Keystroke Dynamics Authentication

Romain Giot, Mohamad El-Abed, Christophe Rosenberger

8

Keystroke Dynamics Authentication

Romain Giot, Mohamad El-Abed and Christophe Rosenberger
GREYC Research Lab
Université de Caen Basse Normandie, CNRS, ENSICAEN
France

1. Introduction

Everybody needs to authenticate himself on his computer before using it, or even before using different applications (email, e-commerce, intranet, ...). Most of the times, the adopted authentication procedure is the use of a classical couple of *login* and *password*. In order to be efficient and secure, the user must adopt a strict management of its credentials (regular changing of the password, use of different credentials for different services, use of a strong password containing various types of characters and no word contained in a dictionary). As these conditions are quite strict and difficult to be applied for most users, they do not not respect them. This is a big security flaw in the authentication mechanism (Conklin et al., 2004). According to the 2002 NTA Monitor Password Survey[1], a study done on 500 users shows that there is approximately 21 passwords per user, 81% of them use common passwords and 30% of them write their passwords down or store them in a file. Hence, password-based solutions suffer from several security drawbacks.

A solution to this problem, is the use of *strong authentication*. With a strong authentication system, you need to provide, at least, two different authenticators among the three following: (a) *what you know* such as passwords , (b) *what you own* such as smart cards and (c) *what you are* which is inherent to your person, such as biometric data. You can adopt a more secure password-based authentication by including the *keystroke dynamics* verification (Gaines et al., 1980; Giot et al., 2009c). In this case, the strong authentication is provided by what we know (the password) and what we are (the way of typing it). With such a scheme, during an authentication, we verify two issues: (i) is the credential correct ? (ii) is the way of typing it similar ? If an attacker is able to steal the credential of a user, he will be rejected by the verification system because he will not be able to type the genuine password in a same manner as its owner. With this short example, we can see the benefits of this behavioral modality. Figure 1 presents the enrollment and verification schemes of keystroke dynamics authentication systems.

We have seen that keystroke dynamics allows to secure the authentication process by verifying the way of typing the credentials. It can also be used to secure the session after its opening by detecting the changing of typing behavior in the session (Bergadano et al., 2002; Marsters, 2009). In this case, we talk about continuous authentication (Rao, 2005), the computer knows how the user interacts with its keyboard. It is able to recognize if another individual uses the

[1] http://www.nta-monitor.com/

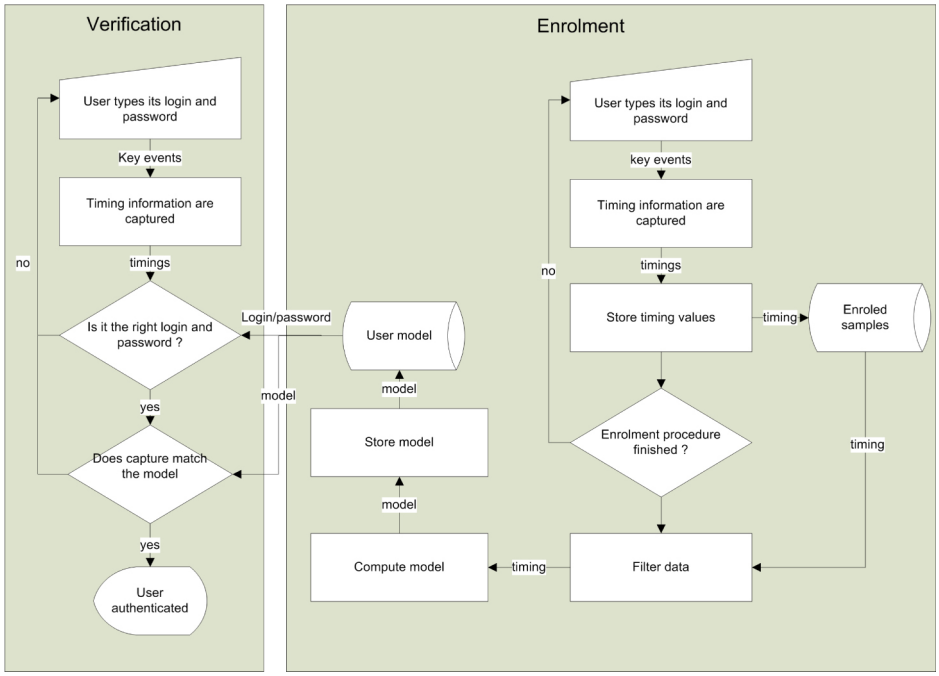

Fig. 1. Keystroke dynamics enrolment and authentication schemes: A password-based authentication scenario

keyboard, because the way of interacting with it is different. Moreover, keystroke dynamics can also prevent the steal of data or non authorized computer use by attackers.
In this chapter, we present the general research field in keystroke dynamics based methods. Section 2 presents generalities on keystroke dynamics as the topology of keystroke dynamics methods and its field of application. Even if it has not been studied a lot comparing to other biometric modalities (see Table 1), keystroke dynamics is a biometric modality studied for many years. The first reference to such system dates from 1975 (Spillane, 1975), while the first real study dates from 1980 (Gaines et al., 1980). Since, new methods appeared all along the time which implies the proposal of many keystroke dynamics systems. They can be static, dynamic, based on one or two classes pattern recognition methods. The aim of this section is to explain all these points.

Modality	keystroke dynamics	gait	fingerprint	face	iris	voice
Nb doc.	2,330	1,390	17,700	18,300	10,300	14,000

Table 1. Number of documents referenced by Google Scholar per modality. The query is "*modality* biometric authentication"

In section 3, we present the acquisition and features extraction processes of keystroke dynamics systems. Section 4 presents the authentication process of such keystroke dynamics based methods. These methods can be of different types: one class based (in this case, the model of a user is only built with its own samples), or two classes based (in this case, the model of a user is built also with samples of impostors). For one class problems, studies are based

on distance measures Monrose & Rubin (1997), others on statistical properties (de Magalhaes et al., 2005; Hocquet et al., 2006) or bioinformatics tools Revett (2009). Concerning two classes problems, neural networks (Bartmann et al., 2007) and Support Vectors Machines (SVM) (Giot et al., 2009c) have been used. Section 5 presents the evaluation aspects (performance, satisfaction and security) of keystroke dynamics systems. A conclusion of the chapter and some emerging trends in this research field are given in section 6.

2. Generalities

2.1 Keystroke dynamics topology

Keystroke dynamics has been first imagined in 1975 (Spillane, 1975) and it has been proved to work in early eigthies (Gaines et al., 1980). First studies have proved that keystroke dynamics works quite well when providing a lot of data to create the model of a user. Nowadays, we are able to perform good performance without necessitating to ask a user to give a lot of data. "A lot of data" means typing a lot of texts on a computer. This possibility of using, or not, a lot of data to create the model allows us to have two main families of keystroke dynamics methods (as illustrated in Figure 2):

- The *static families*, where the user is asked to type several times the same string in order to build its model. During the authentication phase, the user is supposed to provide the same string captured during his enrollment. Such methodology is really appropriate to authenticate an individual by asking him to type its own password, before login to its computer session, and verifying if its way of typing matches the model. Changing the password implies to enroll again, because the methods are not able to work with a different password. Two main procedures exist: the use of a real *password* and, the use of a *common secret*. In the first case, each user uses its own password, and the pattern recognition methods which can be applied can only use one class classifiers or distance measures. In the second case, all users share the same password and we have to address a two classes problem (genuine and impostor samples) (Bartmann et al., 2007; Giot et al., 2009c). Such systems can work even if all the impostors were not present during the training phase (Bartmann et al., 2007).
- The *dynamic families* allow to authenticate individuals independently of what they are typing on the keyboard. Usually, they are required to provide a lot of typing data to create their model (directly by asking them to type some long texts, or indirectly by monitoring their computer use during a certain period). In this solution, the user can be verified on the fly all the time he uses its computer. We can detect a changing of user during the computer usage. This is related as continuous authentication in the literature. When we are able to model the behavior of a user, whatever the thing he types, we can also authenticate him through a challenge during the normal login process: we ask the user to type a random phrase, or a shared secret (as a one-time password, for example).

2.2 Applications and interest

From the topology depicted in Figure 2, we can imagine many applications. Most of them have been presented in scientific papers and some of them are proposed by commercial applications.

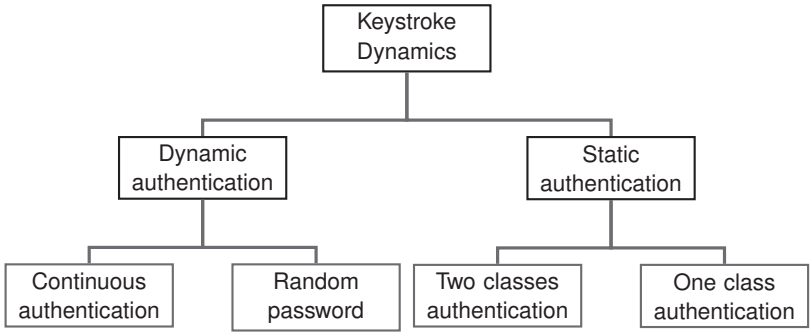

Fig. 2. Topology of keystroke dynamics families

2.2.1 Authentication for logical access control

Most of commercial softwares are related to static keystroke dynamics authentication by modifying the Operating System login procedure. The authentication form is modified to include the capture of the timing information of the password (see Section 3.2.1), and, in addition of verifying the password, the way of typing is also verified. If it matches to the user profile, he is authenticated. Otherwise, he is rejected and considered as an impostor. By this way, we obtain two authentication factors (strong authentication): (i) *what we know*, which is the password of the user; (ii) *what we are*, which is the way of typing the password. The best practices of password management are rarely (even never) respected (regular change of password, use of a complex password, forbid to write the password on a paper, ...), because they are too restrictive. Moreover, they can be easily obtained by sniffing network, since a wide range of websites or protocols do not implement any protection measures on the transmission links. That is here, where keystroke dynamics is interesting, since it allows to avoid impostors which were able to get the password to authenticate instead of the real user. In addition, some studies showed that keystroke dynamics holds better performance when using simple passwords, than more complicated ones. If the user keeps a simple password, he remembers it more easily, and, administrators lost less time by giving new passwords.
When used in a logical access control, the keystroke dynamics process uses different information such as the name of the user, the password of the user, the name and the password of the user, an additional passphrase (common for all the users, unique to the user). Modi & Elliott (2006) show that, sadly, using spontaneously generated password does not give interesting performance. This avoids the use of one time passwords associated to keystroke dynamics (when we are not in a monitoring way of capturing biometric data).

2.2.2 Monitoring and continuous authentication

Continuously monitoring the way the user interacts with the keyboard is interesting (Ahmed & Traore, 2008; Rao, 2005; Song et al., 1997). With such a mechanism, the system is able to detect the change of user during the session life. By this way, the computer is able to lock the session if it detects that the user is different than the one which has previously been authenticated on this computer. Such monitoring can also be used to analyse the behavior of the user (instead its identity), and, detect abnormal activities while accessing to highly restricted documents or executing tasks in an environment where the user must be alert at all the times (Monrose & Rubin, 2000).

Continuous authentication is interesting, but has a lot of privacy concerns, because the system monitors all the events. Marsters (2009) proposes a solution to this problem of privacy. His keystroke dynamics system is not able to get the typed text from the biometric data. It collects quadgraphs (more information on ngraphs is given later in the chapter) for latency and trigraphs for duration. Instead of storing this information in an ordered log, it is stored in a matrix. By this way, it is impossible to recover the chronological log of keystroke, and, improve the privacy of the data.

2.2.3 Ancillary information

Keystroke dynamics can also be used in different contexts than the authentication. Monrose & Rubin (2000) suggest the use of keystroke dynamics to verify the state of the user and alert a third party if its behavior is abnormal. But, this was just a suggestion, and not a verification. Hocquet et al. (2006) show that keystroke dynamics users can be categorised into different groups. They automatically assign each user to a group (authors empirically use 4 clusters). The parameters of the keystroke dynamics system are different for each group (and common for each user of the group), which allows to improve the performance of the system. However, there is no semantic information on the group, as everything is automatic. Giot & Rosenberger (2011) show that it is possible to recognize the gender of an individual who types a predefined string. The gender recognition accuracy is superior to 91%. This information can be useful to automatically verify if the gender given by an individual is correct. It can be also used as an extra feature during the authentication process in order to improve the performance. Authors achieved an improvement of 20% of the Error Equal Rate (EER) when using the guessed gender information during the verification process. Epp (2010) shows that it is possible to get the emotional state of an individual through its keystroke dynamics. The author argues that if the computer is able to get the emotional state of the user, it can adapt its interface depending on this state. Such ability facilitates computer-mediated communication (communication through a computer). He respectively obtains 79.5% and 84.2% of correct classification for the relaxed and tired states. Khanna & Sasikumar (2010) show that 70% of users decrease their typing speed while there are in a negative emotional state (compared to a neutral emotional state) and 83% of users increase their typing speed when their are in a positive emotional state. Keystroke dynamics is also used to differentiate human behavior and robot behavior in keyboard use. This way, it is possible to detect a bot which controls the computer, and, intercepts its actions (Stefan & Yao, 2008).

3. Keystroke dynamics capture

The capture phase is considered as an important issue within the biometric authentication process. The capture takes place at two different important times:

- *The enrollment*, where it is necessary to collect several samples of the user in order to build its model. Depending of the type of keystroke dynamics systems, the enrollment procedure can be relatively different (typing of the same fixed string several times, monitoring of the computer usage, . . .), and, the quantity of required data can be totally different between the studies (from five inputs (Giot et al., 2009c) to more than one hundred Obaidat & Sadoun (1997)).
- *The verification*, where a single sample is collected. Various features are extracted from this sample. They are compared to the biometric model of the claimant.

This section first presents the hardware which must be used in order to capture the biometric data, and, the various associated features which can be collected from this data.

3.1 Mandatory hardware and variability

Each biometric modality needs a particular hardware to capture the biometric data. The price of this hardware, as well as the number of sensors to buy, can be determinant when choosing a biometric system supposed to be used in a large infrastructure with number of users (*e.g*, necessity to buy a fingerprint sensor for each computer, if we choose a logical access control for each machine). Keystroke dynamics is probably the biometric modality with the cheapest biometric sensor : it uses only a simple keyboard of your computer. Such keyboard is present in all the personal computers and in all the laptops. If a keyboard is broken and it is necessary to change it, it would cost no more than 5$. Table 2 presents the sensor and its relative price for some modalities, in order to ease the comparison of these systems.

Modality	*keystroke*	*fingerprint*	*face*	*iris*	*hand veins*
Sensor	keyboard	fingerprint sensor	camera	infrared camera	near infra red camera
Price	very cheap	normal	normal	very expensive	expensive

Table 2. Price comparison of hardware for various biometric modalities

Of course, each keyboard is different on various points:

- The shape (straight keyboard, keyboard with a curve, ergonomic keyboard, ...)
- The pressure (how hard it is to press the key)
- The position of keys (AZERTY, QWERTY, ...). Some studies only used the numerical keyboard of a computer (Killourhy & Maxion, 2010; Rodrigues et al., 2006).

Hence, changing a keyboard may affect the performances of the keystroke recognition. This problem is well known in the biometric community and is related as *cross device* matching (Ross & Jain, 2004). It has not been treated a lot in the keystroke dynamics literature. Figure 3 presents the shape of two commonly used keyboards (laptop and desktop). We can see that they are totally different, and, the way of typing on it is also different (maybe mostly due by the red ball on the middle of the laptop keyboard).

(a) Desktop keyboard (b) Laptop keyboard

Fig. 3. Difference of shape of two classical keyboards

Having this sensor (the keyboard) is not sufficient, because (when it is a classical one), the only information it provides is the code of the key pressed or released. This is not at all a biometric information, all the more we already know if it is the correct password or not, whereas we

are interested in if it is the right individual who types it. The second thing we need is an accurate timer, in order to capture at a sufficient precision the time when an event occurs on the keyboard. Once again, this timer is already present in every computer, and, each operating system is able to use it. Hence, we do not need to buy it. There is a drawback with this timer: its resolution can be different depending on the chosen programming language or the operating system. This issue has been extensively discussed by Killourhy & Maxion (2008), where it is shown that better performance are obtained with higher accuracy timer. Some researchers have also studied the effect of using an external clock instead of the one inside the computer. Pavaday. et al. (2010) argue that it is important to take into consideration this timer, especially when comparing algorithms, because it has an impact on performance. They also explain how to configure the operating system in order to obtain the best performances. Even on the same machine, the timer accuracy can be different between the different languages used (by the way, keep in mind, that web based keystroke dynamics implementation use interpreted languages –java or javascript– which are known to not have a precise timer on all the architectures).

Historically, keystroke dynamics works with a classical keyboard on a computer, and avoids the necessity to buy a specific sensor. However, some studies have been done by using other kinds of sensors in order to capture additional information and improve the recognition. Some works (Eltahir et al., 2008; Grabham & White, 2008) have tested the possibility of using a pressure sensor inside each key of the keyboard. In this case, we can exploit an extra information in order to discriminate more easily the users: the pressure force exerced on the key. Lopatka & Peetz (2009) propose to use a keyboard incorporating a Sudden Motion Sensor (SMS)[2]. Such sensor (or similar ones) is present in recent laptops and is used to detect sudden motion of the computer in order to move the writing heads of the hard drive when a risk of damage of the drive is detected. Lopatka & Peetz use the movement in the z axis as information. From these preliminary study, it seems that this information is quite efficient.

Sound signals produced by the keyboard typing have also been used in the literature. Nguyen et al. (2010) only use sound signals when typing the password, and obtain indirectly through the analysis of this signal, key-pressed time, key-released time and key-typed forces. Performance is similar to classical keystroke dynamics systems. Dozono et al. (2007) use the sound information in addition to the timing values (*i.e.*, it is a feature fusion) which held better performance than the sound alone, or the timing information alone. Of course, as keystroke dynamics can work with any keyboard, it can also work with any machine providing a keyboard, or something similar to a keyboard. One common machine having a keyboard and owned by a lot of people is the mobile phone where we can use keystroke dynamics on it. We have three kinds of mobile phones:

- Mobile phone with a numerical keyboard. In this case, it is necessary to press several times the same key in order to obtain an alphabetical character. Campisi et al. (2009) present a study on such a mobile phone. They argue that such authentication mechanism must be coupled with another one.
- Mobile phone with all the keys (letters and numbers) accessible with the thumbs. This is a kind of keyboard quite similar to a computer's keyboard. Clarke & Furnell (2007) show its feasibility and highlight the fact that such authentication mechanism can only be used by regular users of mobile phones.

[2] http://support.apple.com/kb/HT1935

- Mobile phone without any keyboard, but a touch screen. We can argue that the two previous mobile phones are already obsolete and will be soon replaced by such kind of mobile phones. Although, there are few studies on this kind of mobile phone, we think the future of keystroke dynamics is on this kind of material. With such a mobile phone, we can capture the pressure information and position of the finger on the key which could be discriminating.

Figure 4 presents the topology of the different keystroke dynamics sensors, while the Figure 5 presents the variability on the timer.

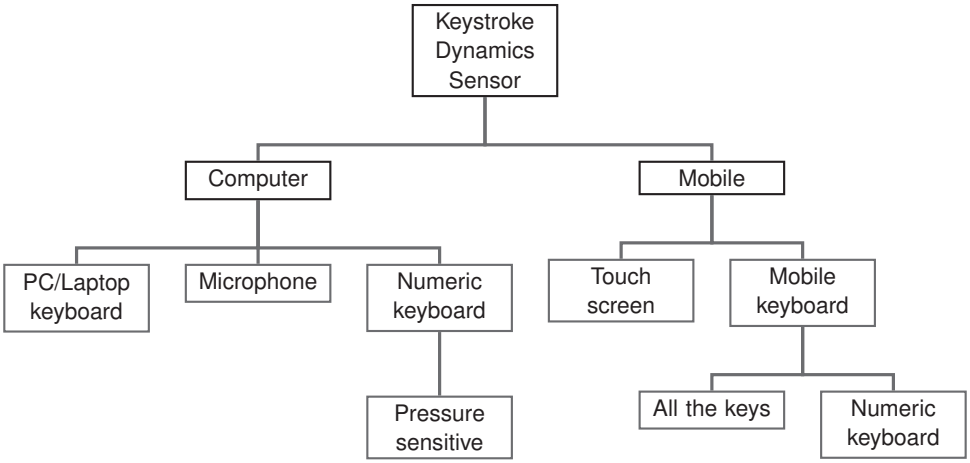

Fig. 4. Topology of keystroke dynamics sensors of the literature

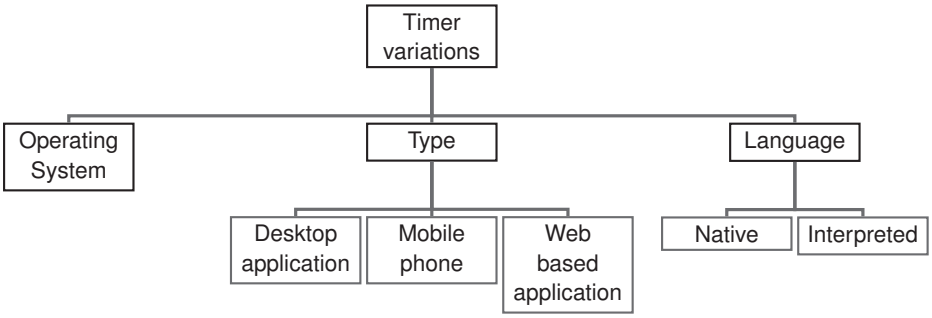

Fig. 5. Topology of factors which may impact the accuracy of the timer

3.2 Captured information

As argued before, various kinds of information can be captured. They mainly depend on the kind of used sensors. Although, we have presented some sensors that are more or less advanced in the previous subsection, we only emphasize, in this chapter, on a classic keyboard.

3.2.1 Raw data

In all the studies, the same raw data is captured (even if they are not manipulated as explained here). We are interested by events on the keyboard. These events are initiated by its user. The raw biometric data, for keystroke dynamics, is a chronologically ordered list of events: the list starts empty, when an event occurs, it is appended at the tail of the list with the following information:

- **Event**. It is generated by an action on the key. There are two different events:
  - **press** occurs when the key is pressed.
  - **release** occurs when the key is released.
- **Key code**. It is the code of the key from which the event occurs. We can obtain the character from this code (in order to verify if the list of characters corresponds to the password, for example). The key code is more interesting than the character, because it gives some information on the location of the key on the keyboard (which can be used by some keystroke dynamics recognition methods) and allows to differentiate different keys giving the same character (which is a discriminant information (Araujo et al., 2005)). This key code may be dependant of the platform and the language used.
- **Timestamp**. It encodes the time when the event occurs. Its precision influence greatly the recognition performance. Pavaday. et al. (2010) propose to use the Windows function $QueryPerformanceCounter$[3] with the highest priority enabled for Windows computers, and, changing the scheduler policy to FIFO for Linux machines. It is usually represented in milliseconds, but this is not mandatory.

The raw data can be expressed as (with n the number of events on the form $n = 2 * s$ with s the number of keys pressed to type the text):

$$\begin{cases} (keycode_i, event_i, time_i), \forall i, 0 <= i < n \\ keycode_i \in \mathbb{Z} \\ event_i \in \{PRESS, RELEASE\} \\ time_i \in \mathbb{N} \end{cases} \quad (1)$$

Umphress & Williams (1985) only use the six first time values of each word (so $s \leq 6$). Depending on the kind of keystroke dynamics application, the raw data is captured in different kind of scenarios: in the authentication form to type the login and password, in a form asking to type a predefined or random text different than the login and password, or in continuous capture during the use of the computer.

3.2.2 Extracted features

Various features can be extracted from this raw data, we present the most commonly used in the literature.

3.2.2.1 First order

The most often extracted features are local ones, computed by subtracting timing values.

- **Duration**. The duration is the amount of time a key is pressed. For the key i (i is omitted for sake of readability) it is computed as following:

$$duration = time\{event = RELEASE\} - time\{event = PRESS\} \quad (2)$$

[3] http://msdn.microsoft.com/en-us/library/ms644904%28v=VS.85%29.aspx

We then obtain a timing vector (of the size of the typed text), also named **PR** in the literature, containing the duration of each key press (by order of press).

$$\forall i, 1 \leq i \leq n, \quad PR_i = duration_i \tag{3}$$

- **Latencies**. Different kinds of latencies can be used. They are computed by getting the differences of time between two keys events. We can obtain the **PP** latencies which are the difference of time between the pressure of each key:

$$\forall i, 1 \leq i < n, \quad PP_i = time_{i+1}\{event_{i+1} = PRESS\} - time_i\{event_i = PRESS\} \tag{4}$$

We can obtain the **RR** latencies which are the difference of time between the release of each key:

$$\forall i, 1 \leq i < n, \quad RR_i = time_{i+1}\{event_{i+1} = RELEASE\} - time_i\{event_i = RELEASE\} \tag{5}$$

We can obtain the **RP** latencies which are the difference of time between the release of one key and the pressure of the next one:

$$\forall i, 1 \leq i < n, \quad RP_i = time_{i+1}\{event_{i+1} = PRESS\} - time_i\{event_i = RELEASE\} \tag{6}$$

Most of the time, a feature fusion is operated by concatenating the duration vector with, at least, one of the latency vector (it seems that most of the time, the selected latency vector is the PP one, but it is not always indicated in the papers). A recent paper Balagani et al. (2011) discusses on the way of using these extracted features in order to improve the recognition rate of keystroke dynamics systems. Other kinds of data can be encountered in various papers Ilonen (2003). They are mainly global types of information:

- **Total typing**. The total time needed to type the text can also be used. The information can be used as an extra feature to append to the feature vectors, or as a normalisation factor.
- **Middle time**. The time difference between the time when the user types the character at the middle of the password, and the time at the beginning of the input.
- **Mistake ratio**. When the user is authorised to do typing mistakes (this is always the case in continuous authentication, but almost never the case in static authentication), counting the number of times the backspace key is hit gives an interesting feature.

Another concept that is often encountered in the literature, is the notion of *digraph*. A digraph represents the time necessary to hit two keys. The digraph features **D** of a password is computed as following:

$$\forall i, 1 \leq i < n, \quad D_i = time_{i+1}\{event_{i+1} = RELEASE\} - time_i\{event_i = PRESS\} \tag{7}$$

This notion has been extended to *ngraph*, with n taking different values. *trigraph* are heavily used in (Bergadano et al., 2002). de Ru & Eloff (1997) use a concept of typing difficulty based on the fact that certain key combinations are more difficult to type than other. The typing difficulty is based on the distance (on the keyboard) between two successive characters (to type), and if several keys are needed to create a character (*i.e.*, use of shift key).

3.2.2.2 Second order

Some features are not extracted from the raw biometric data, but from the first order features.

- **min/max**. It consists to get the minimum and maximum value of each type of data (latency and duration).
- **mean/std**. It consists to get the mean value and its standard deviation of each type of data (latency and duration).
- **Slope**. By using the slope of the biometric sample, we are interested in the global shape of the typing. We expect that users type in the same way even if the speed may be different (Modi & Elliott, 2006). The new features ($\boldsymbol{result}$) set is computed as following (with $\boldsymbol{source}$):

$$\forall i, 1 \leq i < n, \quad result_i = source_{i+1} - source_i \tag{8}$$

- **Entropy**. The entropy inside a sample has been only studied in (Monrose et al., 2002).
- **Spectral information**. Chang (2006a) applies a discrete wavelet transformation to the original extracted features. All the operations are done with the wavelet transformed data.

We can imagine more complicated features, but the final biometric data is always a single vector composed of various features. While computing the model with several samples (see next section) feature selection mechanisms can remove non informative features. We do not insist on papers using other information than timing values in the rest of this chapter (pressure force, movements, ...). We have seen in this section that several features can be extracted. Verification procedures performance greatly depends on the chosen features, but, most of the time, papers only use one latency and the duration.

4. Authentication framework

Once the different biometric data during enrolment procedure have been captured, it is time to build the model of each user. The way of computing it greatly depends on the used verification methods. During an authentication, the verification method compares the query sample (the biometric data captured during the authentication) to the model. Based on the result of this comparison (which is commonly a distance), the decision module accepts or rejects the user.

4.1 Enrolment

The enrolment step allows to create the model of each user, thanks to its enrolled samples. Most of the time, the number of samples used during the enrolment is superior to 20. Such a high quantity of data can be really boring for the users to provide.

4.1.1 Outliers detection

It is known that the classifier performance greatly depends on outliers presence in the learning dataset. Most keystroke dynamics studies do not take care of the presence of outliers in the learning set. Some studies (mainly in free text) remove times superior to a certain threshold. In (Gaines et al., 1980), filtering is done by removing timing values superior to 500ms, while in (Umphress & Williams, 1985) it is timing values superior to 750ms. Rogers & Brown (1996) cleanup data with using a Kohonen network (Kohonen, 1995) using impostors samples. They also use a statistical method. Killourhy & Maxion (2010) also detect outliers in biometric samples. An outlier feature is detected in the following way (for each feature): the feature is more than 1.5 inter-quartile range greater than the third quartile, or more than 1.5

inter-quartile less than the first quartile. When a feature is detected as being an outlier, it is replaced by a random sample (which is not an outlier) selected among possible values of this feature for this user. The procedure is operated for each feature of each sample. By this way, the number of samples is always the same.
It seems that, most of the time, the outlier detection and correction is operated on the whole dataset, and not on the learning set. This allows to cleanup the used dataset to compute the algorithm performance (and obtain better performance), but not the enrolled samples of the user.

4.1.2 Preprocessing

Biometric data may be normalized before being used. Such pre-processing allows to get better performance by using a normalisation function (Filho & Freire (2006) observed that the timing distribution is roughly Log-Normal) :

$$g(x) = \frac{1}{1 + exp\left(-\frac{K(log_e(x) - \mu}{\sigma}\right)} \tag{9}$$

We did not find other references to other pre-processing approaches in the literature. The parameters K (k is chosen in order to minimise the squared error between the approximated function and the cumulative distribution function of the logarithm of timings distribution), μ and $sigma$ respectively represent an optimisation factor, the mean of the logarithm of the timing values, the standard deviation of the logarithm of the timing values.

4.1.3 Feature selection

A feature selection mechanism can be applied to remove irrelevant features. It seems that this point has also been rarely tested. The aim of the feature selection is to reduce the quantity of data and speed up the computation time, and, eventually to improve the performance. Very few studies have applied such kind of mechanism. Two different kinds of feature extraction systems can be used:

- *Filter* approach which does not depend on the verification algorithm. The aim is to remove irrelevant features based on different measures (*e.g.*, the variance);
- *Wrapper* approach which depends on the verification algorithm. Different feature subsets are generated and evaluated. The best one is kept.

Boechat et al. (2006) select a subset of N features with the minors of standard deviation, which allows to eliminate less significant features. Experiments are done at Zero False Acceptance Rate. False Rejection Rate reduces when the number of selected features increases. Keeping 70% of the features gives interesting results. Azevedo et al. (2007) use a wrapper system based on Particle Swarm Optimization (PSO) to operate the feature selection. The PSO gives better results than a Genetic Algorithm. Bleha & Obaidat (1991) use a reduction technique based on Fisher analysis. However, the technique consists in keeping $m - 1$ dimension for each vector, with m the number of users in the system (they have only 9 users in their system). Yu & Cho (2004) use an algorithm based an Support Vector Machines (SVM) and Genetic Algorithms (GA) to reduce the size of samples and keep only key values for each user. Other similar methods are present in the literature (Chen & Lin, 2005).

4.1.4 Model computation

There are numbers of methods to verify if a query corresponds to the expected user. Some of them are based on statistical methods, other on data mining methods. Some methods use one-class assumption (they only use the enrolment samples of the user), while other use two-class or multi-class assumption (they also use impostors enrollment samples to compute the model). When impostors samples are needed, they may be automatically generated (Sang et al., 2004), instead of being collected with real impostors (Clarke & Furnell, 2006; Obaidat & Sadoun, 1997). Generally, data mining methods use a really huge number of enrolled samples to compute the model (several hundred of samples in some neural network methods) which is not realistic at all. Most used way of model computing are:

- Computing the mean vector and standard deviation of enrolled samples (Umphress & Williams, 1985);
- Store the enrolled vectors in order to use them with k nearest neighbourg methods (Rao, 2005) (variations being in the distance computing method (Kang & Cho, 2009));
- Learning of bayesian classifiers (Janakiraman & Sim, 2007; Rao, 2005);
- Learning clusters with *k-mean* (Hwang et al., 2006; Obaidat & Sadoun, 1997) ;
- Learning parameters of generative functions: Hidden Markov Model (HMM) (Galassi et al., 2007; Pohoa et al., 2009; Rodrigues et al., 2006) or Gaussian Mixture Models (GMM) (Hosseinzadeh & Krishnan, 2008) ;
- Neural network learning (Bartmann et al., 2007; Clarke & Furnell, 2006; Obaidat & Sadoun, 1997; Rogers & Brown, 1996) ;
- *SVM* learning (Giot et al., 2009c; Rao, 2005; Sang et al., 2004; Yu & Cho, 2004).

4.2 Verification

The verification consists in verifying if the input of the user corresponds to the claimed identity. The way of capturing these inputs greatly depends on the kind of used keystroke dynamics system (*e.g.*, for static authentication, the user must type its login and password). While the features are extracted from the raw biometric sample (same procedure than during the enrollment), they are compared to the model of the claimed user. Usually, the verification module returns a comparison score. If this score is below than a predefined threshold, the user is authenticated, otherwise, he is rejected. Several verification methods exist and depend on the way the enrollment is done, so they are similar to the present list. $\boldsymbol{query}$ represents the query biometric sample (the test capture to compare to the model). $\|.\|_p$ represents the p norm of vector. The main families of computing are (Guven & Sogukpinar, 2003):

- The minimal distance computing.

  In (Monrose & Rubin, 1997), the euclidean distance between the query and each of enrolled samples is computed. The comparison score is the min of these distances.

$$score = \min \|\boldsymbol{query} - \boldsymbol{enrolled}_u\|_2, \forall_{u \in [1, Card(enrolement)]} \tag{10}$$

- The statistical methods.

One of the oldest methods is based on bayesian probabilities (Bleha et al., 1990). $\boldsymbol{\mu}$ is the mean value of enrolled samples:

$$score = \frac{(\boldsymbol{query} - \boldsymbol{\mu})^t (\boldsymbol{query} - \boldsymbol{\mu})}{\|\boldsymbol{query}\|_2 \cdot \|\boldsymbol{\mu}\|_2} \tag{11}$$

A normalized version is also presented in the study. The statistical method presented in (Hocquet et al., 2006) computes the score depending on the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ of the enrolled samples:

$$score = 1 - \frac{1}{Card(\boldsymbol{query})} \left\| exp\left(- \frac{|\boldsymbol{query} - \boldsymbol{\mu}|}{\boldsymbol{\sigma}} \right) \right\|_1 \tag{12}$$

Filho & Freire (2006) present another method which also computes a distance:

$$score = \|\boldsymbol{query} - \boldsymbol{\mu}\|_2^2 \tag{13}$$

- Application of fuzzy rules de Ru & Eloff (1997).
- Class verification.
  For classifiers able to give a label, the verification consists in verifying if the guessed label corresponds to the label of the claimed identity (cf. neural networks, SVM, $k - nn$).
- Some methods are based on the disorder degree of vectors (Bergadano et al., 2002).
- Others are based on timing discretisation (Hocquet et al., 2006).
- Bioinformatic methods based on string motif searching are also used (Revett et al., 2007).

4.3 Improving the performance

Different ways can be used to improve the performance of the recognition. Several studies (Bartmann et al., 2007; Hosseinzadeh & Krishnan, 2008; Killourhy & Maxion, 2010; Revett, 2009) request the user to type the verification text several times (mainly between two and three), when he is rejected, in order to give him more chances of being verified. Such procedure reduces the False Rejection Rate without growing the False Acceptance Rate too much. Other studies try to update the model of a user after being authenticated (Hosseinzadeh & Krishnan, 2008; Revett, 2009). This way, the model tracks the behavior modifications of the user through time, and integrate them in the model. As the keystroke data deviates progressively with time, performance degrades with time when not using such procedure. It is not always clear in the various studies if the template update is done in a supervised way (impostors samples never added), or in a semi-supervised way (samples added if the classifier recognizes them as being genuine). Even if the aim is to improve performance, the result can be totally different: semi-supervised methods may add impostor samples in the model. This way, the model deviates from the real biometric data of the user and attracts more easily impostors samples. Classifier performances greatly depend on the number of used samples to compute them. Chang (2006b) artificially generates new samples from the enrolled samples in order to improve keystroke recognition. The system uses a transformation in frequential domains thanks to wavelets. Another way to improve recognition performance is to fuse two samples together (Bleha & Obaidat, 1991). This way, timing values are smoothed when merging the two samples and light hesitation are suppressed. The fusion (Ross et al., 2006) of several keystroke dynamics methods on the same query is also a good way to improve performances:

- Bleha et al. (1990) associate a bayesian classifier to a minimal distance computing between the query vector and the model.
- Hocquet et al. (2006) apply a fusion between three different keystroke dynamics methods, which greatly improves the performance.
- Different kinds of weighted sums score fusion functions are proposed in Giot, El-Abed & Rosenberger (2010); Teh et al. (2007).

Keystroke dynamics has also been successfully fused with other modalities, like face (Giot, Hemery & Rosenberger, 2010) or speaker recognition (Montalvao Filho & Freire, 2006). Hwang et al. (2006) have defined various measures to get the unicity, consistancy and discriminality. By analysing the behavior of these measures comparing the recognition performance, they find that it is possible to improve performance by asking users to artificially add pauses (helped by cues for being synchronized) when typing the password. Karnan et al. (2011) propose an interesting review of most of the keystroke dynamics recognition methods.

4.4 User identification

The verification consists in verifying if the identity of the claimant is correct, while the identification consists to determine the identity of the user. We may find methods specifics to identification, or compare the query to each model, the identity being the owner of the model returning the lowest distance (or a reject if this distance is higher a threshold). Bleha et al. (1990) use a bayesian classifier to identify the user. Identification based on keystroke dynamics has not been much experimented in the literature.

5. Evaluation of keystroke dynamics systems

Despite the obvious advantages of keystroke dynamics systems in enhancing traditional methods based on a *secret*, its proliferation is still not as much as expected. The main drawback is notably the lack of a generic evaluation method for such systems. We need a reliable evaluation methodology in order to put into obviousness the benefit of a new method. Nowadays, several studies exist in the state-of-the-art to evaluate keystroke dynamics systems. It is generally realized within three aspects: performance, satisfaction and security.

5.1 Performance

The goal of this evaluation aspect is to quantify and to compare keystroke dynamics systems. In order to compare these systems, we need generally to compute their performance using a predefined protocol (acquisition conditions, test database, performane metrics, ...). According to the International Organization for Standardization ISO/IEC 19795-1 (2006), the performance metrics are divided into three sets:

- Acquisition performance metrics such as the Failure-To-Enroll rate (FTE).
- Verification system performance metrics such as the Equal Error Rate (EER).
- Identification system performance metrics such as the False-Negative and the False-Positive Identification Rates (FNIR and FPIR, respectively).

Several benchmark databases exist in order to compare keystroke dynamics systems. A benchmark database can contain real samples from individuals, which reflect the best the real use cases. Nevertheless, it is costly in terms of efforts and time to create such a database. As argued by Cherifi et al. (2009), a good benchmark database must satisfy various requirements:

1. As keystroke dynamics is a behavioral modality, the database must be captured among different sessions, with a reasonable time interval between sessions, in order to take into account the variation of individuals behavior.
2. The database must also contain fake biometric templates to test the robustness of the system. It seems that there is no other reference to this kind of experiment in the literature.
3. The benchmark must embed a large diversity of users (culture, age, ...). This point is essential for any biometrics, but, it is really difficult to attain.

We present an overview of the existing benchmark databases:

DB 1 Chaves
Montalvão *et al.* have used the same keystroke databases in several papers (Filho & Freire, 2006). The databases are available at `http://itabi.infonet.com.br/biochaves/br/download.htm`. The databases do not seem to be yet available on their website. The maximum number of users in a database is 15, and, the number of provided samples per user is 10. Each database contains the raw data. The database is composed of couples of ASCII code of the pressed key and the elapsed time since the last key down event. Release of a key is not tracked. Four different databases have been created. Most databases were built under two different sessions spaced of one week or one month (depending on the database). Each database is stored in raw text files.

DB 2 DSN2009
Killourhy & Maxion (2009) propose a database of 51 users providing four hundred samples captured in height sessions (there are fifty inputs per session). The delay between each session is one one day at minimum, but the mean value is not stated. This is the dataset having the most number of samples per user, but, a lot of them are typed on a short period (50 at the same time). Each biometric data has been captured when typing the following password: ".tie5Roanl". The database contains some extracted features: hold time, interval between two pressures, interval between the release of a key, and the pressure of the next one. The database is available at `http://www.cs.cmu.edu/~keystroke/`. It is stored in raw text, csv or Excel files.

DB 3 Greyc alpha
Giot et al. (2009a) propose the most important public dataset in term of users. It contains 133 users and, 100 of them provided samples of, at least, five distinct sessions. Each user typed the password "greyc laboratory" twelve times, on two distinct keyboards, during each session (which give 60 samples for the 100 users having participated to each session). Both extracted features (hold time and latencies) and raw data are available (which allow to build other extracted features). The database is available at `http://www.ecole.ensicaen.fr/~rosenber/keystroke.html`. It is stored in an sqlite database file.

DB 4 Pressure-Sensitive Keystroke Dynamics Dataset
Allen (2010) has created a public keystroke dynamics database using a pressure sensitive keyboard. The database is available at `http://jdadesign.net/2010/04/pressure-sensitive-keystroke-dynamics-dataset/` in a csv or sql file. It embeds the following raw data: key code, time when pressed, time when release, pressure force. 104 users are present on the database, but, only 7 of them provided a significant amount of data (between 89 and 504), whereas the 97 other have only provided between 3 and 15 samples. Three different passwords have been typed: "pr7q1z", "jeffrey allen" and "drizzle".

DB 5 Fixed Text

The most recent database has been released in 2010 Bello et al. (2010). 58 volunteers participated to the experiment. Each session consists in typing 14 phrases extracted from books and 15 common UNIX commands. It seems that almost all the users have done only one session. The database is available at `http://www.citefa.gov.ar/si6/k-profiler/dataset/` in a raw text file. Press and release times for each key are saved, as well as the user agent of the browser from which the session has been done, the age, gender and handness of the user and other information.

We can see that some databases are available. Each of them has been created for keystroke dynamics on computer (*i.e.* no public dataset available for smartphones). Despite this, these databases do not always fit the previous requirements, which may explain why none of them have been used by researchers different than their creators. Although, it would be the best kind of dataset, no public dataset has been built with one login/password different for each user. Table 3 presents a summary of these public datasets.

Dataset	Type	Information	Users	Samples /users	Sessions
Filho & Freire (2006)	Various	Press events	< 15	< 10	2
Killourhy & Maxion (2009)	1 fixed string	Duration and 2 latencies	51	400	8
Giot et al. (2009a)	1 fixed String	Press and release events. Duration and 3 latencies	> 100	60	5
Allen (2010)	3 fixed strings	Press and release events and pressure	7/97	(89-504) /(3-15)	few months
Bello et al. (2010)	14 phrases and 15 unix commands	Press and release time	58	1	1

Table 3. Summary of keystroke dynamics datasets

Most of the proposed keystroke dynamics methods in the literature have quantified their methods using different protocols for their data acquisition (Giot et al., 2009c; Killourhy & Maxion, 2009). Table 4 illustrates the differences of the used protocols in this research area for some major studies. The performance comparison of these methods is quite impossible, as stated in (Crawford, n.d.; Giot et al., 2009a; Karnan et al., 2011; Killourhy & Maxion, 2009), due to several reasons. First, most of these studies have used different protocols for their data acquisition, which is totally understandable due to the existence of different kinds of keystroke dynamics systems (static, continuous, dynamic) that require different acquisition protocols. Second, they differ on the used database (number of individuals, separation between sessions ...), the *acknowledgement* of the password (if it is an imposed password, a high *FTA* is expected), the used keyboards (which may deeply influences the way of typing), and the use of different or identical passwords (which impacts on the quality of impostors' data). In order to resolve such problematic, Giot et al. (2011) presents a comparative study of seven methods (1 contribution against 6 methods existing in the literature) using a predefined protocol, and GREYC alpha database (Giot et al., 2009a). The results from this study show a promising EER value equal to to 6.95%. To our knowledge, this is the only work that compares

keystroke methods within the same protocol, and using a publicly available database. The performance of keystroke dynamics systems (more general speaking, of behavioral systems) provides a lower quality than the morphological and biological ones, because they depend a lot on user's feelings at the moment of the data acquisition: user may change his way of performing tasks due to its stress, tiredness, concentration or illness. Previous works presented by Cho & Hwang (2006); Hwang et al. (2006) focus on improving the quality of the captured keystroke features as a mean to enhance system overall performance. Hwang et al. (2006) have employed pauses and cues to improve the uniqueness and consistency of keystroke features. We believe that it is relevant to more investigate the quality of the captured keystroke features, in order to enhance the performance of keystroke dynamics systems.

Paper	A	B	C	D	E	*FAR*	*FRR*
Obaidat & Sadoun (1997)	8 weeks	15	112	no	no	0%	0%
Bleha et al. (1990)	8 weeks	36	30	yes	yes	2.8%	8.1%
Rodrigues et al. (2006)	4 sessions	20	30	/	no	3.6%	3.6%
Hocquet et al. (2007)	/	38	/	/	no	1.7%	2.1%
Revett et al. (2007)	14 days	30	10	/	no	0.15%	0.2%
Hosseinzadeh & Krishnan (2008)	/	41	30	no	no	4.3%	4.8%
Monrose & Rubin (1997)	7 weeks	42	/	no	no	/	20%
Revett et al. (2006)	4 weeks	8	12	/	/	5.58%	5.58%
Killourhy & Maxion (2009)	8 sessions	51	200	yes	no	9.6%	9.6%
Giot et al. (2009c)	5 sessions	100	5	yes	no	6.96%	6.96%

Table 4. Summary of the protocols used for different studies in the state-of-the-art (A: Duration of the database acquisition, B: Number of individuals in the database, C: Number of samples required to create the template, D: Is the acquisition procedure controlled?, E: Is the threshold global?). "/" indicates that no information is provided in the article.

5.2 Satisfaction

This evaluation aspect focuses on measuring users' acceptance and satisfaction regarding the system (Theofanos et al., 2008). It is generally measured by studying several properties such as easiness to use, trust in the system, *etc.* The works done by El-Abed et al. (2010); Giot et al. (2009b) focusing on studying users' acceptance and satisfaction of a keystroke dynamics system (Giot et al., 2009a), show that the system is well perceived and accepted by the users. Figure 7 summarizes users' acceptance and satisfaction while using the tested system. Satisfaction factors are rated between 0 and 10 (0 : not satisfied $\cdots$ 10 : quite satisfied). These results show that the tested system is well perceived among the five acceptance and satisfaction properties. Moreover, there were no concerns about privacy issues during its use. In biometrics, there is a potential concern about the misuse of personal data (*i.e.*, templates) which is seen as violating users' privacy and civil liberties. Hence, biometric systems respecting this satisfaction factor are considered as usefull.

5.3 Security

Biometric authentication systems present several drawbacks which may considerably decrease their security. Schneier (1999) compares traditional security systems with biometric systems. The study presents several drawbacks of biometric systems including:

- The lack of secrecy: everybody knows our biometric traits such as iris,
- and, the fact that a biometric trait cannot be replaced if it is compromised.

El-Abed et al. (2011) propose an extension of the Ratha *et al.* model (Ratha et al., 2001) to categorize the common threats and vulnerabilities of a generic biometric system. Their proposed model is divided into two sets as depicted in figure 6: architecture threats and system overall vulnerabilities.

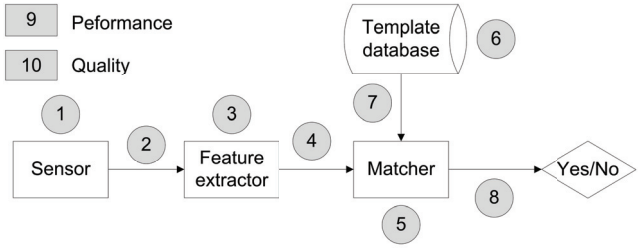

Fig. 6. Vulnerability points in a general biometric system.

5.3.1 Set I architecture threats

1) Involves presenting a fake biometric data to the sensor. An example of such attack is the zero-effort attempts. Usually, attackers try to impersonate legitimate users having weak templates;

2) and **4)** In a replay attack, an intercepted biometric data is submitted to the feature extractor or the matcher bypassing the sensor. Attackers may collect then inject previous keystroke events features using a keylogger;

3) and **5)** The system components are replaced with a Trojan horse program that functions according to its designer specifications;

6) Involves attacks on the template database such as modifying or suppresing keystroke templates;

7) The keystroke templates can be altered or stolen during the transmission between the template database and the matcher;

8) The matcher result (accept or reject) can be overridden by the attacker.

5.3.2 Set II system overall vulnerabilities

9) Performance limitations

By contrast to traditional authentication methods based on "what we know" or "what we own" (0% comparison error), biometric systems is subject to errors such as False Acceptance Rate (FAR) and False Rejection Rate (FRR). This inaccuracy illustrated by statistical rates would have potential implications regarding the level of security provided by a biometric system. Doddington et al. (1998) assign users into four categories:

- Sheep: users who are recognized easily (contribute to a low FRR),
- Lambs: users who are easy to imitate (contribute to a high FAR),
- Goats: users who are difficult to recognize (contribute to a high FRR), and
- Wolves: users who have the capability to spoof the biometric characteristics of other users (contribute to a high FAR).

A poor biometric in term of performance, may be easily attacked by lambs, goats and wolves users. There is no reference to this user classification in the keystroke dynamics literature. Therefore, it is important to take into consideration system performance within the security evaluation process. The Half Total Error Rate (HTER) may be used as an illustration of system overall performance. It is defined as the mean of both error rates FAR and FRR:

$$HTER = \frac{FAR + FRR}{2} \tag{14}$$

10) Quality limitations during enrollment
The quality of the acquired biometric samples is considered as an important factor during the enrollment process. The absence of a quality test increases the possibility of enrolling authorized users with weak templates. Such templates increase the probability of success of zero-effort impostor, hill-climbing and brute force (Martinez-Diaz et al., 2006) attempts. Therefore, it is important to integrate such information within the security evaluation process. In order to integrate such information, a set of rules is presented in (El-Abed et al., 2011).

According to the International Organization for Standardization ISO/IEC FCD 19792 (2008), the security evaluation of biometric systems is generally divided into two complementary assessments:

1. Assessment of the biometric system (devices and algorithms), and
2. Assessment of the environmental (for example, is the system is used indoor or outdoor?) and operational conditions (for example, tasks done by system administrators to ensure that the claimed identities during enrollment of the users are valid).

A type-1 security assessment of a keystroke dynamics system (Giot et al., 2009a) is presented in El-Abed et al. (2011). The presented method is based on the use of a database of common threats and vulnerabilities of biometric systems, and the notion of risk factor. A risk factor, for each identified threat and vulnerability, is considered as an indicator of its importance. It is calculated using three predefined criteria (effectiveness, easiness and cheapness) and is defined between 0 and 1000. More the risk factor is near 0, better is the robustness of the Target of Evaluation (ToE). Figure 7 summarizes the security assessment of the TOE, which illustrates the risk factors of the identified threats and system overall vulnerabilities among the ten assessment points (the maximal risk factor is retained from each point).

5.4 Discussion

The evaluation of keystroke dynamics modality are very few in comparison to other types of modalities (such as fingerprint modality). As shown in section 5.1, there is only a few public databases that could be used to evaluate keystroke dynamics authentication systems. There is none competition neither existing platform to compare such behavioral modality. The results presented in the previous section show that the existing keystroke dynamics methods provide promising recognition rates, and such systems are well perceived and accepted by users. In our opinion, we believe that keystroke dynamics systems belong to the possible candidates that may be implemented in an Automated Teller Machine (ATM), and can be widely used for e-commerce applications.

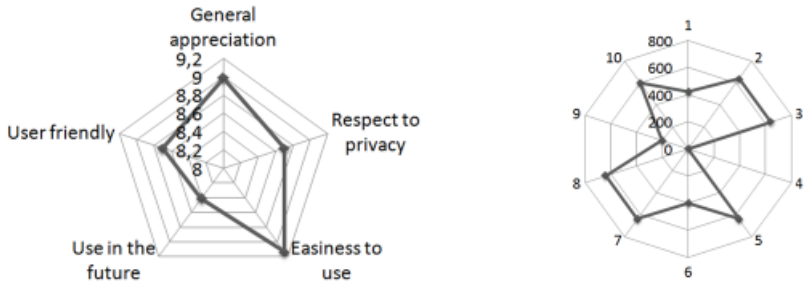

Fig. 7. Satisfaction (on the left) and security (on the right) assessment of a keystroke dynamics based system.

6. Conclusion and future trends

We have presented in this chapter an overview of keystroke dynamics literature. More information on the subject can be found in various overviews: Revett (2008, chapter 4) deeply presents some studies. We believe that the future of the keystroke dynamics is no more on desktop application, whereas it is the most studied in the literature, but in the *mobile* and *internet* worlds, because mobile phones are more popular than computers and its use is very democratized. They are more and more powerful every year (in terms of calculation and memory) and embeds interesting sensors (pressure information with tactile phones). Mobile phone owners are used to use various applications on their mobile and they will probably agree to lock them with a keystroke dynamics biometric method. Nowadays, more applications are available in a web browser. These applications use the classical couple of login and password to verify the identity of a user. Integrating them a keystroke dynamics verification would harden the authentication process. In order to spread the keystroke modality, it is necessary to solve various problems related to:

- The cross devices problem. We daily use several computers which can have different keyboards on timing resolution. These variability must not have an impact on the recognition performances. Users tend to change often their mobile phone. In an online authentication scheme (were the template is stored on a server), it could be useful to not re-enroll the user on its new mobile phone.
- The aging of the biometric data. Keystroke dynamics, is subject to a lot of intra class variability. One of the main reasons is related to the problem of template aging: performances degrade with time because user (or impostors) type differently with time.

7. Acknowledgment

The authors would like to thank the Lower Normandy Region and the French Research Ministry for their financial support of this work.

8. References

Ahmed, A. & Traore, I. (2008). *Handbook of Research on Social and Organizational Liabilities in Information Security*, Idea Group Publishing, chapter Employee Surveillance based on Free Text Detection of Keystroke Dynamics, pp. 47–63.

Allen, J. D. (2010). *An analysis of pressure-based keystroke dynamics algorithms*, Master's thesis, Southern Methodist University, Dallas, TX.

Araujo, L., Sucupira, L.H.R., J., Lizarraga, M., Ling, L. & Yabu-Uti, J. (2005). User authentication through typing biometrics features, *IEEE Transactions on Signal Processing* 53(2 Part 2): 851–855.

Azevedo, G., Cavalcanti, G., Carvalho Filho, E. & Recife-PE, B. (2007). An approach to feature selection for keystroke dynamics systems based on pso and feature weighting, *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*.

Balagani, K. S., Phoha, V. V., Ray, A. & Phoha, S. (2011). On the discriminability of keystroke feature vectors used in fixed text keystroke authentication, *Pattern Recognition Letters* 32(7): 1070 – 1080.

Bartmann, D., Bakdi, I. & Achatz, M. (2007). On the design of an authentication system based on keystroke dynamics using a predefined input text, *Techniques and Applications for Advanced Information Privacy and Security: Emerging Organizational, Ethical, and Human Issues* 1(2): 149.

Bello, L., Bertacchini, M., Benitez, C., Carlos, J., Pizzoni & Cipriano, M. (2010). Collection and publication of a fixed text keystroke dynamics dataset, *XVI Congreso Argentino de Ciencias de la Computacion (CACIC 2010)*.

Bergadano, F., Gunetti, D. & Picardi, C. (2002). User authentication through keystroke dynamics, *ACM Transactions on Information and System Security (TISSEC)* 5(4): 367–397.

Bleha, S. & Obaidat, M. (1991). Dimensionality reduction and feature extraction applications inidentifying computer users, *IEEE transactions on systems, man and cybernetics* 21(2): 452–456.

Bleha, S., Slivinsky, C. & Hussien, B. (1990). Computer-access security systems using keystroke dynamics, *IEEE Transactions On Pattern Analysis And Machine Intelligence* 12 (12): 1216–1222.

Boechat, G., Ferreira, J. & Carvalho, E. (2006). Using the keystrokes dynamic for systems of personal security, *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 18, pp. 200–205.

Campisi, P., Maiorana, E., Lo Bosco, M. & Neri, A. (2009). User authentication using keystroke dynamics for cellular phones, *Signal Processing, IET* 3(4): 333 –341.

Chang, W. (2006a). Keystroke biometric system using wavelets, *ICB 2006*, Springer, pp. 647–653.

Chang, W. (2006b). Reliable keystroke biometric system based on a small number of keystroke samples, *Lecture Notes in Computer Science* 3995: 312.

Chen, Y.-W. & Lin, C.-J. (2005). Combining svms with various feature selection strategies, *Technical report*, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan.

Cherifi, F., Hemery, B., Giot, R., Pasquet, M. & Rosenberger, C. (2009). *Behavioral Biometrics for Human Identification: Intelligent Applications*, IGI Global, chapter Performance Evaluation Of Behavioral Biometric Systems, pp. 57–74.

Cho, S. & Hwang, S. (2006). Artificial rhythms and cues for keystroke dynamics based authentication, *In International Conference on Biometrics (ICB)*, pp. 626–632.

Clarke, N. & Furnell, S. (2006). Advanced user authentication for mobile devices, *computers & security* 27: 109–119.

Clarke, N. L. & Furnell, S. M. (2007). Authenticating mobile phone users using keystroke analysis, *International Journal of Information Security* 6: 1–14.

Conklin, A., Dietrich, G. & Walz, D. (2004). Password-based authentication: A system perspective, *Proceedings of the 37th Hawaii International Conference on System Sciences, Hawaii*.

Crawford, H. (n.d.). Keystroke dynamics: Characteristics and opportunities, *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on*, IEEE, pp. 205–212.

de Magalhaes, T., Revett, K. & Santos, H. (2005). Password secured sites: stepping forward with keystroke dynamics, *International Conference on Next Generation Web Services Practices*.

de Ru, W. G. & Eloff, J. H. P. (1997). Enhanced password authentication through fuzzy logic, *IEEE Expert: Intelligent Systems and Their Applications* 12: 38–45.

Doddington, G., Liggett, W., Martin, A., Przybocki, M. & Reynolds, D. (1998). Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation, *ICSLP98*.

Dozono, H., Itou, S. & Nakakuni, M. (2007). Comparison of the adaptive authentication systems for behavior biometrics using the variations of self organizing maps, *International Journal of Computers and Communications* 1(4): 108–116.

El-Abed, M., Giot, R., Hemery, B. & Rosenberger, C. (2010). A study of users' acceptance and satisfaction of biometric systems, *44th IEEE International Carnahan Conference on Security Technology (ICCST)*.

El-Abed, M., Giot, R., Hemery, B., Shwartzmann, J.-J. & Rosenberger, C. (2011). Towards the security evaluation of biometric authentication systems, *IEEE International Conference on Security Science and Technology (ICSST)*.

Eltahir, W., Salami, M., Ismail, A. & Lai, W. (2008). Design and Evaluation of a Pressure-Based Typing Biometric Authentication System, *EURASIP Journal on Information Security, Article ID* 345047(2008): 14.

Epp, C. (2010). *Identifying emotional states through keystroke dynamics*, Master's thesis, University of Saskatchewan, Saskatoon, CANADA.

Filho, J. R. M. & Freire, E. O. (2006). On the equalization of keystroke timing histograms, *Pattern Recognition Letters* 27: 1440–1446.

Gaines, R., Lisowski, W., Press, S. & Shapiro, N. (1980). Authentication by keystroke timing: some preliminary results, *Technical report*, Rand Corporation.

Galassi, U., Giordana, A., Julien, C. & Saitta, L. (2007). Modeling temporal behavior via structured hidden markov models: An application to keystroking dynamics, *Proceedings 3rd Indian International Conference on Artificial Intelligence (Pune, India)*.

Giot, R., El-Abed, M. & Chri (2011). Unconstrained keystroke dynamics authentication with shared secret, *Computers & Security* pp. 1–20. [in print].

Giot, R., El-Abed, M. & Rosenberger, C. (2009a). Greyc keystroke: a benchmark for keystroke dynamics biometric systems, *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2009)*, IEEE Computer Society, Washington, District of Columbia, USA, pp. 1–6.

Giot, R., El-Abed, M. & Rosenberger, C. (2009b). Keystroke dynamics authentication for collaborative systems, *International Symposium on Collaborative Technologies and Systems*, pp. 172–179.

Giot, R., El-Abed, M. & Rosenberger, C. (2009c). Keystroke dynamics with low constraints svm based passphrase enrollment, *IEEE International Conference on Biometrics: Theory,*

Applications and Systems (BTAS 2009), IEEE Computer Society, Washington, District of Columbia, USA, pp. 1–6.

Giot, R., El-Abed, M. & Rosenberger, C. (2010). Fast learning for multibiometrics systems using genetic algorithms, *The International Conference on High Performance Computing & Simulation (HPCS 2010)*, IEEE Computer Society, Caen, France, pp. 1–8.

Giot, R., Hemery, B. & Rosenberger, C. (2010). Low cost and usable multimodal biometric system based on keystroke dynamicsand 2d face recognition, *IAPR International Conference on Pattern Recognition (ICPR)*, IAPR, Istanbul, Turkey, pp. 1128–1131. Acecptance rate: 54/100.

Giot, R. & Rosenberger, C. (2011). A new soft biometric approach for keystroke dynamics based on gender recognition, *Int. J. of Information Technology and Management (IJITM), Special Issue on: "Advances and Trends in Biometric* pp. 1–17. [in print].

Grabham, N. & White, N. (2008). Use of a novel keypad biometric for enhanced user identity verification, *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, pp. 12–16.

Guven, A. & Sogukpinar, I. (2003). Understanding users' keystroke patterns for computer access security, *Computers & Security* 22(8): 695–706.

Hocquet, S., Ramel, J.-Y. & Cardot, H. (2006). Estimation of user specific parameters in one-class problems, *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, pp. 449–452.

Hocquet, S., Ramel, J.-Y. & Cardot, H. (2007). User classification for keystroke dynamics authentication, *The Sixth International Conference on Biometrics (ICB2007)*, pp. 531–539.

Hosseinzadeh, D. & Krishnan, S. (2008). Gaussian mixture modeling of keystroke patterns for biometric applications, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 38(6): 816–826.

Hwang, S.-s., Lee, H.-j. & Cho, S. (2006). Improving authentication accuracy of unfamiliar passwords with pauses and cues for keystroke dynamics-based authentication, *Intelligence and Security Informatics* 3917: 73–78.

Ilonen, J. (2003). Keystroke dynamics, *Advanced Topics in Information Processing–Lecture* .

ISO/IEC 19795-1 (2006). Information technology biometric performance testing and reporting, *Technical report*, International Organization for Standardization ISO/IEC 19795-1.

ISO/IEC FCD 19792 (2008). Information technology – security techniques –security evaluation of biometrics, *Technical report*, International Organization for Standardization ISO/IEC FCD 19792.

Janakiraman, R. & Sim, T. (2007). Keystroke dynamics in a general setting, *Lecture notes in computer science* 4642: 584.

Kang, P. & Cho, S. (2009). A hybrid novelty score and its use in keystroke dynamics-based user authentication, *Pattern Recognition* p. 30.

Karnan, M., Akila, M. & Krishnaraj, N. (2011). Biometric personal authentication using keystroke dynamics: A review, *Applied Soft Computing* 11(2): 1565 – 1573. The Impact of Soft Computing for the Progress of Artificial Intelligence.

Khanna, P. & Sasikumar, M. (2010). Recognising Emotions from Keyboard Stroke Pattern, *International Journal of Computer Applications IJCA* 11(9): 24–28.

Killourhy, K. & Maxion, R. (2008). The effect of clock resolution on keystroke dynamics, *Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection*, Springer, pp. 331–350.

Killourhy, K. & Maxion, R. (2009). Comparing anomaly-detection algorithms for keystroke dynamics, *IEEE/IFIP International Conference on Dependable Systems & Networks, 2009. DSN'09*, pp. 125–134.

Killourhy, K. & Maxion, R. (2010). Keystroke biometrics with number-pad input, *IEEE/IFIP International Conference on Dependable Systems & Networks, 2010. DSN'10*.

Kohonen, T. (1995). Self-organising maps, *Springer Series in Information Sciences* 30.

Lopatka, M. & Peetz, M. (2009). Vibration sensitive keystroke analysis, *Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning*, pp. 75–80.

Marsters, J.-D. (2009). *Keystroke Dynamics as a Biometric*, PhD thesis, University of Southampton.

Martinez-Diaz, M., Fierrez-Aguilar, J., Alonso-Fernandez, F., Ortega-Garcia, J. & Siguenza, J. (2006). Hill-climbing and brute force attacks on biometric systems: a case study in match-on-card fingerprint verification, *Proceedings of the IEEE of International Carnahan Conference on Security Technology (ICCST)*.

Modi, S. K. & Elliott, S. J. (2006). Kesytroke dynamics verification using spontaneously generated password, *IEEE International Carnahan Conferences Security Technology*.

Monrose, F., Reiter, M. & Wetzel, S. (2002). Password hardening based on keystroke dynamics, *International Journal of Information Security* 1(2): 69–83.

Monrose, F. & Rubin (1997). Authentication via keystroke dynamics, *Proceedings of the 4th ACM conference on Computer and communications security*, ACM Press New York, NY, USA, pp. 48–56.

Monrose, F. & Rubin, A. (2000). Keystroke dynamics as a biometric for authentication, *Future Generation Computer Syststems* 16(4): 351–359.

Montalvao Filho, J. & Freire, E. (2006). Multimodal biometric fusion–joint typist (keystroke) and speaker verification, *Telecommunications Symposium, 2006 International*, pp. 609–614.

Nguyen, T., Le, T. & Le, B. (2010). Keystroke dynamics extraction by independent component analysis and bio-matrix for user authentication, *in* B.-T. Zhang & M. Orgun (eds), *PRICAI 2010: Trends in Artificial Intelligence*, Vol. 6230 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 477–486.

Obaidat, M. & Sadoun, B. (1997). Verification of computer users using keystroke dynamics, *Systems, Man and Cybernetics, Part B, IEEE Transactions on* 27(2): 261–269.

Pavaday., N., ., S. S. & Nugessur, S. (2010). Investigating & improving the reliability and repeatability of keystroke dynamics timers, *International Journal of Network Security & Its Applications (IJNSA)*, 2(3): 70–85.

Pohoa, V. v., Pohoa, S., Ray, A. & Joshi, S. S. (2009). Hidden markov model (hmm)-based user authentication using keystroke dynamics, patent.

Rao, B. (2005). *Continuous keystroke biometric system*, Master's thesis, University of California.

Ratha, N. K., Connell, J. H. & Bolle, R. M. (2001). An analysis of minutiae matching strength, *Audio- and Video-Based Biometric Person Authentication*.

Revett, K. (2008). *Behavioral biometrics: a remote access approach*, Wiley Publishing.

Revett, K. (2009). A bioinformatics based approach to user authentication via keystroke dynamics, *International Journal of Control, Automation and Systems* 7(1): 7–15.

Revett, K., de Magalhães, S. & Santos, H. (2006). Enhancing login security through the use of keystroke input dynamics, *Lecture notes in computer science* 3832.

Revett, K., de Magalhaes, S. & Santos, H. (2007). On the use of rough sets for user authentication via keystroke dynamics, *Lecture notes in computer science* 4874: 145.

Rodrigues, R., Yared, G., do NCosta, C., Yabu-Uti, J., Violaro, F. & Ling, L. (2006). Biometric access control through numerical keyboards based on keystroke dynamics, *Lecture notes in computer science* 3832: 640.

Rogers, S. J. & Brown, M. (1996). Method and apparatus for verification of a computer user's identification, based on keystroke characteristics. US Patent 5,557,686.

Ross, A. & Jain, A. (2004). Biometric sensor interoperability: A case study in fingerprints, *Proc. of International ECCV Workshop on Biometric Authentication (BioAW)*, Springer, pp. 134–145.

Ross, A., Nandakumar, K. & Jain, A. (2006). *Handbook of Multibiometrics*, Springer.

Sang, Y., Shen, H. & Fan, P. (2004). Novel impostors detection in keystroke dynamics by support vector machine, *Proc. of the 5th international conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2004)*.

Schneier, B. (1999). Inside risks: the uses and abuses of biometrics, *Commun. ACM* .

Song, D., Venable, P. & Perrig, A. (1997). User recognition by keystroke latency pattern analysis, *Retrieved on* 19.

Spillane, R. (1975). Keyboard apparatus for personal identification.

Stefan, D. & Yao, D. (2008). Keystroke dynamics authentication and human-behavior driven bot detection, *Technical report*, Technical report, Rutgers University.

Teh, P., Teoh, A., Ong, T. & Neo, H. (2007). Statistical fusion approach on keystroke dynamics, *Proceedings of the 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System-Volume 00*, IEEE Computer Society, pp. 918–923.

Theofanos, M., Stanton, B. & Wolfson, C. A. (2008). Usability & biometrics: Ensuring successful biometric systems, *Technical report*, The National Institute of Standards and Technology (NIST).

Umphress, D. & Williams, G. (1985). Identity verification through keyboard characteristics, *Internat. J. ManâĂŞMachine Studies* 23: 263–273.

Yu, E. & Cho, S. (2004). Keystroke dynamics identity verification – its problems and practical solutions, *Computers & Security* 23(5): 428–440.