Automated Generation of Benchmark Data for Keystroke Dynamics Based Authentication Schemes

Tashreen Shaikh Jamaluddin and Kenneth Revett
Faculty of Informatics and Computer Science
The British University in Egypt
El Sherouk City, Cairo, Egypt
ken.revett@bue.edu.eg

Abstract

Keystroke dynamics is a type of behavioral biometric that relies on the typing style of the person. Features of typing such as dwell time and digraph times are used to build a profile of a person's typing style, which is considered to be unique and difficult to forge. In order to evaluate the uniqueness of typing styles, large scale studies involving potentially thousands of subjects is required. Running such large studies is very difficult due to logistic and other reasons. This work proposes the development of an automated typing system that reproduces salient features of human typing such as speed, error rate, and natural variation in typing. Such a system can generate 10,000s of typing samples per hour, making large scale testing of various keystroke dynamics based approaches to biometrics feasible.

Keywords: *automated typing system; behavioral biometrics; biometrics; keystroke dynamics; keystroke modelling*

1. Introduction

The field of biometrics has broadened in scope from purely anatomical measurements such as fingerprints and iris scans for the purposes of person identification/authentication. Today, biometrics encompasses a range of non-anatomical modalities which can be broadly classified under the terms "behavioral" and "cognitive" biometrics. This paper focuses on a particular instance of a behavioral biometric based on keystroke dynamics. Keystroke dynamics is an approach to user authentication/identification that utilises the dynamics of keystroke input. For instance, when a person presses a key on a standard PC keyboard, a code representing the particular key pressed is stored into the keyboard buffer. Typically within a PC/Windows/Unix environment, each key is encoded using an ASCII/UNICODE encoding scheme which serves to uniquely identify each key available on the keyboard. Further, it should be noted that there are two actions which are generated whenever a person types: a KeyDown and KeyUp event. Keyboard sniffers use this knowledge in order to extract passwords and related information. In the context of biometrics, these events can be used, in conjunction with an accurate timer (millisecond accuracy), to build a typing profile. More specifically, every time a keypress event is generated (pressed and/or released), a timer is activated which records when the event occurred.

From this simple scheme, features such as the dwell time can be obtained, which is a measure of the time spent depressing a key (KeyUp – KeyDown time). Further, derivative features such as the time between pressing successive keys can be recorded – the digraph time. This concept can be extended to include trigraphs as well (time between 3 separate key presses). These features form the basis of typing dynamics – the foundation of keystroke dynamics based biometrics. The critical issue in this approach is the uniqueness and reproducibility of typing. If two people are asked to type in the same sequence of characters, will they do so with the same temporal dynamics? Research has indicated that people type in fairly unique ways – with their own distinguishable characteristics. This question can certainly be addressed empirically; though this requires large samples when estimating uniqueness (reproducibility is not addressed in this work directly). This common sense notion was understood during the days of telegraph systems deploying Morse code in the middle of the 19^{th} century [1]. With this background in mind, the heart of the matter in this paper can be addressed: developing an automated method of generating text samples that can be used as benchmark data for evaluating the efficacy of various algorithms deployed in keystroke dynamics based authentication applications.

Forsen was one of the first investigators to rigorously evaluate a wide range of potential biometric features such as the electrocardiogram (ECG), fingerprints, and keystroke dynamics to name a few [2]. Since his pioneering studies, hundreds of papers on the subject of keystroke dynamics have been published, all claiming a variety of levels of accuracies for a given set of inputs (subjects) [3]-[8]. As with virtually all biometric studies, users were asked to enroll into the system by entering a login ID and password multiple times (typically 10). Keystroke dynamics based data such as dwell times and digraph times are acquired from these samples and a model of each user is generated and stored in a database for use during the authentication phase, where the same features extracted during a login attempt are compared to the reference model for that login ID/password. A common feature in these studies is the relatively small sample size – typical studies only deploy 5-20 subjects. This is a problem in that if one is attempting to develop a deployable biometric system – it must be tested on a significant fraction of the population before one can make claims of efficacy with a significant degree of certainty. The problem is the lack of subjects typically deployed in these experiments - it is not a trivial task to perform such a study with 1,000+ subjects. The purpose of this work is to develop a system that will generate typing samples programmatically in order to provide an ad libitum supply. In the next section, we describe the requirements for such a system, then move onto our implementation, followed by some preliminary results, and finally a conclusion summarising our findings.

2. System Requirements

The purpose of this work is to create typing samples of variable length – from a single character to an encyclopedia – as dictated by the needs of the researcher. The typing samples will consist of scan codes that are to be time stamped with 1 ms accuracy. Further, the typing samples should emulate human typing as much as possible - as opposed to random typing. One approach is to attempt to simulate a typist, which has a long and venerable history in the field of operational psychology. A number of attempts have been made to replicate or simulate the human hand in the act of typing (transcriptional), attempting to reproduce a series of observations acquired during recorded typing sessions from a physiological and anatomical perspective. Note that the literature has focused on transcriptional typing – in the case at hand – in the context of biometrics, the task strictly speaking is not transcriptional, as the typist is required only to type their login details which are stored (hopefully – not copied from a sticky note in their desk drawer!) in long term memory. This distinction should not be a deterrent, as one can envision a simple model for both scenarios that have the following generic form:

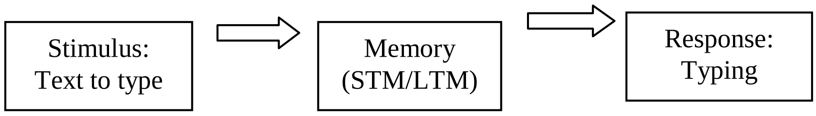

Figure 1. A typical typing processing pipeline with a perceptual ('Stimulus' component), cognitive ('memory' component), and a motoric component, arranged in series.

One can make the simplifying assumption that the difference between transcriptional typing and entering in one's security details differ only in the source of the stimulus (external versus internal respectively) – then there is a considerable degree of overlap between the two processes. The aim of this work is not to create a cognitively complete and accurate model of the typing phenomenon per se – but merely to derive a phenomenological model. Therefore, utilizing the literature on transcriptional typing is simply an economical approach to solving the problem at hand.

One of the earliest published works in this field was by Card and colleagues in the early 1980's [9]. Their approach was captured in what they termed the Keystroke level Model (KLM) of typing, which examined typing quantitatively in terms of speed and errors as a function of the typing task. The context of this research was on the Human Computer Interface (HCI) – and how the design of the applications influenced the effectiveness of the HCI. Rumelhart and Norman published a seminal paper in this field in 1982, focusing on a more cognitive/perceptual approach to typing in order to account for variations in typing speed and errors between individuals [10]. This approach added the connectionist perspective – culminating in what these authors called the Activation Triggered Schema system (ATS). This system provided additional insights over the KLM approach – by quantitatively accounting for specific errors such as transpositional and doubling errors which the former did not formally address. The Model Human Processor (MHP) model was presented to the literature in 1983 by Card and colleagues and has become the standard model for human typing [11]. The MHP approach is another HCI domain based approach that deployed an engineering perspective that attempted to make absolute predictions about human capabilities with regards to physical interactions with devices of all sorts. They proposed a series of three processors (perceptual, cognitive, and motor) with a series of working and long term memories, linked appropriately and tuned to provide a simulation of human typing. Like its predecessors, it accounted for some but not all of the observations concerning human typing. Bonnie E. John published another model called TYPIST (TheorY of Performance of the Model Human Processor) [12]. This is an extension of the MHP – but is a purely phenomenological approach that attempted to account for variations in typing proficiency (novice, intermediate, and expert) using a three state model (perceptual, cognitive and motor) similar to the MHP approach. The TYPIST approach accounted for many of the same phenomenon reported by more 'sophisticated' models such as the MHP and ATS systems, but with minimal cost from a computational perspective. The most recent approach is an extension of the MHP approach, termed the Queuing network model of transcription typing (QN-MHP), first proposed by Liu and colleagues in 2006 [13]. Briefly, this cognitive modelling approach combines two different approaches: a queuing network (QN) approach with a cognitive modelling based approach such as the MHP/GOMS. The approach provides a computational model of concurrent processes that focus on cognitive activities – of which typing is but an example. The system design allows modularization of the cognitive model, which can be replaced and integrated into the transport model provided by the network queuing component.

These models of cognitive processing have provided a wealth of information regarding how humans interact with physical devices such as pointing devices, typewriters and keyboards. The purpose of these models was to develop a working model which facilitates our understanding of the limits of human performance skills in the context of human-computer interaction (HCI). In the current context, a series of 34 'robust phenomenon' (observations) have been generated that provide quantitative constraints that should be adhered to when developing a model of human typing (transcriptional). The interested reader should consult [12], [13] for a detailed list of the robust phenomenon in transcription typing. In the present work, no attempt has been made to incorporate all 34 observations – as our goal is to create a simplified phenomenological model of human typing that generates short strings of text in a parametised fashion (it has serendipitously accounted for 12 of them). No attempt has been made to integrate typing skills into the larger context of human performance skills. Therefore, factors such as units of typing (copy span etc.) are not incorporated into our model. The model we present in this paper accounts for typing speed and error rates as a first approximation to human typing. The next section describes our model in detail.

3. Typing Model

In the context of static keystroke dynamics based user authentication, a user is typically required to enter their login ID and password as a means of authentication. Other systems may require that a passphrase is entered, which may entail entering a longer text string (typically several hundred characters). In continuous authentication schemes, users may be queried while they are in the middle of creating a text document for instance, which is more likely to contain a transcriptional element. The obvious common denominator is typing – and as such the model described here will work in all the above mentioned scenarios. It will be left to future work to explore the continuous authentication mode that entails more of a transcriptional character explicitly. The following assumptions were integrated into our typing model:

1. typing will be performed using a standard 101-key PC style QWERTY keyboard
2. the typist has full use of both hands
3. classes of typing errors are treated equally
4. consistency is inversely proportional to the typing speed.
5. Variance for digraph times is linear – follows no distribution

The model is parameterized as much as possible to allow for a range of typing skills to be simulated, such as beginner, intermediate, and expert. This allows different labs/experiments to utilise the same typing samples. Any differences in results in terms of accuracy in the context of keystroke dynamics based biometrics (i.e. reported EER values) will be independent of the typing samples used in the study. Typically, different experiments use different text inputs, different classification strategies, and different classification algorithms. We wish to remove the input data as a variable – limiting its influence on the results. To this end, we wish to produce a generic model of typing, which will simulate a wide range of typing skills – as the premise for this system is to reproduce *any* potential text input from any user. As a first approximation, factors such as typing speed and error rate must be included and accounted for in the model.

The parameters deployed in the model are parametised typically by a single parameter. Typically, whenever possible, the values for parameters are acquired empirically from our own analysis or from published literature data. Any study that uses this system can produce the same typing samples. The cost is that corpi produced will be adhering to the same underlying assumptions. The basic model can be described by the following equation (1):

$$\Theta *(\alpha*DG \ +/- \ \beta*CV) + \gamma \qquad (1)$$

where DG is the mean value in ms for a digraph, CV is the Coefficient of Variation for the digraph, Θ is a parameter that takes into account the type of digraph (explained further in the text), α determines the typing speed/skill level (beginner, intermediate, or expert), β reflects the consistency of the typing, and γ is the error rate. The simple formulation in (1) utilises input from a file containing the characters to be typed including all shift and control characters. The value of the digraphs (DG in (1)) is critical and can be obtained in two principle approaches: empirical and seeded. In the empirical formulation, one can collect data from a small group of typists. This was the approach taken in this study: two average typists (31 and 33 gwpm) were asked to type a corpus of text (6,000 characters) containing a wide variety of digraphs. The text was obtained from a page of writing downloaded from the internet, in English, containing content appropriate for a college level student to comprehend. The text contained a variety of digraphs with frequencies ranging from 2 to 36. Both typists entered the same text on the same PC (and 101-PC style keyboard) at the same time of day and under the same conditions. The digraphs were extracted from the typing sample using software that was developed by the authors. The digraphs were pooled and statistics such as the mean (reported as the DG in (1)) and the CV (used in (1)) were calculated. In addition, the histogram for each digraph was computed in order to determine if there were any regularities in their form. For instance, do the digraphs follow a linear or Gaussian distribution [14]. Are there differences between certain digraphs in terms of their distribution – a tight versus loose Gaussian etc? In order to make the system more general and practical, we did not insist that each digraph contained equal sample numbers, nor did we insist upon a threshold number of digraphs, although these issues will be explored in further work. The values for the parameters must next be determined in order to start the system. The model assumes typing speed is dominated by the digraph times (DG). Further, typing speed was considered to be categorical, in that we have novice typists (< 30 gwpm), intermediate typists (31-59 gwpm), and expert typists (>= 60 gwpm). Note that 'gwpm' standards for 'gross words per minute,' which does not take into account any errors which may occur. This 'typist' classification scheme provides discrete values for the 'α' term deployed in (1). The values for all parameters are presented in Table 1.

Table 1. Value of all 4 major parameters used in the model for each typist category.

Novice	$\alpha = 1$	$\beta = 1$	$\gamma = 7\%$	$\varepsilon = 1$ $\Theta = 1.0$
Intermediate	$\alpha = 0.75$	$\beta = 0.75$	$\gamma = 3\%$	$\varepsilon = 2$ $\Theta = 0.90$
Expert	$\alpha = 0.5$	$\beta = 0.5$	$\gamma = 1\%$	$\varepsilon = 3$ $\Theta = 0.80$

These values reflect the average differences in typing speed for the typing speed categories – and we placed an upper bound of 100 gwpm for the expert typist, though the Guinness Book of World records has recorded speeds in excess of 200 wpm [15]! As a quality control measure, it should be noted that the sum of the DG times should be approximately equal to the typing speed. The values given above adhere to this quality control measure quite well. The CV term in (1) reflects the consistency of typing: it is an assumption that consistency is inversely proportional to the typing speed. The values for CV was extracted from the data (typing samples from the two subjects), and for this study this was used for the intermediate typists. The parameter 'β' (1) is used to modulate the CV and used was instantiated to values in the same fashion used for calculating the 'α' term in (1). This approach does not apply any distribution on top of the data – only the spread around the mean is taken into account in this formulation. Clearly, a distribution could be incorporated into the model for this parameter, which is an area of future work. The error rate, the 'γ' in (1) reflects the number of errors produced by the typist per minute of typing time, and is assumed to be independent of the content of the text being typed. The types of errors are not distinguished – that is transposition, substitution, omission, doubling, and related errors are not distinguished in the model. The error rate is therefore the sum of all possible errors that can be made (see [16]). The scheme deployed for error rates are presented in Table 1. Note that the error rates are percentages, so the actual number of errors will depend on the gwpm of the typist. Errors are generated randomly based on the value of 'γ' and typing speed during the course of typing. Further, the type of error inserted into the text is generated from a list of typical types of typing errors, which include transpositional, intrusion, omission, and substitution errors, which are assigned randomly to the current instance of a typing error. The last term in the model, 'ε' determines the type of digraph that is entered. There are several types of digraphs that can be deployed when two successive keys are typed. For instance, a single fingered typist will generate a KeyDown and KeyUp event for every character they enter. When typing two successive keys, there is a single digraph path they generate: KeyDown -> KeyUp -> KeyDown ->KeyUp. Note that the KeyDown events are used as the timer markers when recording digraph times throughout this work, though this could be parametised as well if KeyUp times were required. The intermediate or expert typist will typically deploy two hands – certainly more than one finger when typing. This opens up additional digraph paths that must be taken into account. This is the purpose of the 'ε' parameter in (1). In this study, we assume that there are 3 possible digraph paths, as depicted in Table 2. The probability that a particular digraph path is selected depends on both the experience and skill of the typist and the characters in the digraphs. The values for 'ε' are specified in table 1 are arbitrary. As previously mentioned, novice typists might simply use the hunt-and-peck approach, which would yield an 'ε' = 1. Likewise, more skilled typists would use multiple fingers (typically on both hands), and therefore the value of 'ε'= 2 or 3. In addition, intermediate typists may switch between hunt-and-peck and two-handed typing. The deciding factor is the position of the characters in the digraph on the keyboard (assuming a typical QWERTY style keyboard in this discussion). If two characters in a digraph are close together, typically on the same side of the keyboard, then they will tend to be typed using one hand (either 'ε' = 1 or 2), and the deciding factor is the skill of the typist (novice will use an epsilon of 1 and an intermediate or expert 2). If the characters are geographically remote – on each hemisphere (left/right halves) of the keyboard, then they can be typed with both hands simultaneously (either 'ε' = 2 or 3). The two-handed approach tends to yield shorter digraph times (at least faster typing times than single handed typing). The distinction between 'ε' values 2 and 3 takes this into account. A value of 2 for 'ε' indicates that the digraphs are to be typed single handed (they are on the same half of the keyboard) and a value of 3 indicates they are on opposite sides of the keyboard and can be typed with 2 hands. Note that this feature is used to alter the typing speed – with values depicted I table 1. Note that the value of epsilon is acquired during the typing process itself and knowledge of the typing skill to be simulated. If the typist is assumed to be a novice, then 'ε' is set to one throughout and the value of 'Θ' in (1) remains at 1.0 throughout. Otherwise, the typist is not a novice and the label of 'ε' will be set to 2 by default (and Θ will be set to 0.90). If the characters in the digraph allow 2-hands, then the label will be set to 3 (and Θ set to 0.80). Note that for the intermediate and expert typist, the values of 'ε' will change dynamically as indicated by the text to be typed. In addition, the label associated with 'ε' provides valuable information in the context of keystroke dynamics. Its value is associated with each character in the text file to be generated that serves as the template for automated typing (the text input). Such systems need to know *how* the characters were entered – not just the characters themselves. Though this information is implicitly available in the scan code, systems that rely on the ASCII code will need to explicitly capture this information. Also note that if one is using empirically derived data, the value of Θ is already accounted for by the reduced digraph time. It is incorporated into the model for completeness – allowing one to use purely synthetic data. In the work described in this paper, Θ was set to 1.0.

When a keyevent (KeyDown or KeyUp) occurs, the timer is triggered and the event is time stamped (using QueryTimer, C#, 2010 .NET), along with the scan code. Normally, when someone types on a PC style keyboard within a Windows Operating System based machine, the scan code is passed into the keyboard buffer, where it will be processed by the application. Keystroke dynamics based applications operate in this fashion, polling and reading from the keyboard buffer. What we have opted to do in this work is to bypass writing to the keyboard buffer, and instead we write the scan code and timestamp directly to a text file. The text file then contains the scan code and time stamp for every keypress event. Note that the scan code provides information regarding a KeyDown or KeyUp event, and a table can be used with all of this information if a different keyboard or dictionary is to be utilised. The data file can then be processed in real-time as the data is inserted, and processed as if the data was inserted into the keyboard buffer. Ultimately, this protocol saves the step of polling the keyboard buffer and extracting the information, which will then be stored in memory for processing before being output to permanent storage. The time difference between polling the keyboard buffer and reading from a file is probably negligible in this context. One could certainly stuff the characters into the keyboard buffer if that is required – this is another area of expansion of the model. With the model development at hand, it is time to examine some of the key results of the model.

Table 2. Value of 'ε' and the corresponding digraph path when the KeyDown-KeyDown digraph timing method is utilised. Note that the subscripts ((a) and (b)) refer to a particular key in the digraph.

'ε'	Digraph path
1	$\text{KeyDown}_{(a)}\text{KeyUp}_{(a)} \Longrightarrow \text{KeyDown}_{(b)}\ \text{KeyUp}_{(b)}$
2	$\text{KeyDown}_{(a)}\ \text{KeyDown}_{(b)} \Longrightarrow \text{KeyUp}_{(a)}\ \text{KeyUp}_{(b)}$
3	$\text{KeyDown}_{(a)}\text{KeyDown}_{(b)} \Longrightarrow \text{KeyUp}_{(b)}\ \text{KeyUp}_{(a)}$

4. Results

The model produces typing samples as dictated by the input text and a configuration file with all parameter values set as required for the simulation. The output of the system is a text file that contains scan codes followed by timestamps (1 ms accuracy). To evaluate the effectiveness of the system, several quantitative aspects need to be examined. These include the computed gwpm and wpm, the number of and types of errors, and the accuracy of the keypress events with respect to the stream of scan codes for given sets of simulation parameters. For instance, we can specify parameters for an expert typist, which sets all parameter values in (1) – which must also include a wpm value (e.g. 60 wpm) and provide the input file and then examine the resulting output file. As far as our trials have gone, 1,000s of tests were generated in 1 hour, using a series of 10 standard PCs. The principal from our preliminary analysis is that the resulting error rates and wpm (corrected gwpm) are consistent with the input specifications to within a few percent (see Table 3.) The values for gwpm and wpm are quantized into rather large bins, making it difficult to produce an exact judgment of the efficacy. In our test case of 1,000 samples, all values with respect to typing speed and category were consistent with the typing samples that we had from the two subjects from which the digraphs were derived. When the system was tested on 3 expert typists (64, 72, and 75 gwpm), the results were accurate within a 3% error rate. The same sort of results was produced when the system was tested against 3 novice typists (23, 25, and 29 gwpm). The results of these small studies indicate that the system is able to reproduce the typing speed and error rate of the human typist, though the error that were produced were not consistent with that of the actual typists (only correct with respect to specific categories in about 30% of the cases). Note that in addition to testing on the original text sample, additional tests (inputs) were produced by moving paragraphs up and down in a modulo fashion,. This approach yields a different corpus, but maintains the same character set and the context more or less remains the same for each test text corpus that is produced (8 in this study). The results indicated above apply to all 8 studies/combinations of text input repeat 1,000 times.

Further, the effect of varying the number of hacking attempts on the FAR/FRR was investigated. An expert typist was simulated and enrolled using an 8 character login ID and 8 character password (generated randomly)10 times. This served as the authentic user. A set of imposters was created with intermediate typing skills. This imposter set attempted to log into the system using the same ID and password as the authentic user. Note that each imposter attempted a single login attempt.

Table 3. Sample of digraphs and typing speeds from a user and our ATS system. Note that times and speed are reported as average values. Note that the typing speeds for the typists and ATS were 31-34 gwpm – no statistical differences were detected when text ordering was altered.

Digraph	Digraph Time for User (ms)	Digraph Time for ATS(ms)
‘th’	276	281
‘ea’	234	231

The results of this experiment are presented in table 3. We deployed a probabilistic neural network based approach for the data generated for table 3 (see [17] for details on this approach). As a quality control measure, the same data used in this experiment was used to calculate FAR/FRR data using a bioinformatics based approach (see [18] for details). The results from the bioinformatics approach were consistent with those presented in table 4.

Table 4. False Rejection Rates (FRR) and False Acceptance Rates (FAR) as function of the number of attacks (top row) using a PNN approach [17]. Note the FAR/FRR data are in the left/right hand portion of each column respectively (second row).

10 attempts		100 attempts		10^3 attempts		10^4 attempts	
5.1%	2.0%	3.2%	4.2%	1.1%	5.0%	0.6%	8.5%

5. Conclusion

This study has examined the development of a simple but effective model of human typing that can be used to generate large pools of textual data such as login IDs and passwords that are typically deployed in keystroke dynamics based biometric authentication systems. This is an extremely and necessary utility to have available when developing keystroke dynamics based applications. One of the critical issues in this domain is the consistency of conditions across experiments. A variety of keyboards are deployed, different texts are used for the login IDs and passwords – though this effect may be small in short text strings such as login IDs and passwords. More importantly is the issue of sample sizes. Most experiments deploy a handful of subjects (typically 5-20). This work has demonstrated that the FAR/FRR values may be dependent upon the sample size. These results were validated using two different classification approaches: a probabilistic neural network and a bioinformatics local alignment score (essentially a statistical matching scheme). Both approaches demonstrated a dependency on sample size: an inverse relationship between FAR and sample size and a direct relationship between sample size and FRR values. Whether this holds generally is a matter of empirical validation – and ultimately forms the basis of the need for an ATS system such as this one.

Even though the model is a simplification of the work that has already been produced in this field, it accounts for a significant number of features reported in the literature (12 out of . It is meant to be a phenomenological model which omits most of the cognitive aspects of motor activity incorporated into models such as the KLM and MHP. The parameters are somewhat global in scope – and tend to quantize the actual real word data – actual typing speeds for instance. One could envision the alpha term being more continuous thereby mapping more closely at the upper end of the spectrum to lower intermediate typists for instance. These discontinuities need to be addressed in the next stage of development. Errors can be modeled more closely – they could be modulated by the particular digraph that is being typed rather than being generated randomly from a set of errors. These changes would facilitate the deployment of this model in continuous authentication schemes and would thereby increase the utility of this model. It should be noted that this work does not rely solely on typing speed as a major factor. Implicit in the model (the ‘ε’ parameter) is a critical aspect of typing – the order of digraph characters. This allows the model to transcend mere typing speed and error rate as the major feature of the input space for keystroke dynamics. The order in which characters are entered is important and reflects the typing skill of the person. Expanding on this aspect of typing is another area of further work in this field.

References

[1]. Revett, K., Behavioral Biometrics: A remote access approach, John Wiley and Sons, 2008.

[2]. Forsen, G., Nelson, M. & Staron, R., Personal attributes authentication techniques, In A.F.B. Griffin, Ed. Rome Air Development Center report RADC-TR-77-1033, RADC, New York, 1977

[3]. Revett K., Magalhaes, S. & Santos, H., Developing a Keystroke Dynamics Based Agent Using Rough Sets, The 2005 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Compiegne, pp 56-61, 2005.

[4]. Peacock, A., Typing Patterns: A Key to User Identification. IEEE Security and Privacy, 2(5) September/October 2004, pp. 40-47.

[5]. Monrose, F. & Rubin, A., Authentication via keystroke dynamics. Fourth ACM Conference on Computer and Communications Security, Zurich, Switzerland, pp. 48-56, 1997.

[6]. Leggett, J. & Williams, G., Verifying identity via keystroke characteristics. International Journal of Man-Machine Studies, 28(1), pp. 67-76,1988.

[7]. Brown M., and Rogers S.J., User identification via keystroke characteristics of typed names using neural networks, International Journal of Man-Machine Studies, vol. 39, pp. 999-1014, 1993.

[8]. Bergadano F., Gunetti D., & Picardi C. User authentication through keystroke dynamics, ACM Transactions on Information and System Security, 5(4), pp. 367-397, 2002.

[9]. Card, S., Moran, T.P., & Newell, A., The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

[10]. Rumelhart, D.E., Norman, D.A., 1982. Simulating a skilled typist: a study of skilled cognitive-motor performance. Cognitive Science 6, 1–36.

[11]. Card, S., Moran, T., & Newell, A. The Model Human Processor: An engineering model of human performance. In Boff, K. (Eds.) Handbook of Perception and human performance, New York: Wiley, 1986.

[12]. John, B. E., TYPIST: A theory of performance in skilled typing. Hum. Comput. Inter. 11, pp. 321-355, 1996.

[13]. Liu, Y., Feyen, R., & Tsimoni, O., Queueing network-model human processor (QNMHP): A computational architecture for multitask performance, ACM Transaction on Human-Computer Interactions, 13(1), pp. 31-70, 2006.

[14]. Salthouse, T. A., Perceptual, cognitive, and motoric aspects of transcription typing. Psych.Bull. 99(3), pp. 303-33, 1986.

[15]. http://www.guinnessworldrecords.com/records-9000/fastest-hands-free-typing

[16]. Gentner, D. R., The acquisition of typewriting skill. Acta Psychologica *54*, 233–248, 1983.

[17]. Revett, K., Gorunescu, F., Gorunescu, M., Ene, M., Tenreiro de Magalhães, S., & Santos, H.M.D., Authenticating computer access based on keystroke dynamics using a probabilistic neural network, International Conference on Global E-security, London, UK 20-22 April, 2006 , pp 65- 71.

[18]. Revett, K., A Bioinformatics Based Approach to user Authentication via Keystroke Dynamics, *International Journal of Control, Automation, and Systems,7(1*), pp.7-15, Feb,2009 (ISSN:1598-6446).