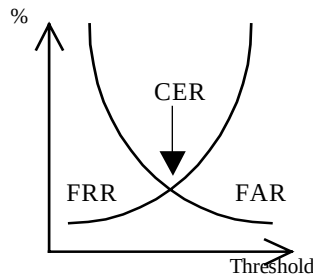


Figure 1 – The Crossover Error Rate (CER) is obtained at the point where FAR=FRR.

This algorithm presented a CER of 5,58% and it can achieve, at the lowest thresholds, a FRR of near zero that maximizes the comfort of the user. At the higher demanding thresholds the algorithm presents a near zero FAR, maximizing the security. The several obtainable rates can be found in Figure 2.

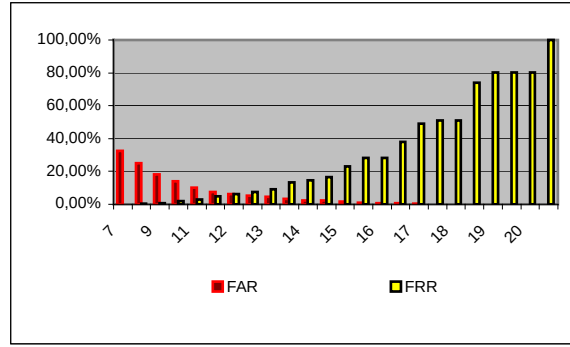


Figure 2 – False Acceptance Rates and False Rejection Rates for the several possible thresholds

4. The performance factors

Revett and Khan [11] concluded that adding keyboard partitioning reduces the impostor success rate (FAR). According to those results, access codes that contained letters from each of the partitions of Figure 3, in a manner that forced the user to enter characters that were scattered across the keyboard, provide more accuracy to keystroke dynamics systems. Furthermore, the speed at which we enter our access codes may actually compromise our access code protection mechanism. According to those results, the best results are achieved when the user does not type at his maximum typing speed.

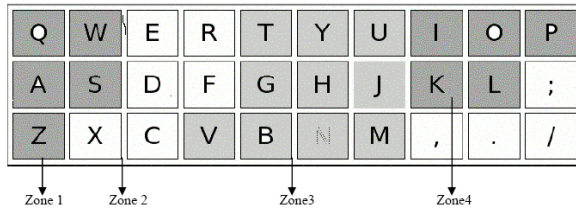


Figure 3 – The keyboard partition into divided four disjoint zones proposed by Revett and Khan [11]

But to impose a password or a typing speed to the user decreases the comfort level and in Web-based applications to lose 1% of the users can mean losing some thousands. The large-scale factor increases the importance of keeping the system comfortable, non-intrusive and with a low FRR. Therefore the keystroke dynamics algorithms must find a way to include user's performance factors in the decision process, instead of forcing users to change their everyday patterns.

5. An altered algorithm that scores the time latencies

The need to include in the decision process the user's performance factors (the same factors for everyone but maybe with different values for the corresponding variables used) generates the need to score the time latencies. In [11] an alteration was proposed to do that by multiplying the vector coordinates used to generate the sum A in the algorithm described in section four by several variables that must be adjusted for each user if we intend to maximize the accuracy. Unfortunately, we can not assume that the Web users are advanced technology users and, therefore, in this context we must instantiate the variables and find constants, k , that will maximize not each user's CER but the

general Web community's CER. So, now $A = \sum_{i=1}^n (k_1 k_2 \dots k_p f(t_n))$, where n is the number of time latencies

resultant from the password string, p is the number of performance factors included in the decision process and $f(t_n)$ is the result of applying the previously described algorithm to the n^{th} time latency.

The work presented in the paper “Enhancing login security using keystroke hardening and keyboard gridding” [12] suggested that a good factor to start with could be the average typing speed and the empirical data showed that its weighing factor is $k_{speed} = 2$. Intuition says that users that are right-handed would be more systematic when changing from the left hand to the right hand and the cited work [12] indicated that the characters in contiguous areas (Figure 3) tend to have more stable time latencies. So we decided to give more credit to time latencies that resulted from typing two characters in contiguous zones going from left to right. Also in this case the best constant has the value 2 (two). The result, as showed in the next section, was a reduction on the CER.

6. The Rough Sets approach

Revett et al [13] used rough sets to mine a small database of keystroke based biometric data – using only digraph times. The purpose was to develop an approach to developing a situated agent that could be used to determine whether a login attempt was legitimate or not. Using a reasonable sized dataset, we generated a decision table by including the correct decision class (legitimate or non-legitimate owner). That methodology based on rough sets was able to predict with a high degree of accuracy whether the attempt was legitimate or not based on the decision rules that we generated from rough sets (97% or more classification accuracy). The most interesting result indicated that the digraph times and specific digraphs were sufficient to determine whether a user was legitimate. In addition to typing speed, Revett concluded that probably, using only a subset of time latencies is enough – depending on the particular login ID characteristics of the owner. This reduction in the number of attributes that must be stored and searched through reduces the computational load of the verification system. The use of rules generated from rough sets based classifiers can be enhanced by the addition of more attributes into the decision table.

7. Conclusions

The results published on keystroke dynamics show that there is a need for the integration of the several results obtained and that there is a need for an integrating software and consequent test of performance.

Anyway, it is now clear that this technology can be used with success.

10. References

- [1] – Yan, J., Blackwell, A.F., Anderson, R. & Grant, A. , 2004, Password memorability and security: Empirical results, *IEEE Security and Privacy* 2(5), 25-31.
- [2] – Magalhães, S. T. and Santos, H. D., 2005, An improved statistical keystroke dynamics algorithm, *Proceedings of the IADIS MCCSIS 2005*.
- [3] – Chen, Z., 2000. *Java Card Technology for Smart Cards*. Addison Wesley, U.S.A.
- [4] – Ord, T. and Furnell, S. M., 2000. User authentication for keypad-based devices using keystroke analysis. *Proceedings of the Second International Network Conference – INC 2000*. Plymouth, U.K.
- [5] – Gaines, R. et al, 1980. Authentication by keystroke timing: Some preliminary results. Rand Report R-256-NSF. Rand
- [6] - Joyce, R. and Gupta, G., 1990. Identity authorization based on keystroke latencies. *Communications of the ACM*. Vol. 33(2), pp 168-176.
- [7] – Monrose, F. et al, 2001. Password Hardening based on Keystroke Dynamics. *International Journal of Information Security*.
- [8] – Monrose, F. and Rubin, A. D., 1997. Authentication via Keystroke Dynamics. *Proceedings of the Fourth ACM Conference on Computer and Communication Security*. Zurich, Switzerland.
- [9] – Monrose, F. and Rubin, A. D., 2000. Keystroke Dynamics as a Biometric for Authentication. *Future Generation Computing Systems (FGCS) Journal: Security on the Web*.
- [10] – Peacock, A. et al, 2004: Typing Patterns: A Key to User Identification. *IEEE Security and Privacy*. September/October

- [11] – Magalhães, Revett and Santos, 2005: *Password Secured Sites - Stepping Forward With Keystroke Dynamics*, Proceedings of the IEEE International Conference on Next Generation Web Services Practices, IEEE CS Press,.
- [12] – Revett, K. and Khan, A., 2005, Enhancing login security using keystroke hardening and keyboard gridding, *Proceedings of the IADIS MCCSIS 2005*.
- [13] - Revett, Magalhães and Santos, 2005: *Developing a Keystroke Dynamics Based Agent Using Rough Sets*, Proceedings of the Workshop on Rough Sets and Soft Computing in Intelligent Agent and Web Technology , Compiègne University of Technology, Compiègne, France.