Biometric patterns recognition using keystroke dynamics

Jefferson L. S. Ferreira[1], Mardson F. Amorim[1], Ruy A. P. Altafim[1]

[1]Informatic Center – Federal University of Paraíba (UFPB)
Postal Code 58.058-600 – João Pessoa – PB – Brazil

jefferson.lacerda@eng.ci.ufpb.br, mdsamorim@gmail.com, altafim@gmail.com

Abstract. *This paper aims to describe a strategy for biometric authentication embedded system that uses keystroke dynamics to recognize the users. The main motivation of this work is a gap identified on the biometric authentication devices market that demonstrates the lack of a low cost and high efficiency product. Therefore, the use of low cost microcontrollers coupled with a good biometric authentication strategy could fill this gap. The PIC and ESP microcontrollers were used to create a prototype with the purpose of performing measurements and generating users' biometric models. During these measurements 9 volunteers had their typing characteristics extracted and stored. After data collection, several tests were performed and values of 36% for FRR and 7.2% for FAR were found. More expensive results can still be achieved by modifying some punctualities in data collection, as commented at the end of the paper.*

1. Introduction

Security is becoming a concern, due mainly to the increase in violence in its various aspects [van de Weijer et al. 2018]. This mainly includes cybercrimes. Therefore, several new means of authentication have arisen today in order to increase the security of more traditional methods of authentication, such as passwords and PINs. Biometric authentication methods are among these new methods and are proven to be more secure than traditional methods, which have already proven to be out of date [Nandakumar and Jain 2012]. Wayman highlights the 5 main qualities of biometrics: *robustness* because the characteristics measured in the users have little or none variation over the time; *distinctive* because there is great variation among the individuals of a population; *Available* because the entire population ideally has multiple characteristics to be benchmarked; *Accessible* because they are easy to extract using electronic sensors; And *acceptable* because these measures are intrinsic to each person and can not be taken from them [WAYMAN et al. 2015].

Actually some phones use the iris or even the features of the face to allow access to your system [SAMSUNG 2017]. We can also identify several computers that use sensors capable of identifying the patterns of our fingerprint to allow access to the files [DIMEP 2015]. Furthermore, for offices or residences, some companies manufacture electronic locks that use biometric features such as facial features, fingerprints, etc. [Leroy Merlin 2018, DLock 2018]. One common problem that all these products try to solve is secure access to a particular medium. Commonly they can reach a solution for this. However, all the biometric solutions mentioned above have an intrinsic characteristic: they are biometric authentication methods that use physiological characteristics. These types of biometric solutions use specific hardware (such as high-quality cameras) and generally require image processing, which makes the products that come to market more expensive [TEH et al. 2013].

Face, iris, digital patterns (physiological characteristics); and voice and keystroke patterns (behavioral characteristics) are the most common biometric authentication methods. One of these patterns that proves capable of arousing the curiosity of any person due to its simplicity and effectiveness is the one of keystroke.

The keystroke dynamics is a well-explored research area, with papers being published since the 1970s [IBM 1975, FORSEN et al. 1977]. Since then many others papers has been published. What attracted such a large number of researchers was exactly the simplicity of the technique, as different from physiological biometric methods did not require specific hardware to extract the characteristics of an individual. Only one keyboard and one computer were enough to generate a single user pattern.

Keystroke dynamics uses behavioral patterns to generate a unique model of an individual. This means that it is possible to identify a person just by the way they enter a password, for example, by just comparing their pattern with a previously stored one. For generation of this pattern the most important is the way the individual behaves while typing, not just the characters of the entered password. The way you type it becomes essential, even allowing the user to let his password become public (although password verification is also considered an extra layer of security). Monrose goes so far as to say that when we are dealing with typing dynamics, it does not matter *what* you are typing, but *how* you are typing [MONROSE and RUBIN 2000]. This allows us to state that the user will have a different way of typing for each password, since the typing pace changes depending on the position of each key. And, if the pattern is compromised, the individual can simply change the password and generate a new template to invalidate the attempted intrusion. Regarding this [TROJAHN and ORTMEIER 2013] make an important observation. They claim that a system that uses this biometric authentication method coupled with the password secret can be considered a two factor authentication system, greatly elevating the security degree of systems that use only the secret. What is most interesting is that all of this occurs in a non-intrusive way to the user, because the behavioral characteristics are extracted naturally when typing, not requiring a different ritual when entering the password.

It is plausible to ask, then, the possibility of using a type authentication method in electronic lock systems, for example, since the hardware needed to extract the biometric information closely resembles what is already used today. Would an electronic lock using a biometric authentication method would not add an extra layer of security for access to buildings and offices? Thinking this way, this work has as main objective to present an inexpensive and viable solution of biometric authentication using the already explored area of the keystroke dynamics applied to numerical keyboards and that can be embedded in microcontrollers. A prototype was developed to perform several tests using 9 volunteers in order to evaluate the efficiency of the system. Expressive numbers of FAR and FRR were reached using this approach and confirm the feasibility of its application in systems with low processing power. More results and discussions about these can be found in Chapter 4.

2. Related Works

The keystroke dynamics is an area of research already well explored by several researchers and, although many validated works can be found, it is difficult to compare all these works directly in order to define a course for a new work. This was the reason for generating several comparatives between different authentication approaches using keystroke dynamics. The next two sections detail the work that carried out comparisons between different biometric authentication methods, which use the keystroke dynamics of the users and some works that propose different methods of authentication.

2.1. Comparative Works

[KILLOURGHY and MAXION 2009] made a comparative among several classifiers, such as Manhattan, Nearest Neighbor, Outiler Count, etc., in order to define the best approaches among those most used by researchers in the area. However, they cite precisely the problem of different authors using different methods to achieve their results. Another point that draws attention is the collection of data, since each work has its own database or use a different way to collect data.

[SHANMUGAPRIYA and PADMAVATHI 2009] present a comparison between different approaches to security and challenges in implementing this biometric authentication technique. They also cite that a system that uses typing dynamics can operate in the identification and verification modes. Identification would be the process of identifying a person by examining previously calculated biometric standards. In this mode of operation, the system compares the user information with the previously stored patterns of all other users for a match. For a system in check mode, the patterns generated in a particular session are only tested with the patterns of that same person to verify their identity.

Another work of comparison between biometric authentication methods using keystroke dynamics found was written by [TEH et al. 2013]. They note that approximately 49 % of the researchers use what they called Flight Time and 41 % use what they called *Dwell Time*. For this work the expressions *Flight Time* and *Dwell Time* will be called *Keyup-keydown Time* and *Hold Time*, respectively. The significance of these characteristics will be shown later.

According to the aforementioned works, for the generation of an individual's model a biometric authentication system that uses keystroke dynamics can take into consideration how many characteristics the researchers understand. The aforementioned comparisons show this. However, some features deserve to be highlighted. Are they:

- Time features such as *hold time*, *keyup-keydown time* and *keydown-keydown time*.
  - HT - Hold time: is the term used to define the time in which a key is held down;
  - UDT - Keyup-keydown time: denomination to define the time between a key to be released and another key to be pressed;
  - DDT - Keydown-Keydown: time defined between pressing two consecutive keys.
- Consider whether or not to enter as part of the password;
- Use pressure sensors on each key.

Also according to the above comparisons, the most used method to verify the effectiveness of the systems is through the calculation of 2 metrics:

- *False Rejection Rate* (FRR): refers to the rate at which the system declines a legitimate user from the total legitimate users.

- *False Acceptance Rate* (FAR): is the rate at which the system accepts an unauthorized user with respect to the total of impostors.

The way authors calculate FRR and FAR is not explicitly demonstrated in their work. However, because it is a rate based on the relation between access attempts and success or not in authentication, it is possible to infer that the calculation mode is the same for all.

2.2. Works that propose different approaches to keystroke dynamics

[ARAÚJO et al. 2005] use a string of up to 10 characters for your input data. They collected 10 samples from each user, where the values of HT, UDT and DDT were extracted. The password characters were also used as information for model generation. Thus, the model was formed by 4 characteristics. The developed system operated with a precision of 1ms for each measurement. They also comment that a distance-based statistical method was used, since neural networks are not adequate for this approach and that fuzzy logic has already been explored in another work of the same. In the end, they achieved 1.45 % FRR and 1.89 % FAR at their best. It is worth mentioning that the system of [ARAÚJO et al. 2005] had a re-training mechanism, allowing the samples used to generate the models were always updated.

[HAIDER et al. 2000] have compared the application of several methods: fuzzy logic, neural networks, statistical methods. The set of data formed by them was composed of a vector of measures of HT and UDT. For the statistical method, the system generated, from these measurements, the mean and standard deviation, in addition to generating a confidence interval. After this, during the authentication mode, the system recorded the times generated in that session and verified if it was within the confidence interval for its model. Using this approach, it was possible to achieve FRR and FAR rates of 2 % and 13 % respectively, which is the most accurate approach to their work.

[TROJAHN and ORTMEIER 2013] evaluated the dynamics of typing on mobile devices using touch screens. Classification algorithms were used to generate user models. The data collected was slightly different. They analyzed what they called the “ digrafo ”, which are the characteristics like speed and acceleration between pressing two keys; the pressure on the screen; and the area of contact between the finger and the screen. They also affirmed that alphanumeric passwords guarantee more security and efficiency for the system. The obtained FRR and FAR minimums were 2.03 % and 2.67 % on average, respectively, using the classifier J48.

The system developed by [LEE et al. 2017] used only 8-digit passwords (or PINs) to generate the users' model. The data were extracted from a conventional numeric keypad plugged into an Arduino UNO microcontroller. After data collection the values were extracted from the microcontroller and the models were generated using Fuzzy logic. The results show a maximum accuracy of 44.12% during the tests.

[GRABHAM and WHITE 2008] have developed a biometric authentication system using keystroke dynamics applied to a numeric keypad similar to ATMs. They modified a keyboard by inserting individual sensors for each key that can measure the applied pressure and the time when a user presses a key. Each password, or PIN as they call it, had 4 digits and the times used were HT, which they called “ *on time* ”, and UDT, which was called ‘*off time* ”. The pressure information used was based on the mean and peak pressure across all keys. It was not considered pressing ENTER to generate the model. They reached a FAR of 15 % and an FRR of 0 %.

The work of [SAINI et al. 2017] also brings some interesting information. They developed a system on an ordinary computer to capture data from their numeric keypad. No hardware changes were made. The passwords used were also only of numbers and only the times were used to generate the model. The time-extracted features were hold time (HT), Press-time time (DDT), Press-release time, Release-release time, and Release-press time (UDT). To classify these data a few different methods were used, including Random Forest and Naive Bayes. The results show a minimum FAR of 2.7 % and a minimum FRR of 35.9 %.

The points that all the works raised in this bibliographic review have as intersection and that deserve to be highlighted is the fact that the keystroke dynamics is efficient in most cases and that, because it does not require specific hardware, it stands out enough compared to other biometric authentication methods.

3. Methodology

The system developed here works by identifying and verifying the identity of the users. Using the PIN numbers identification is performed and, from there, the identity of the user is verified through the stored model. To validate this system it was necessary to develop an embedded prototype with the authentication strategy and a user interface to extract biometric information (keypad). From this prototype a data collection of volunteers was carried out to generate the biometric models of the same in order to perform authentication tests.

3.1. Authentication strategy

The authentication strategy used for such a system must be both efficient in terms of security and computational performance. A statistical method for model generation was chosen. This method consists of using the mean of the times and the standard deviation of these averages, in addition to the characters of the password, to generate such models, similar to the work of [ARAÚJO et al. 2005]. So, the model basically consists of the confidence interval generated from this data for each user.

It is worth noting that the system developed here is intended to implement two-factor authentication, i.e., PIN numbers will also serve as a first step of authentication, meaning that the PIN secret remains important.

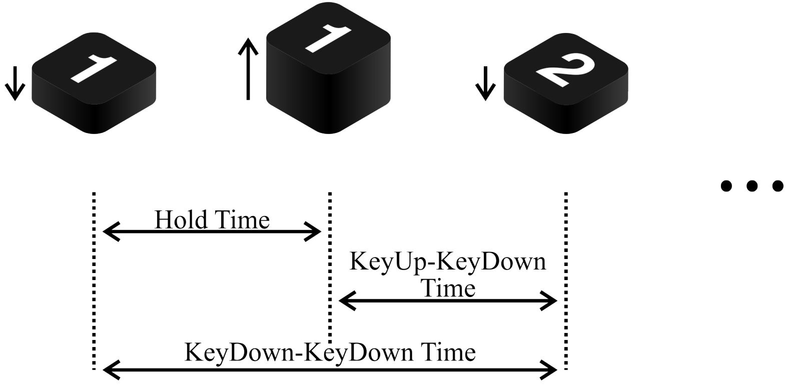
3.1.1. Password Setting

Many of the papers found use alphanumeric passwords to perform user authentication. However, since the focus was the application for locks, where keyboards usually only have numbers, the password must be purely numerical. The term PIN (*Personal Identification Number*) is used in several papers to set a purely numeric password [GRABHAM and WHITE 2008, SAINI et al. 2017, LEE et al. 2017]. Therefore, we will also use it to reference users' passwords in this work. The three authors mentioned above used PINs of different lengths (4, 6 and 8, respectively). In this work, we use 6-digit PIN plus ENTER. This means, roughly speaking, a 7-digit PIN because the ENTER time-out data is also considered for model generation.

3.1.2. Time Metrics Used

Most related work uses three time metrics (HT, UDT, and DDT). However, it is believed that only HT and UDT can obtain relevant results, since there is a clear correlation between DDT and the other two metrics ($DDT = HT + UDT$). The figure 1 provides a better understanding of each metric mentioned above.

Figure 1. Used metrics



3.1.3. Model Generation

A user's model represents his identity for the authentication system. When entering your PIN the system will compare the data obtained with the previously recorded information (the model) and verify if the user is legitimate or not. Therefore, it is essential that each model actually represents the corresponding user of the system. As mentioned above, the model is defined by vectors containing the averages of the HT and UDT metrics and by their standard deviations, in addition to the character of each key. And since the PIN has 7 digits (6 numbers + ENTER), each vector will have 7 averages and 7 standard deviations.

The abovementioned vectors are defined as follows:

$$P_c = \{p_1, p_2, ..., p_i\} \tag{1}$$

$$HT_c = \{\mu_{ht_1}, \mu_{ht_2}, ..., \mu_{ht_i}\} \tag{2}$$

$$DP_{HT_c} = \{\sigma_{ht_1}, \sigma_{ht_2}, ..., \sigma_{ht_i}\} \tag{3}$$

$$UDT_c = \{\mu_{udt_1}, \mu_{udt_2}, ..., \mu_{udt_i}\} \tag{4}$$

$$DP_{UDT_c} = \{\sigma_{udt_1}, \sigma_{udt_2}, ..., \sigma_{udt_i}\} \tag{5}$$

Where the equation 1 represents the vector with the numbers referring to the PIN, Equations 2 and 4 represent the means for the corresponding metric and Equations 3 and 5 correspond to the respective standard deviations, all with respect to the user c. For all cases $1 \leq i \leq 7$ corresponds to the number of keys.

The means and standard deviations, represented by μ_{Met_i} and σ_{Met_i} are respectively the means and standard deviations of the metric Met in relation to the last n entries of the user for the i key. These averages are defined as follows:

$$Met_i = \frac{1}{n}\sum_{j=1}^{n} Met_i(j) \tag{6}$$

The standard deviations are defined as follows:

$$\sigma_{Met_i} = \sqrt{\frac{\sum_{j=1}^{n}(Met_i(j) - \mu_{Met_i})^2}{n}} \tag{7}$$

Where $Met_i(j)$ represents a metric relative to the key i of the j -th sample of user measures.

3.1.4. Definition of decision threshold

The definition of the decision threshold for this type of work is extremely important. It is through this threshold that the system will define whether a user is, in fact, legitimate. Therefore, the use of a relaxed threshold can cause a large number of false positives (high FAR value), while a fairer threshold can cause a high number of false negatives (high FRR value). The intention is therefore to find a middle ground that leads to low values for both FAR and FRR simultaneously, but mainly for FAR.

In this work, a confidence interval with a confidence index of 95 % will be set up. This range will be used to define whether a user is actually legitimate or not. Considering that the amount of samples used is large enough, we can use the Central Limit Theorem to infer that the distribution of such samples is normal. This is important so that we can determine the confidence interval, which in this case is similar to [HAIDER et al. 2000] and is defined as follows:

$$(\mu - z\sigma \leq \overline{\mu} \leq \mu + z\sigma) \tag{8}$$

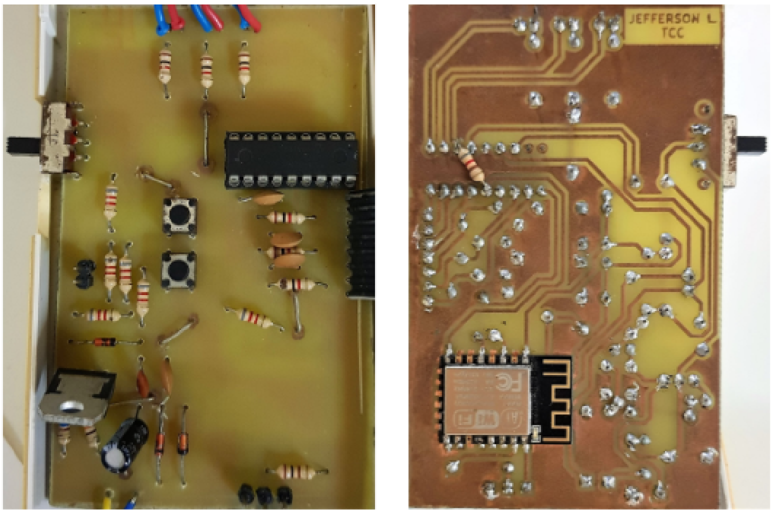
In equation 8, μ is the sample mean stored in the user model, z is a critical value of a normal distribution that defines the degree of confidence and can be obtained through a table [UFPR], $\overline{\mu}$ is the estimator for the mean to be measured and σ is the standard deviation of the samples.

With the given confidence interval, we can say that a user who, upon entering a password, results in averages outside this range will be considered as an impostor. Only users who display values within this range will be authenticated.

3.2. Hardware developed

The hardware developed consisted of a board containing the ESP8266 12F and PIC16F628A microcontrollers. The development of this board was necessary to organize all these components in order to avoid interference and bad contacts. Since there is an I ^{2} C and 4-wire bus where high frequency signals are transmitted, the use of long wires could cause interference between themselves, as seen during some protoboard tests. The figure 2 illustrates the developed board.

Figure 2. PCB developed



3.2.1. ESP8266 12F microcontroller

The ESP is the “brain” of the system. It is responsible for storing the data used to generate each user’s model and perform the necessary calculations on that data. In addition, it acts as an interface between the microcontroller and the developer, allowing the extraction of data via Wi-Fi, without physical access to memory. The ESP was chosen because of the ease of prototyping it provides and due to the low cost compared to other microcontrollers that also have Wi-Fi. The ESP model used was the ESP-12E because it has a good amount of flash memory and enough of available digital pins, plus good stability in your Wi-Fi connection.

3.2.2. PIC16F628A microcontroller

For this work it was necessary to use a microcontroller dedicated to time measurements, in order to guarantee accuracy and integrity of measurements. One reason for choosing the PIC was precisely the degree of accuracy achieved when using the Assembly programming language to program it. The PIC is responsible for performing the lowest level work of the system. It does all the management of the keyboard, calculating and storing times, verifying the integrity of each PIN entered and sending that data in packets of protocol L3 (data protocol developed by LMI researchers), via protocol I^{2}C, to the ESP with each reading. The PIC has two timers counters, one of 8 bits and one of 16 bits. This allows it to count up to 65 milliseconds and 524 milliseconds, respectively, using the appropriate prescaler settings. This means that the PIC, using its native counters, would not be able to count a key press for 2 seconds, for example. To overcome this problem a variable was used to aid in counting, allowing counting to be done with 24 bits. In this way, it is possible to perform counts of up to 16.7 seconds with an accuracy of 1 microsecond. This precision of 1 microsecond is an important point of the work, so it is guaranteed that users' models will be fed with extremely accurate time data. It is necessary to mention that the times mentioned above are based on a clock of 4 MHz for the PIC. The PIC also was responsible for informing the user, through the indication LEDs, if the PIN entered is within the established standards or not. The PIC was chosen for these tasks due to the high reliability it provides and the low cost.

3.3. Software developed

The development of the logical part of the work, *software*, can be divided into two stages:

- Acquisition and testing: *Software* developed for the collection and pre-processing of data;
- Authentication: *Software* responsible for performing user authentication using templates generated from previously collected data.

These two steps were necessary because different processes were to be followed in each of them and there was no need for a single software to perform all these procedures. During each of these steps, different complementary software was developed for both the ESP and the PIC.

3.3.1. Acquisition and tests step

Before implementing the final system, in which the user would enter the PIN and receive the return if it had been authenticated or not, it was necessary to develop intermediary softwares, both for ESP and PIC, to enable the acquisition of user data for performing tests.

The ESP software, at this stage, was responsible for storing user data and providing an interface so that such data could be extracted without the need for physical access to memory. Thus, the data, encapsulated in L3 protocol packets, was received from the PIC through the I^{2}C bus and stored in a file named with the PIN entered by the user. The data was accessible through a server implemented in the ESP.

The software of the PIC was responsible for reading the keyboard, capturing the typing times and sending them to the ESP. The PIC was always waiting for a PIN to be entered and, at the end of the typing, it checked the integrity of the PIN, searching for errors in the number of digits and if the ENTER was the last keystroke entered. If all requirements were confirmed, the PIN and its time data were encapsulated in L3 protocol packets and sent through the I^{2}C bus to the ESP. In addition, the PIC was also responsible for signaling to the user the current status of the system through 3 LEDs.

3.3.2. Authentication step

In the authentication step, some modifications to the acquisition and testing software were necessary. In this case, in addition to storing user data, the system would need to generate templates for each of the users and, from each user's confidence intervals, define whether a new entry represented a legitimate user or not.

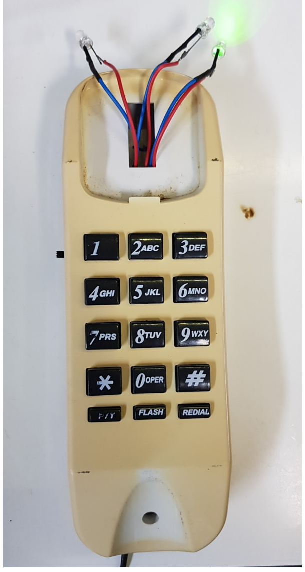
The difference of the software implemented in the ESP, at this stage, was that it should store the models of each user. Thus, after receiving the time data measured by the PIC, it would verify if these values are within the confidence interval for that user and would respond back to the PIC if the user should or should not be authenticated.

In PIC the changes were more subtle. Instead of just sending the data to the ESP after performing the time measurements and checks the PIN integrity, it should wait for the ESP response and turn on the blue LED if the user were legitimate or orange if it were an imposter.

3.4. User Interface

The system interface with users is a keyboard, where they should enter the PIN, and 3 LEDs indicating the status of the system. The mentioned keyboard can be seen in Figure 3.

Figure 3. Prototype keyboard



The numeric keys served as input for the PINs, and the ∗ and # keys functioned as system functions. ∗ was used to restart the action if the user erased the PIN and noticed it in the middle of typing. So the measurements made so far would be discarded and the PIC would be ready for a new reading. The # represents the ENTER and is used to confirm the entered PIN. The other keys were not used. The LEDs, during the acquisition and testing phase, indicated whether the system was in operation (green LED), if a key was being pressed (blue LED), when typos (orange LED) occurred and when the PIC was in the middle transmission with the ESP (blue LED).

3.5. Data collection

During the data collection the developed device was arranged in a fixed position on top of a bench inside a research laboratory of the UFPB, because the change of position of the keyboard and the disposition of the volunteer at the time of the collection is able to change the standard of typing it. 9 volunteers were asked to enter a PIN over 1 week in several sessions per day. The idea was that each time these volunteers entered and left the laboratory the password was typed between 2 and 3 times. The volunteers were divided between men and women between the ages of 20 and 50, undergraduates and doctor, without any background in information security and all right-handed. The PIN used by each volunteer was randomly generated in order to avoid possible prior vices to that sequence. Volunteers, as well as their PINs and the amount of samples collected, can be seen in the Table 1.

Table 1. Volunteers, PINs and amount of collected samples

Volunteer	PIN	Amount of samples
1	557670	102
2	123456	92
3	183249	55
4	618469	55
5	304659	46
6	192408	37
7	513179	37
8	958557	13
9	498957	11

From these samples only the last 40, when available, were used to generate the model since during the first attempts the volunteers are still becoming familiar with the prototype. Different quantities of samples were collected in order to evaluate the ideal quantity for model generation and the minimum number of samples needed for the users to familiarize themselves with the system.

4. Tests and Results

Once the data was collected and the authentication system was developed, it was possible to carry out an analysis of the efficiency of the system as a whole. To perform this analysis, the FAR and FRR numbers obtained were used. The following sections detail the tests performed and the results obtained.

4.1. Tests with collected data

During these tests, a fraction of the data collected from each user was separated to be used as a set of tests, while the rest was used to generate the model. The last 20% of the entire data set was used for testing. As for volunteers with more than 48 samples, the last eight samples were used for the tests and the 40 predecessors, from the end of the test set, were used to generate the model. From there, tests were carried out to verify that the samples from the test set were within the confidence interval of the generated model. During these tests two confidence indexes were used: 95% and 99%. Since the test samples are all legitimate, only FRR values can be obtained. Table 2 illustrates such results.

Table 2. FRR of tests with collected data

Voluntï¿½rio	FRR 95%	FRR 99%
1	50%	12.5%
2	25%	12.5%
3	12.5%	0%
4	37.5%	0%
5	0%	0%
6	87%	57%
7	43%	29%
8	67%	33%
9	50%	0%
Total	**41.33%**	**16%**

It can be noted that the values vary greatly from user to user, considering both confidence indexes. However, it is possible to note that for users with more samples the values are closer to 0.

The 95% confidence index was chosen for the next tests because the major concern in this case was with the system allowing an unauthorized individual to access the system. A confidence index of 95% results in a smaller confidence interval, reducing the margin of error for the user and this theoretically implies a lower number of FAR. However, you can increase the FRR rate if the user model does not match your biometric signature.

4.2. Tests with Embedded Authenticator

At this stage of testing the system operated in authentication mode and both legitimate users and impostors were asked to try to enter PINs. During these tests, where more than 1300 samples were collected in total, FAR and FRR rates were generated for a confidence interval with a confidence index of 95%. The system performs authentication extremely fast and the authentication time, about 30 milliseconds, becomes irrelevant in this mode. Just to get an idea, the fastest time of a press count during the tests (HT) was 60 milliseconds. This means that the time the system takes to authenticate a user is half the time of an extremely fast keystroke.

In the first part, only legitimate users were asked to enter their PINs. In this way, it was possible to obtain the FRR numbers contained in the Table 3.

It is possible to notice that the FRR numbers vary greatly according to the user. Despite this, users 2 and 6 obtained good indexes and this can be explained by the high index of correspondence between the user's biometric model and the model generated from the data collection. In other hand, the data of user 3 show that was not possible to generate a good biometric model to him.

For the next tests, guest impostors who followed the same standards as the volunteers, were separated into two groups: common impostors and impostor spies. The common impostors only knew the PINs of the users and were asked to insert them in the prototype to measure the FAR. The spy impostors, before attempting to enter the PIN, observed how a legitimate user entered their PIN numbers only after attempting authentication. Tests with common impostors resulted in the FAR values shown in Table 4.

Table 3. FRR values of real tests

Volunteer	FRR
1	43.3%
2	4.1%
3	70.2%
4	28.2%
5	34.1%
6	0.05%
7	16.1%
8	53.6%
9	64.1%
Total	**34.9%**

Table 4. FAR values with common impostors

Volunteer	FAR
1	0%
2	0%
3	0%
4	7.5%
5	18.4%
6	20.4%
7	13.4%
8	12.2%
9	3.8%
Total	**9,17%**

FAR rates, in this case, were quite low for users with more than 50 samples. For users with few samples the FAR number was quite high. This means that the standard deviations of the samples collected for these users were quite high, confirming the idea that the first samples should not be used to generate the models as users are still becoming familiar with the system. According to test students, the first 20 samples should be discarded because the users are not yet familiar with the system. It is also worth noting that the impostors had no idea how legitimate users entered the password, as they did not see a legitimate user authentication, further contributing to better FAR rates.

The next tests are those performed with spy impostors (those who observed a legitimate user typing their password before attempting to circumvent the system) and resulted in FAR values that can be seen in Table 5.

The tests with spy impostors provoked a small variation with the tests with common imposters. Even with spy impostors viewing the typing mode of legitimate users, they have not been able to circumvent the system most of the time, considering volunteers with fairer models. This reflects the difficulty of manipulating the dynamics of typing itself in order to reach the model generated by other users. This proves the efficiency of the method when the user has a model generated from data that in fact reflects on their biometric model. In these cases, the dynamics of each user's typing allowed the generation

Table 5. FAR values with spy impostors

Volunteer	FAR
1	0%
2	0%
3	10.2%
4	7.5%
5	37.5%
6	10%
7	22.2%
8	0%
9	12.5%
Total	**11,1%**

of a single model that guaranteed an extra layer of security during authentication.

5. Conclusion

The main objective of this work was to present a behavioral biometric authentication system embedded in microcontrollers that extracted patterns from the users' keystroke dynamics. It is possible to say, then, that the goal was partially achieved, because the resulting system was able to identify and authenticate users using models based on behavioral characteristics of the same with numbers acceptable for an initial work. The minimum FAR and FRR rates reached were 9.3 % and 34.9 %, respectively, considering all users of the system. These numbers are even lower when volunteers with few samples collected are excluded from the results. This fact is explained by the non-real representation of the biometric model. Users with more than 50 samples collected had an average FAR of 7.2 % and FRR of 36 %. This reinforces the idea that, approximately, the first 20 samples serve only to familiarize the user with the system, making it impossible to use these samples to generate the user representative model. Despite the good results, it is believed that improving the data collection process, establishing more rigorous criteria and selecting volunteers more committed to the cause, allows to obtain results even more satisfactory and close to reality.

For future works some points of this work could be improved and new features added. As mentioned earlier, data collection is a plausible point of improvement. Establishing a minimum number of samples for each user is a possible improvement in this process. In the case of authentication system, this could receive a re-training mechanism, where the model would be constantly updated with new data collected from the successful authentication of the users. This would allow a more flexible and representative model of the user, which would follow the day-to-day pace changes of the user. In addition, a complementary study also needs to be done to better define the most adequate index of confidence, in order to establish a good FAR index without compromising FRR.

References

ARAÚJO, L. C. F., SUCUPIRA, L. H. R., LIZÁRRAGA, M. G., LING, L. L., and YABUUTI, J. B. T. (2005). User authentication through typing biometrics features. ***IEEE Transactions On Signal Processing***, 53(3).

DIMEP (2015). Entenda como funciona um notebook com leitor biométrico.

DLock (2018). Fechadura biométrica dl 2800.

FORSEN, G. E., NELSON, M. R., and STARON, R. J. (1977). Personal attributes authentication techniques. ***Rome Air Development Center***.

GRABHAM, N. J. and WHITE, N. M. (2008). Use of a novel keypad biometric for enhanced user identity verification. ***IEEE International Instrumentation and Measurement Technology Conference***.

HAIDER, S., ABBAS, A., and ZAIDI, A. K. (2000). A multi-technique approach for user identification through keystroke dynamics. ***Systems, Man, and Cybernetics, 2000 IEEE International Conference on***, 2:1336–1341.

IBM (1975). **Keyboard Apparatus for Personal Identification**. IBM.

KILLOURGHY, K. S. and MAXION, R. A. (2009). Comparing anomaly-detection algorithms for keystroke dynamics. ***2009 IEEE/IFIP International Conference on Dependable Systems Networks***, pages 125–134.

LEE, H. S., LAU, T. S., LAI, W. K., KING, Y. C., and LIM, L. L. (2017). User identification of numerical keypad typing patterns with subtractive clustering fuzzy inference. ***2017 IEEE 15th Student Conference on Research and Development***, pages 83–88.

Leroy Merlin (2018). Fechaduras eletrônicas.

MONROSE, F. and RUBIN, A. (2000). Keystroke dynamics as a biometric for authentication. ***Future Generation Computer Systems***, 16(4):351–359.

Nandakumar, K. and Jain, A. K. (2012). Biometric authentication: System security and user privacy. ***Computer***, 45:87–92.

SAINI, B. S., KAUR, N., and BHATIA, K. S. (2017). Keystroke dynamics based user authentication using numeric keypad. ***2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence***, pages 25–29.

SAMSUNG (2017). Leitor de íris: Samsung galaxy s8 e s8+.

SHANMUGAPRIYA, M. D. and PADMAVATHI, D. G. (2009). A survey of biometric keystroke dynamics: Approaches, security and challenges. ***International Journal of Computer Science and Information Security***, 5(1):115–119.

TEH, P. S., TEOH, A. B. J., and YUE, S. (2013). A survey of keystroke dynamics biometrics. ***The Scientific World Journal***, 2013.

TROJAHN, M. and ORTMEIER, F. (2013). Toward mobile authentication with keystroke dynamics on mobile phones and tablets. ***2013 27th International Conference on Advanced Information Networking and Applications Workshops***, pages 697–702.

UFPR. Tabela normal padrão.

van de Weijer, S. G., Leukfeldt, R., and Bernasco, W. (2018). Determinants of reporting cybercrime: A comparison between identity theft, consumer fraud, and hacking. ***European Journal of Criminology***, 0(0):1477370818773610.

WAYMAN, J., JAIN, A., MALTONI, D., and MAIO, D. (2015). An introduction to biometric authentication systems. ***Biometric Systems***, pages 1–20.