



HAL
open science

Unconstrained Keystroke Dynamics Authentication with Shared Secret

Romain Giot, Mohamad El-Abed, Baptiste Hemery, Christophe Rosenberger

► **To cite this version:**

Romain Giot, Mohamad El-Abed, Baptiste Hemery, Christophe Rosenberger. Unconstrained Keystroke Dynamics Authentication with Shared Secret. *Computers & Security*, 2011, 30 (6-7), pp.427-445. 10.1016/j.cose.2011.03.004 . hal-00628554

HAL Id: hal-00628554

<https://hal.science/hal-00628554>

Submitted on 3 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Unconstrained Keystroke Dynamics Authentication with Shared Secret

Romain Giot*, Mohamad El-Abed, Baptiste Hemery, Christophe Rosenberger

*GREYC Laboratory
ENSICAEN - University of Caen - CNRS
6 Boulevard Maréchal Juin 14000 Caen Cedex - France*

Abstract

Among all the existing biometric modalities, authentication systems based on keystroke dynamics present interesting advantages. These solutions are well accepted by users and cheap as no additional sensor is required for authenticating the user before accessing to an application. In the last thirty years, many researchers have proposed, different algorithms aimed at increasing the performance of this approach. Their main drawback lies on the large number of data required for the enrollment step. As a consequence, the verification system is barely usable, because the enrollment is too restrictive. In this work, we propose a new method based on the Support Vector Machine (SVM) learning satisfying industrial conditions (*i.e.*, few samples per user are needed during the enrollment phase to create its template). In this method, users are authenticated through the keystroke dynamics of a shared secret (chosen by the system administrator). We use the GREYC keystroke database that is composed of a large number of users (100) for validation purposes. We compared the proposed method with six methods from the literature (selected based on their ability to work with few enrollment samples). Experimental results show that, eventhough the computation time to build the template can be longer with our method (54 seconds

*Corresponding author

Email addresses: romain.giot@greyc.ensicaen.fr (Romain Giot),
mohamad.elabed@greyc.ensicaen.fr (Mohamad El-Abed),
baptiste.hemery@greyc.ensicaen.fr (Baptiste Hemery),
christophe.rosenberger@greyc.ensicaen.fr (Christophe Rosenberger)

against 3 seconds for most of the others), its performance outperforms the other methods in an industrial context (Equal Error Rate of 15.28% against 16.79% and 17.02% for the two best methods of the state-of-the-art, on our dataset and five samples to create the template, with a better computation time than the second best method).

Key words: Biometrics, Authentication, Keystroke dynamics, Support Vector Machine learning.

1. Introduction

Authentication systems allow entities to be recognized before using resources; these resources can be physical, like a building, or logical, like a computer application. Traditionally, individuals authenticate themselves on computers by using the classical couple of *username* and *password*. This scheme, which is based only on one factor: the knowledge of the username and the password, suffers from various security holes [1]. *Strong authentication* uses multiple authentication factors to improve security. In this case, individuals are authenticated with the help of at least two authentication methods using one or several different factors among: (1) something *we know*; (2) something *we have*; (3) something *we are*.

Biometric systems can take part in the strong authentication scheme by providing the factor what *we are* when used with one of the two other factors. We can provide strong authentication in the password authentication scheme (what *we know*) by combining it with *keystroke dynamics* [2], which is a behavioral biometric modality monitoring the way individuals type on the keyboard (what *we are*). Its main interest lies the fact that it is considered as *unobtrusive*, because users already use passwords for authentication on computers and keystroke timing captures do not affect the user's habit. Several types of keystroke dynamics systems exist in the literature and are generally based on *very long texts* [3], *passwords* [4] or *shared secrets* [5] although several studies used a shared secret without referring to this term. The biometric sample can be cap-

23 tured *statically* (*i.e.*, at login phase) or *continuously* (*i.e.*, during the computer
24 session). In this study, we focus on static authentication with shared secrets.
25 Using a shared secret means that all users use the same password. The system
26 always acts as an authentication system, because only a certain group of people
27 is aware of this secret (what *we know*) while all the members of the group type
28 it differently (what *we are*). This kind of authentication is interesting and can
29 be used in different contexts: (i) several users use the same account, but it can
30 be useful to track which user is really using the account (in this case, we talk
31 about *identification* if the user does not specify his own username. This case
32 will not be treated in this paper), (ii) in analogy with password-protected build-
33 ings, an application requires the same password for all users and this password
34 is changed at regular intervals, etc.

35 During the verification, the system checks if the password is the required one,
36 if it differs from what is expected, the user is rejected, otherwise, the system
37 checks if the keystroke dynamics match. If the keystroke dynamics correspond
38 to the claimant's, the user is accepted, otherwise he is rejected. We argue on
39 the fact that most of the results presented in studies in the literature cannot be
40 compared easily due to various reasons which will be presented in this paper. In
41 order to help solve this problem, we propose a dataset whose aim is to be used
42 as a reference database in further keystroke dynamics studies. We also propose
43 a new method based on Support Vector Machine (SVM) [6] for unconstrained
44 shared secret keystroke dynamics.

45 The paper is organized as follows: this first section has presented the ob-
46 jective of this work. In the second section, we present the state-of-the-art of
47 keystroke dynamics. In the third section, we detail the proposed method. In
48 the fourth section, we present an experimental study for the validation of the
49 proposed method. These results are discussed in the fifth section. The sixth
50 section discusses the results. We conclude and present some perspectives in the
51 last section.

52 2. Background

53 In this section, biometric systems are first presented. An overview of their
54 evaluation aspects is then provided. Finally various discussions on the differ-
55 ences of keystroke dynamics studies are presented.

56 2.1. General biometric systems

57 2.1.1. Presentation

58 The aim of biometric systems is to verify the identity of an entity which
59 access to a resource. In the case of *physical access*, this resource can be a
60 building or a room, whereas in the case of *logical access*, this resource can be an
61 application on a computer.

62 Different biometric modalities can be classified among three main families
63 (even though we can find slightly different characteristics in the literature like
64 the biological one that is often forgotten):

- 65 • *Biological*: recognition based on the analysis of biological data linked to
66 an individual (e.g., DNA, EEG analysis, ...).
- 67 • *Behavioral*: based on the analysis of an individual behavior while perform-
68 ing a specific task (e.g., keystroke dynamics, signature dynamics, gait, ...).
- 69 • *Morphological*: based on the recognition of different physical patterns,
70 which are, in general, permanent and unique (e.g., fingerprint, face recog-
71 nition, ...).

72 In this work, we are interested in a behavioral biometric modality: the *key-*
73 *stroke dynamics* for managing *logical access* (*i.e.*, access to a computer applica-
74 tion).

75 Biometric authentication systems are generally composed of two main mod-
76 ules: (a) the *enrollment module* which consists in creating a template (or refer-
77 ence) for the user with the help of one or several biometric captures (or samples),
78 and (b) the *verification module* which consists in verifying if the provided sample

79 belongs to the claimed user by comparing it with its template. After verifica-
80 tion, a decision is taken to decide to accept or to reject the user depending on
81 the result of the comparison. We can also use an optional (c) adaptive module
82 which updates the template of a user after a successful authentication in order
83 to reduce the intra-class variability (the biometric data are not stable which
84 implies that different captures of the same user may be quite different).

85 2.1.2. Evaluation Methodologies

86 Many works have already been done on the evaluation of biometric sys-
87 tems [7, 8, 9]. This evaluation may be realized within three different aspects:

- 88 • *performance*: the objective is to measure various statistical criteria on
89 the performance of the system (*Capacity* [10], *Equal Error Rate (EER)*,
90 *Failure To Enroll (FTE)*, *Failure To Acquire (FTA)*, *computation time*,
91 *Receiver Operating Characteristic (ROC) curves*, *False Acceptance Rate*
92 *(FAR)*, *False Rejection Rate (FRR)* etc [8]);
- 93 • *acceptability and user satisfaction*: this gives some information on the
94 individuals' *perception*, *opinions* and *acceptance* with regard to the sys-
95 tem [7, 11];
- 96 • *security*: this quantifies how well a biometric system (algorithms and de-
97 vices) can resist several types of logical and physical attacks such as Denial
98 of Service (DoS) attack or spoofing or mimicking attacks [12].

99 In this work, we are mainly interested in performance evaluation, as our work
100 deals with authentication algorithms and not a whole system and its working
101 environment. The used metrics are the following ones:

102 **FAR** *False Acceptance Rate* which represents the ratio of impostors accepted
103 by the system;

104 **FRR** *False Rejection Rate* which represents the ratio of genuine users rejected
105 by the system;

106 **EER** *Equal Error Rate* which is the error rate of the system when it is con-
 107 figured in order to obtain a *FAR* value equal to the *FRR* one. We used
 108 this error rate as a measurement of performance to compare the proposed
 109 method with six existing methods from the state-of-the-art.

110 We believe that the three aspects (performance security, acceptability and
 111 user satisfaction) should be taken into account simultaneously when comparing
 112 different biometric systems: we cannot say that a system is good if it provides
 113 very low error rates (*i.e.*, very good performance) but has a very low user
 114 acceptance (*i.e.*, a high probability to be refused by users) [13].

115 2.2. Keystroke dynamics principles

116 In this section, we present the general principles of keystroke dynamics. The
 117 aim of keystroke dynamics systems (when used in a static authentication model)
 118 is to provide more security for password-based authentication systems which
 119 suffer of many drawbacks [14]: (i) passwords can be shared between users, (ii)
 120 passwords can be stolen (written on a piece of paper, from the database where
 121 it is stored, through network sniffing, ...), (iii) passwords can be guessed (social
 122 engineering [15]). Keystroke dynamics introduces an additional parameter to
 123 the password authentication process: something that qualifies the user or his
 124 behavior (*i.e.*, the way of typing passwords). Using this additional parameter
 125 strengthens the password authentication. The capture process is presented in
 126 Figure 1. It consists in capturing several features when the keys are pressed and
 127 released (timestamp of the event, code of the key, ...).

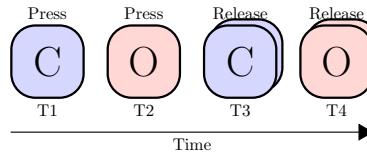


Figure 1: Information capture in a keystroke dynamics system when pressing C and O keys

128 The features extraction consists mainly in computing different latencies and
 129 duration times between each key. Figure 1 shows an a example where the user

130 presses two keys of the keyboard. The user presses "C" at T_1 , "O" at T_2 and
131 releases "C" at T_3 and "O" at T_4 . Note that the following relation is always
132 respected: $T_3 > T_1$ and $T_4 > T_2$ (we always release a key after pressing it),
133 while the following condition may not always be respected: $T_2 > T_3$ (because,
134 as in our example, a user may press another key before releasing the previous
135 one). We can extract three different types of latencies (T_2-T_1 , T_4-T_3 , T_2-T_3)
136 which we will call PP (latency between two pressures), RR (latency between
137 two releases), RP (latency between one release and one pressure) respectively
138 and one type of duration (T_3-T_1 or T_4-T_2) which we will call PR (duration
139 of a key press). In our example, T_2-T_3 is negative because the user presses O
140 before releasing C (this happens frequently when a user types fast). This is not
141 always true, but it is quite discriminating. The described process is repeated
142 for all the keys.

143 While it is possible to capture these four types of extracted features, the
144 selected features are not the same in all the studies. The PR and RP timings
145 seem to be the most used in the literature, but sometimes, authors only speak
146 about latency without defining which one is being used. In this paper, we use
147 the four types of timing values (even though they are linearly dependant).

148 2.3. State of the art

149 Keystroke dynamics were experimented for the first time in 1980 in a study
150 where seven secretaries were asked to type three different texts [16]. The results
151 were promising, but lacked a sufficient number of users involved in the database.

152 The first patent on keystroke dynamics was registered in 1986 [17]. Other
153 methods have been defined during the last twenty years, and, one of the latest
154 were proposed recently and uses *Hidden Markov Models* [18].

155

156 In 1990, Bleha *et al.* [19] proposed an authentication method based on key-
157 stroke dynamics of the *user name* combined with a *static phrase*. They used
158 a *Bayesian classification* and *distance measures*. The authors argued that the
159 longer the password is, the less the error rate becomes; the error rate decreases

160 when the number of enrolled patterns increases and results are better when us-
161 ing the person’s name instead of a password (due to their habit of typing it).
162 Studies using neural networks have appeared since 1993 [20]. Brown and Rogers
163 showed the possibility to use neural networks in static keystroke dynamics veri-
164 fication¹. A template is created for each user by using approximately 30 user
165 samples and 45 impostors samples where the samples are the timing information
166 extracted from the typing of the name of the user. Obaidat and Sadoun [5] used
167 latency and duration times of digraphs as features. They obtained an error rate
168 of 0% but used a large number of enrolled patterns (112). Monroe and Rubin
169 worked on keystroke dynamics for free texts [21]. They also used statistical
170 methods, and proposed to split users in different groups in order to speed up
171 the computation time (this is one of the first appearance of soft biometrics).

172

173 Cho and Hwang [22] allowed individuals to use *pause* helped by *cues* (which
174 act as metronomes) to improve *unicity*, *consistency* and *discriminability* of their
175 password and make the forgery of typing dynamics more difficult. Rodrigues *et*
176 *al.* [23] used *Hidden Markov Model* in their authentication method. By using
177 passwords only composed of numbers, they obtained an *EER* of 3.6%. This
178 study was interesting since it demonstrated the use of keystroke dynamics for
179 pin code authentication based environment (*i.e.*, ATM or cell phone). Sang *et*
180 *al.* [24] tested the efficiency of SVM for keystroke dynamics verification. They
181 used a one-class and a two-class SVM (in this case, simulated impostors’ data
182 are generated). The performance tradeoff and time computation were better
183 and faster than with neural networks, but the experiment was done with only
184 10 individuals, a number too few to be representative (in our study, we used
185 SVM in a different way by using pre-processing (the discretization) and post-
186 processing (the score computing) and validating on a much bigger database).
187 SVM has also been used in a one-class way [25]. This work uses SVM as a
188 novelty detector to detect impostor’s pattern (a novel pattern). The presented

¹even if they use the word *identification* in this paper.

189 framework includes feature selection through genetic algorithm which greatly
190 improves the recognition rate, but the number of user pattern necessary to cre-
191 ate the template is 50.

192

193 In 2007, Hocquet *et al.* [4] automatically classied the individuals in differ-
194 ent classes depending on various parameters and assigned different thresholds
195 configuration for each class. They obtained an *EER* of about 2%. The classes
196 definition and thresholds configuration are realized using a validation database.
197 Another study [26] used various digraph information and time of typing for
198 both username and password and discretized them into an alphabet of twenty
199 discrete elements. The classification was done by using the rough set paradigm.
200 They obtained an *EER* value lower than 1%. Gaussian mixture modeling of key-
201 stroke patterns was used in [27]. Hosseinzadeh and Krishnan also gave valuable
202 informations on how to create good keystroke dynamics databases, and how to
203 present the results. The obtained *EER* was around 4.5%. They argued that
204 if passwords have more than 8 letters, the number of typing mistakes increases
205 (also interpreted in the *Failure to Acquire Rate*) even though the number of
206 recognition errors decreases.

207 Some studies [28, 29, 30] took into account the typing evolution of the user
208 in order to adapt his template after each authentication. The aim of these ap-
209 proaches is to improve the system performance: as being a behavioral modality,
210 keystroke patterns are subject to evolve through the life time of the keystroke
211 dynamics authentication system. If it does not take into account this variabil-
212 ity, we can get a high number of false rejects. It seems that in the majority of
213 papers, the update of the biometric template is realized only with captures from
214 the genuine users (whereas in reality, if an impostor succeeded in authenticate
215 himself on the system, his fake pattern would be added to the template). For
216 more information, readers can access a recent review on keystroke dynamics
217 available in [31].

218 *2.4. Discussion*

219 In this section, we point out why it is really difficult to compare keystroke
220 dynamics methods presented in the state-of-the-art.

221 *2.4.1. Differences in Acquisition Protocols*

222 Most of the studies in the literature use different protocols for their data
223 acquisition [32, 33]. This is totally understandable due to the existence of dif-
224 ferent kinds of keystroke dynamics systems (static, continuous, dynamic) that
225 require different acquisition protocols. It is known that the performance of each
226 algorithm can vary depending on the used database [27]. In the keystroke dy-
227 namics research field, various protocols are used to collect the data. They differ
228 on the *number of individuals* taking part in the study, the *acknowledgement* of
229 the password (the user chooses the password, or the password is an imposed
230 one). This impacts the typing speed and affects the *FTA* measure. They dif-
231 fer on the use of *different computers* (which can impact the timing accuracy
232 depending on the operating system), *different keyboards* (which may impact on
233 the way of typing), the *quantity* of collected data, the *duration* of the collection
234 of the whole database, the *control* of the acquisition process (*i.e.*, acquisition
235 done without knowledge of the researcher who can verify if it is done with re-
236 spect to the protocol or made at home where no verification is possible), the use
237 of *different* or *identical* passwords (which impacts on the quality of impostors'
238 data). Table 1 illustrates the differences in the protocol used in existing studies
239 in keystroke dynamics.

240 *2.4.2. Differences on the Objective Analysis*

241 Many performance metrics can be used to qualify a biometric system. Nev-
242 ertheless, two important issues should be considered carefully during the com-
243 parison of authentication algorithms:

- 244 1. the benchmark database used, most of time, is private, and
- 245 2. the number of samples required during the enrollment phase.

Table 1: Summary of the protocols used in 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.

Paper	A	B	C	D	E	<i>FAR</i>	<i>FRR</i>
[5]	8 weeks	15	112	no	no	0%	0%
[19]	8 weeks	36	30	yes	yes	2.8%	8.1%
[23]	4 sessions	20	30	??	no	3.6%	3.6%
[4]	??	38	??	??	no	1.7%	2.1%
[26]	14 days	30	10	??	no	0.15%	0.2%
[27]	??	41	30	no	no	4.3%	4.8%
[21]	7 weeks	42	??	no	no	??	20%
[34]	4 weeks	8	12	??	??	5.58%	5.58%
[33]	8 sessions	51	200	yes	no	9.6%	9.6%

246 More generally speaking, it is impossible to compare a study using twenty
 247 vectors for the enrollment process with another using only five vectors (obvi-
 248 ously, the more samples used during the enrollment phase, the better the created
 249 templates). In addition to the enrollment size, the degree of expertise of the
 250 volunteers has an impact on the illustrated performance results [35]. The same
 251 argument also holds when comparing research works using a global threshold,
 252 with those using per-user threshold. Table 1 presents the number of vectors
 253 used for creating the enrolled template and the use of a global or individual
 254 threshold for some protocols in the literature.

255 2.4.3. Laboratory Environment

256 The problem of the laboratory environment is inherent for most of keystroke
 257 dynamics studies. For this reason, most of the passwords are *artificial* ones
 258 generated differently in each study (*i.e.*, dictionary words, random password:
 259 combination of letters, numbers and symbols, etc.) and the individuals are not
 260 at ease when typing these passwords (because they do not use them daily, and
 261 they do not choose the password). In some controlled environments, individuals
 262 are in a quiet room without any interference. This does not reflect the reality

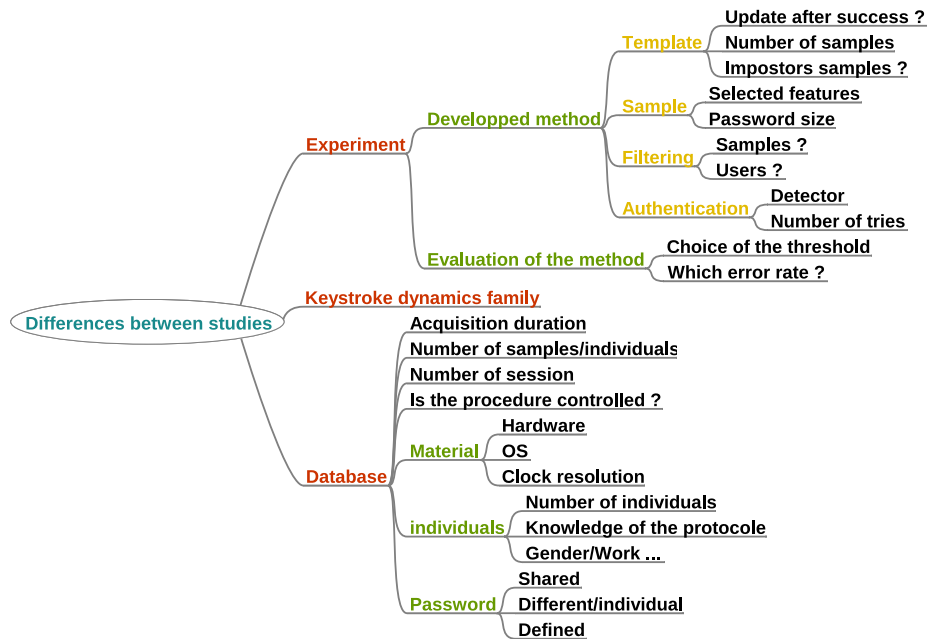


Figure 2: Summary of the differences which can be observed in keystroke dynamics studies

263 where we can authenticate on our machines while talking with other people or
 264 in a noisy environment. In an uncontrolled environment, nothing guarantees
 265 that all the typing patterns of a user have been done by the same user with
 266 respect to the protocol.

267 Figure 2 presents a mindmap of the differences between the keystroke dy-
 268 namics studies referenced in this paper. These differences were also presented
 269 in [33].

270 2.5. Conclusion

271 Most systems in the literature do not propose viable solutions for a daily
 272 use at work owing to the high quantity of captures required to create the tem-
 273 plate. The diversity of the protocols implies the difficulty to compare them.
 274 The comparison between all the keystroke dynamics studies is impossible due
 275 to the use of different protocols, and especially, the lack of a public database.
 276 Another problem concerns the “configuration” of the algorithms by using differ-

277 ent numbers of captures to create the template, or by using template adaptation
278 methods or not. Moreover, very few works use incremental learning which is
279 fundamental for behavioral biometric systems.

280 The aim of the following section is to present a solution to these problems.
281 We compared our algorithm with six others following a rigorous protocol with
282 a database [36] we created which contains more than 100 users and is acquired
283 from 5 sessions separated each by, at least, one week.

284 **3. Proposed Method**

285 The goal of the developed method is to limit the number of captures required
286 during the enrollment step (for obvious usability reasons) while maintaining
287 good performance. Its originality is due to:

- 288 • the use of discretization as pre-processing,
- 289 • the computation of a decision score from the response of the SVM (in
290 order to correct some errors of the SVM classification),
- 291 • the use of different supervised incremental learning schemes to update the
292 biometric template of an individual after each genuine verification.

293 The template structure is explained in Section 4.2.2. We present some details
294 on the above points in the following subsections. Figure 3 summarizes the global
295 process.

296 *3.1. Enrollment*

297 Users are asked to type the passphrase set by the administrator five times.
298 The feature vector is discretized in an alphabet of five values with equal size
299 bins (we did not try other scheme of discretization) during the preprocessing
300 steps. The bin computation method is presented in the next section. Then,
301 a support vector machine is used for the learning step (see Figure 4). The
302 template contains two informations: the information on the bins (in order to be
303 able to correctly discretize test patterns) and the trained SVM.

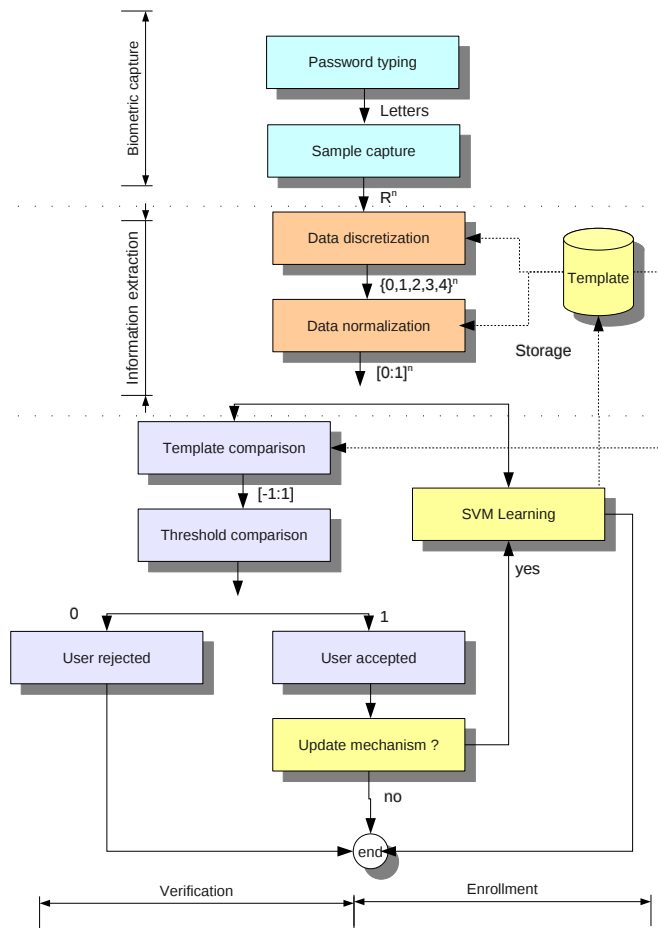


Figure 3: Global view of the system.

304 For the enrollment, the machine learning method is a two-class SVM. Sup-
 305 posing that we have a training set $\{\mathbf{x}_i, \mathbf{y}_i\}$ where \mathbf{x}_i is an enrolled vector and
 306 \mathbf{y}_i the class of the associated individual (genuine/impostor). For problems with
 307 two classes, with the classes $y_i \in \{-1, 1\}$, a support vector machine [6, 37] im-
 308 plements the following algorithm. First, the training points $\{\mathbf{x}_i\}$ are projected
 309 into a space \mathcal{H} (of possibly infinite dimension) by means of a function $\Phi(\cdot)$.
 310 The second step is to find an optimal decision hyperplane in this space. The
 311 criterion for optimality will be defined shortly. Note that for the same training
 312 set, different transformations $\Phi(\cdot)$ may lead to different decision functions.

A transformation is achieved in an implicit manner using a kernel $K(\cdot, \cdot)$ and consequently the decision function can be defined as:

$$f(\mathbf{x}) = \langle w, \Phi(\mathbf{x}) \rangle + b = \sum_{i=1}^{\ell} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b \quad (1)$$

313 with $\alpha_i^* \in \mathbb{R}$. The values w and b are the parameters defining the linear decision
 314 hyperplane. In the proposed system, we use a linear function as the kernel
 315 function.

In SVMs, the optimality criterion to maximize is the margin, that is to say, the distance between the hyperplane and the nearest point $\Phi(\mathbf{x}_i)$ of the training set. The α_i^* which optimize this criterion are obtained by solving the following problem:

$$\left\{ \begin{array}{l} \max_{\alpha_i} \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i K(\mathbf{x}_i, \mathbf{x}_j y_j) \\ \text{with constraints,} \\ 0 \leq \alpha_i \leq C, \\ \sum_{i=1}^{\ell} \alpha_i y_i = 0. \end{array} \right. \quad (2)$$

316 where C is a penalization coefficient for data points located in or beyond the
 317 margin and provides a compromise between their numbers and the width of
 318 the margin. The biometric reference of one user is given by the α_i^* , $i = 1 : \ell$
 319 coefficients.

In addition to obtaining the guessed label, it is possible to calculate an

estimate of its probability:

$$p(y = i|x) \approx P_{A,B}(f) = \frac{1}{1 + e^{A\hat{f}+B}} \quad (3)$$

320 where A and B are estimated by minimizing the negative log-likelihood function
321 using known training data and their decision values \hat{f} .

322 As we are using impostors patterns (patterns from other users of the system,
323 which do not mimic the genuine user behavior) during the enrollment step, the
324 definition of a biometric reference requires the use of all existing references in
325 the database.

326 When the data of all users (m is the number of users) are taken into account
327 (this is the scenario we have chosen in the experiment), there are $5 * m$ training
328 vectors (5 belonging to the user and $5 * (m - 1)$ belonging to the impostors). If
329 a new user is added to the system later, different scenarios can be applied:

- 330 • We compute the template of all users. Thus, there are $m + 1$ templates to
331 compute using $5 * (m + 1)$ samples each. This method can be very long if
332 there are many users.

333 Moreover, the performance of the method for a user might not increase
334 by adding the 5 training vectors of the new user as impostor data. These
335 new impostor data could be insignificant regarding to the existing $5*(m-1)$
336 impostor training vectors. Thus, the ratio between the time consumption
337 and the performance's evolution may not lead to a good trade-off.

- 338 • We compute the template of the new user. This is more efficient because
339 only one template has to be generated.

340 We have not explored which of these scenarios is the best, because we have not
341 tested inclusion of users during the life of the system.

342 3.2. Verification

The verification step consists in realizing a recognition procedure with the SVM algorithm for a given biometric capture. We define a score and we use a

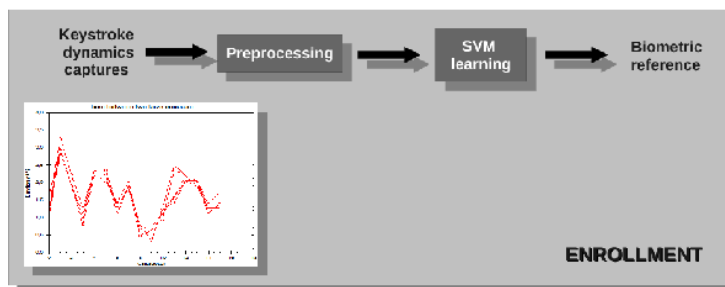


Figure 4: Enrollment scheme

threshold to decide if the user is the genuine one or an impostor. We propose different solutions to set this threshold. If the verification is successful, we use this new capture to update the biometric reference of the user in order to take into account the evolutions of keystroke dynamics (see Figure 5). The test

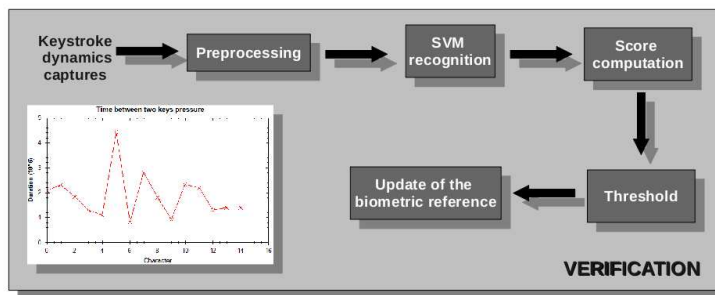


Figure 5: Verification scheme

patterns are discretized according to the information available in the template. We then classify the test pattern using the trained SVM and also estimate its probability. It is then necessary to compute a score in order to obtain a ROC curve with several points allowing a better configuration of the system. The score value for the verification test is computed as follows:

$$Score = -prb * prd \quad (4)$$

343 where prb stands for the probability accorded to the SVM result and prd corre-
 344 sponds to the class of the result which is -1 for an impostor and 1 for a genuine

345 user.

346

347 The decision threshold can be set by following two different approaches:

- 348 • by using the same threshold for all the users
- 349 • by using a user specific threshold for each user.

350 The statical performance of biometric systems is different given this setting [27,
351 38]. It is not the aim of this work to present different ways of selecting these
352 thresholds. Their configuration depend on the targeted security level of the
353 system and could be defined empirically or automatically (by computing it based
354 on the enrolled samples). Both approaches are compared in Section 4.3.5.

355 3.3. Updates of the biometric reference

356 As keystroke dynamics is a behavioral modality, it is useful to update the
357 biometric reference when the user authenticates himself on the system. Among
358 the different algorithms proposed in the literature, the following four methods
359 were implemented:

- 360 • *adaptive*: a method replacing the oldest enrolled sample by the new
361 one [29, 30]. This method is called “sliding window” in [28];
- 362 • *progressive*: a method adding the new sample in the list of enrolled vectors.
363 This method is called “growing window” in [28];
- 364 • *average*: while the number of required enrolled vectors is not reached (set
365 at 15), the progressive method is used, whereas when the total number is
366 reach, the adaptive one is used. Samples are added only when they are not
367 far too different from the enrolled samples (by comparing the difference
368 between the test vector and the mean considering the standard deviation).
369 This method is almost similar to [39];
- 370 • *correlation*: in this mode, the new sample is added to the database only
371 if it is well-correlated with the enrolled samples. To test this correlation,

372 we use the absolute value of the Pearson correlation factor between the
373 test vector and the average enrolled vectors. If the score is higher than
374 0.5, we add the vector to the template in the same way as the “average”
375 procedure”.

376 4. Validation

377 In this section, attempts are made at answering several questions about
378 keystroke dynamics: Which are the parameters value of the verification method
379 ? What is the performance of our method compared to the ones in the literature
380 ? Is there any keyboard dependency ? What is the impact of the number of
381 captures during the enrollment on the performance ? What is the best template
382 update strategy ? Is there any computation timing difference between each
383 method ?

384 4.1. Authentication Method Configuration

385 In this section, we present the process involved during the development of
386 our method. A development benchmark (a private database) was used while
387 creating the method. This is the same as the benchmark used in [40] which was
388 created with the same software. Sixteen users provided fifteen samples in three
389 sessions, each of each was separated by a one week period. Each session consists
390 of five captures of the password “laboratoire greyc”.

391 4.1.1. Choice of the kernel

392 When using SVM, it is necessary to choose the appropriate kernel. For
393 this experiment we chose to compute the feature V (presented in the following
394 paragraph) and used a multi-classe SVM (each user has his own label). Five
395 samples per user are used to create his template, while the other samples are
396 used for the test. The samples were chosen randomly and the experiment was
397 launched 10 times (averaged results are presented). The SVM error rate when
398 using different kernels is presented in Figure 6. We operated a grid search
399 and selected the parameters giving the best results in order to reduce error

400 rate. Having obtained these results, we chose the linear kernel as it works well
 401 and does not require a lot of parameters. This can be explained by the high
 402 dimension of our patterns and because the data is almost linearly separable [41].
 403 In the implemented method, we use the default parameters of the *libsvm* (*i.e.*,
 404 $C = 1$). The method could be improved by selecting the best C parameter using
 405 the data in the enrollment step.

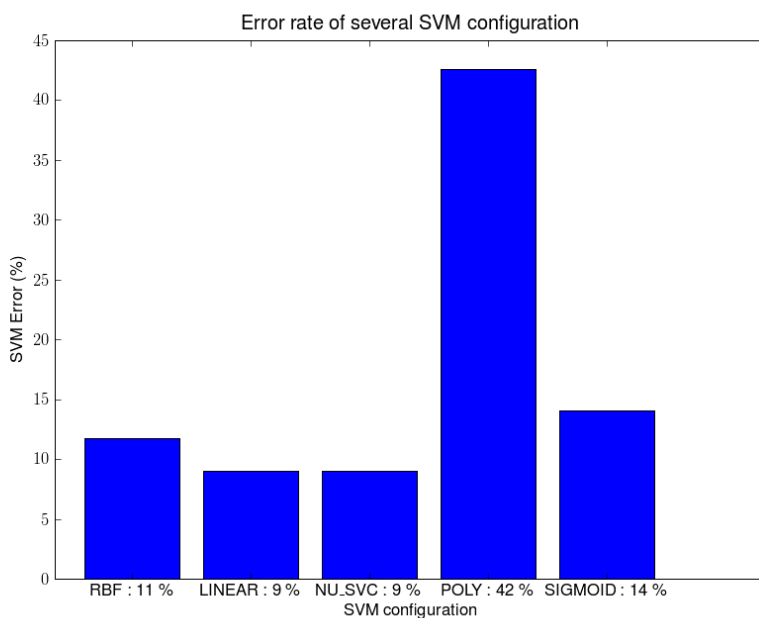


Figure 6: Performance of SVM using different kernels

406 After having chosen the kernel, several additional experiments allowed to
 407 conclude that using a two-class SVM provides better results.

408 4.1.2. Choice of the extracted features

409 It was previously seen that different kinds of extracted features can be used.
 410 Different configurations of extracted features were tested in order to choose the
 411 best one:

- 412 • RR or RP or PR or PP times only;

- 413 • V which is the concatenation of the four previous timing vectors. It is a
414 feature fusion commonly used in keystroke dynamics by using the duration
415 and one type of latency;
- 416 • $ex1$ which is the timing vector V with the total typing time in addition;
- 417 • VN which is vector V divided by the total typing time;

418 Table 2 presents the EER value obtained from the proposed method (without
419 using any discretization) depending on the extracted features used. It can be
420 seen that the extracted vector V gives the best results. This is why it will
421 be used throughout the experiments. These results are better than those in
422 Figure 6 because a two class SVM is used instead of a multi-classe one.

Table 2: Performance of the proposed system depending on the extracted features

Input	RR	RP	PR	PP	V	ex1	VN
EER	07.36%	08.95%	09.00%	12.81%	03.81%	04.36%	04.31%

423 4.1.3. Numbers of bins for the discretization

424 As mentioned before, the proposed method uses a discretization process.
425 Therefore, a parameter of this method is the number of bins to use. We present
426 here the methodology adopted in order to set this number. We computed the
427 EER value of the method with various numbers of bins for the discretization.
428 Supposing we have an n -dimension template and p samples per template (*i.e.*, p
429 enrolled captures). For each dimension, we detect the maximum and minimum
430 value through the p templates, which gives us n different ranges. Each of these
431 ranges is split into i bins of equal width (except of the boundaries where they
432 are of infinite size) (*i.e.*, $(max - min)/i$). To discretize a template, we replace
433 each value by the number of the bin containing it. For example, if the minimum
434 and maximum value of the selected dimension² are 0 and 99 respectively and
435 we decide to use 3 bins. The width of the bin is $(99 - 0)/3 = 33$, the first bin

²The process is repeated for each dimension

436 embeds the range $[-\infty; 33]$, the second one the range $[33; 66]$ and the last one
 437 $[66; +\infty[$. The values 40 and 120 are thus replaced by 1 and 2. Table 3 presents
 438 an example of range computation with a four-dimension pattern.

Table 3: Bin computation

	Dim 1	Dim 2	Dim 3	Dim 4
Sample 1	100	5	200	-8
Sample 2	110	7	300	-7
Sample 3	140	-1	250	-10
Sample 4	80	3	320	2
Min	80	-1	200	-10
Max	140	7	320	2
Width	12	1,6	24	2,4

439 Table 4 presents the *EER* values of different discretization methods and the
 440 differences without using such procedure. By using 5 bins, as in [38], the best
 441 results are obtained. Depending on the database, five bins could not be the best
 442 choice, but it is estimated that a number of bins between five and ten can be
 443 chosen without any problem.

Table 4: EER for different bins size during the discretization

Nb bins	2	3	4	5	6	7	8	9	10	20
EER	49.77%	40.05%	10.04%	2.77%	3.86%	3.59%	3.64%	4.55%	3.64%	3.64%
Difference	-45.96%	-36.24%	-6.23%	1.04%	-0.05%	0.22%	0.17%	-0.74%	0.17%	0.17%

444 After having configured the parameters of the authentication method with
 445 a development benchmark, it is necessary to validate the method with another
 446 one.

447 4.2. Experimental protocol

448 In this section, we define the biometric database used for testing the proposed
 449 method. Six methods were selected from the literature. Their results would be

450 compared with the proposed method. The *EER* value is used as an objective
451 information on the performance.

452 4.2.1. Definition of a validation database

453 Different databases of different qualities have been used in the literature, but
454 they are rarely shared with the community (although another huge database has
455 been constructed at the same time as ours [33]). It is known that the results can
456 be highly dependent on the used database. The main benefit of using a common
457 database is to help researchers avoid having to spend too much time creating
458 a database, and to easily compare the performance of different algorithms with
459 the same input data.

460
461 Hosseinzadeh and Krishnan [27] presented some very interesting informa-
462 tions on a possible method to create a good keystroke dynamics database sup-
463 posed to be used with specific confidence intervals. They applied their method
464 to create a database used in their work, but unfortunately did not make it avail-
465 able. In [13], we argue that to create a good behavioral biometric database, the
466 number of required sessions have to be higher than or equal to three ; these
467 sessions must be spaced in time, the population must be large and diversified.
468 These requirements were not always followed in previous works.

469
470 *GREYC-Keystroke* is a software allowing the creation of keystroke dynam-
471 ics databases. It is available for download at the following address: [http:](http://www.ecole.ensicaen.fr/~rosenber/keystroke.html)
472 [//www.ecole.ensicaen.fr/~rosenber/keystroke.html](http://www.ecole.ensicaen.fr/~rosenber/keystroke.html). A screenshot of the
473 application is shown in Figure 7. We developed this application in order to
474 create our own keystroke dynamics database, to share it with the biometric
475 community and to allow other researchers to create their own databases. The
476 data are stored in an sqlite file which allows quick and easy extraction of specific
477 information, thanks to SQL (Structured Query Language) queries.

478
479 We created a meaningful keystroke dynamics database with the help of the

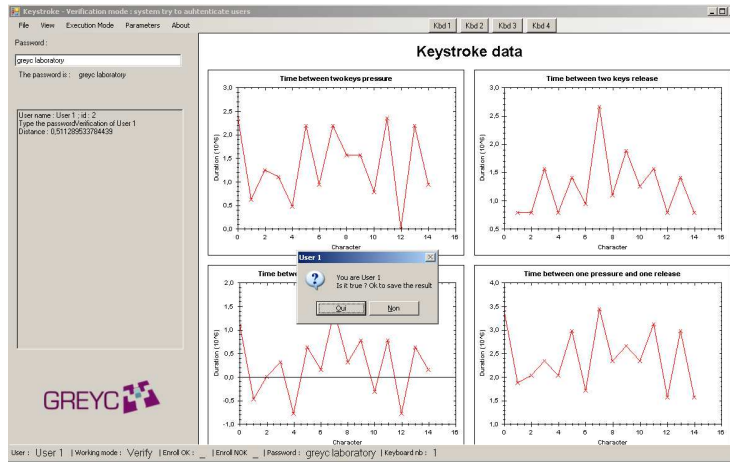


Figure 7: Screenshot of the database collecting tool

480 *GREYC-Keystroke* software by respecting various constraints presented in [13]
 481 as a guideline for creating a good behavioral biometric database (in terms of
 482 number of sessions, duration between each session, number of individuals, etc.)
 483 Most of the population in the database is composed of researchers in computer
 484 science, secretaries, students in computer science and chemistry. There are dif-
 485 ferent kinds of typists: fast, slow, two fingers, all fingers, etc., but we did not
 486 tracked this information.

487

488 A total of 133 individuals in the capture process by typing the passphrase
 489 “greyc laboratory” between 5 and 107 times, between 03/18/2009 and 07/05/2009.
 490 There are 7555 available captures, and the average number of acquisitions per
 491 user is 51, with 100 of them having more than 60 captures. Most of the in-
 492 dividuals participated in at least 5 sessions. We choose this password for two
 493 main reasons (i) this is the name of our laboratory, and using it could help
 494 the laboratory become better known, and (ii) it is a long enough password,
 495 with a good distribution of the keys on the keyboard which can help improve
 496 discriminability [34]. To type this passphrase on an *AZERTY* keyboard, users
 497 would likely need both hands to type with as the keys are positioned across

498 the keyboard. The position of the letters in the password on the keyboard are
 499 represented in Figure 8. We have not tested other passwords due to the amount
 of time required to create another database. The software is available freely in

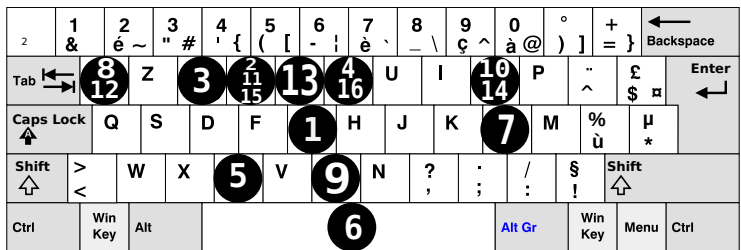


Figure 8: Position of the keys on a French AZERTY keyboard. Marked keys belong to the password. Numbers indicate the order of the character in the password. The original layout is taken from http://fr.wikipedia.org/wiki/Fichier:KB_France.svg.

500

501 the hope that fellow researchers will use it to create other databases in order to
 502 do other kind of experiments. More information about this software is available
 503 in [36].

504 In comparison with the databases presented in Table 1, ours is a rather large
 505 database, collected over a reasonably long period. The participants were asked
 506 to participate in one session every week (a few of them did two sessions within
 507 a week due to time constraints). Each session consists in typing the password
 508 correctly twelve times. Except for the first session during which the participants
 509 have the possibility to practice at typing the password over a short period. It
 510 came to our notice that very few of them actually participated in all the sessions
 511 by considering the number of available samples. Two keyboards (the original
 512 laptop’s keyboard, and a USB keyboard plugged onto the laptop) were used to
 513 verify if the template is only dependent on a user or if it is dependent on both
 514 the user and the keyboard used. That is why during each session, individuals
 515 were asked to type the password six times on each keyboard and to alternate
 516 the keyboard each time. As the participants have to change the keyboard after
 517 typing the password each time, there is a small break before typing the next
 518 password which can help avoid the problem of users typing mechanically too

519 similar patterns (without removing hands from the keyboard or a break between
520 each input, the intra-class variability is too weak and not representative enough
521 of a real keyboard usage where passwords are not typed so frequently in such
522 a short period of time). Figure 9 shows the two different keyboards used for
523 the experiment. It can be seen that their shapes are quite different. The key
524 pressure is also different and the presence of the cursor ball (in red) in the middle
525 of the laptop’s keyboards is disturbing for most users.



Figure 9: Differences of the two keyboards used during the experiment.

526 At the beginning of the first session, the participants were able to practice
527 at typing of the password on the two keyboards as long as they wanted (we did
528 not keep a track of the number of tries per user, but, most of them did it not
529 more than five times). This training is necessary because as it is an imposed
530 password, users are not used to typing it (especially when written in a foreign
531 language). This is a necessary step because intra-class variability would be too
532 significant without this test. So even if five samples are used to create the
533 template, some user may have provided up to ten samples (where only the last
534 five were saved and used). The participants were aware of the fact of being in
535 a training or capturing mode. In another context where it would be up to the
536 user to choose his own password, this training phase may not be so useful. The
537 training step was not allowed during the other sessions.

538 A summary of the subset of the used database for our study is presented
539 in Table 5. It belongs to the family of database with one unique password as
540 in [33, 42].

Table 5: Summary of the information provided in the subset of the database used in the experiment. The users providing answers to our questionnaire are not necessarily the ones who participated in this study.

Information	Description
Users	100 users
Database sample size	6000 passphrases (60 samples per user)
Data sample length	16 characters ('greyc laboratory')
Typing error	not allowed
Controlled acquisition	yes
Age range	between 19 and 56 (repartition presented in Table 6)
Gender	approximately 73% of males and 27% of females
PC usage frequency	unknown
User profession	students, researchers, secretaries, labourers (unknown repartition)
Keyboard	2 AZERTY keyboards (1 laptop, 1 USB)
Acquisition platform	Windows XP/Greyc keystroke software

541 *4.2.2. Biometric Sample*

As mentioned earlier, different kinds of information can be extracted from the keystroke dynamics captures. In this work, we decided to use all the different latencies and durations timings. In the rest of the paper, we call v the biometric sample. This sample is created with the help of one capture of the password. The vector v is built as followed:

$$v = V = \{RR_0, PP_0, RP_0, PR_0, RR_1, PP_1, RP_1, PR_1, \dots\} \quad (5)$$

542 where RR, PP, PR, RP stands for timing between two key releases, two key
543 presses, one press then one release (the duration of pressure of a key), one
544 release then one press respectively. The size of the feature vector depends on
545 the size of the password (this is not a problem because vectors are compared to
546 the template only if the right password has been typed). For a password of n
547 characters, v has a dimension of $3 * (n - 1) + n$.

548 *4.2.3. Selected methods for the comparative study*

549 In this section, we present the methods in the literature that were selected
 550 for the comparative study. We denote v as the test vector (extracted from the
 551 test sample) and i as the size of this vector (and of the other vectors embedded
 552 in the template). As this study is done only with one password, the generated
 553 scores were not normalized.

554 *4.2.4. Statistic-based Algorithm*

Three different statistical methods are tested. They differ in the content of
 their computed template and the complexity of their score computing method.
 In the first method, the template embeds μ which is the mean of the samples [19]:

$$STAT1 = \frac{(v - \mu)^t (v - \mu)}{\|v\| \cdot \|\mu\|} \quad (6)$$

555 For the second method, the template embeds both μ , the mean of the samples
 556 and σ , its standard deviation [4]:

$$STAT2 = 1 - \frac{1}{n} \sum_{i=1}^n e^{-\frac{|v_i - \mu_i|}{\sigma_i}} \quad (7)$$

The third method uses μ , the mean, σ , the standard deviation and m , the
 median [34] of the samples of the enrollment. We name this method *STAT3*.
 While the two previous methods could be represented by a simple equation,
 the third one is more complex because it requires several stages of calculations.
 First, we check if the test vector satisfies the condition specified in (8) which is
 vectorial (the test is done in all the dimensions of v).

$$boolres = \min(\mu, m) * (0.95 - \frac{\sigma}{\mu}) \leq v \leq \min(\mu, m) * (1.05 + \frac{\sigma}{\mu}) \quad (8)$$

557 The result of this comparison is a boolean array containing true when the crite-
 558 rion is verified for the required dimension of the vector, and false otherwise. In
 559 the second step, all the occurrences of false are replaced by a 0, each occurrence
 560 of true preceded by false is replaced by 1.5, while the other true values are re-
 561 placed by 1. We now have now an array of numbers. The third step consists in

562 summing all the elements of the array ; this sum is the score of this biometric
 563 method.

564 4.2.5. Distance Based Algorithm

We consider a simple metric based on an Euclidean distance [21]. In this method, the template is simply the list of enrolled samples. This distance is computed between the test vector and each of the enrolled samples. The score is then the minimum computed distance, as described in (9).

$$DIST = \min \left(\forall_{u \in enrol}, \sqrt{\sum_{i=1}^n (u_i - v_i)^2} \right) \quad (9)$$

565 4.2.6. Rhythm Based Algorithm

This method consists in discretizing keystroke values along five different classes and computing a classical Hamming distance [4]. The template embeds the bin definition (in order to discretize test sample in the same way as the enrolled mean sample) and μ the discretized version of the mean enrolled samples. The score computation is described in (10).

$$RHYTHM = \frac{1}{n} \sum_{i=1}^n abs(class(v_i) - class(\mu_i)) \quad (10)$$

566 where $class(i)$ is a function returning the class of i (*i.e.*, we operate a discretiza-
 567 tion of the time) along five different classes. To compute the classes, we divide
 568 the space in five clusters of the same size between the minimal and the maximal
 569 value of the learning database (Equation 11). The assigned classes of the whole
 570 dimension of each vector is the number of the cluster.

$$cluster_width = \frac{max(train_data) - min(train_data)}{5} \quad (11)$$

571 4.2.7. Neural Networks

572 Neural networks have been used in various keystroke dynamics studies [20,
 573 5, 43, 44]. They usually require a huge number of samples in order to create
 574 the template. Nevertheless, we chose to present it here because it seems to

575 be the closest method to our proposal (*i.e.*, use of impostors samples). In our
 576 experiment, we use a feed forward multi layer perceptron, with one hidden
 577 layer containing 45% of number of input nodes, and one output node giving a
 578 score. We empirically chose this number of hidden nodes in order to limit the
 579 computation time of the learning. The cost function is the sum of the squared
 580 difference. The constrained truncated Newton algorithm (TNC) is used as the
 581 learning method.

582 The learning data are arranged in the same way as our SVM-based method.
 583 No other neural network configuration has been tested. Thus, in this method,
 584 the template embeds the trained network which has been computed with clients'
 585 and impostors' enrolled samples (whereas the other methods from the literature
 586 only use clients' enrolled samples).

587 4.3. Experimental results

588 In this section, we present different experimental results on the database. In
 589 this part of the text, CONTRIB refers to our keystroke verification method.

590 4.3.1. Acquisition

591 100 volunteers were asked to fill in a questionnaire. Their age and gender
 are presented in the Table 6. In keystroke dynamics authentication, there is

Table 6: Diversity of the population in the database (in term of gender and age).

	Male	Female	Total
18-25	46	13	59
26-35	19	6	25
36-49	8	6	14
50+	0	2	2
Total	73	27	100

592 quite a large number of failures during acquisitions. These failures are due to
 593 the fact that no mistake is allowed while typing the password: a typing mistake
 594

595 obliges the user to type the password again from scratch. It would thus be use-
 596 ful to analyze the causes of these mistakes. Figure 10 presents the quantity of
 597 captures done by each user (sorted by amount of provided samples) by dividing
 598 the correct samples (in gray) by the erroneous samples (in black). The number
 599 of mistakes made is quite huge for most of the volunteers. The average mistake
 600 rate is about 20%: one input out of five is incorrect due to typing mistakes.

601

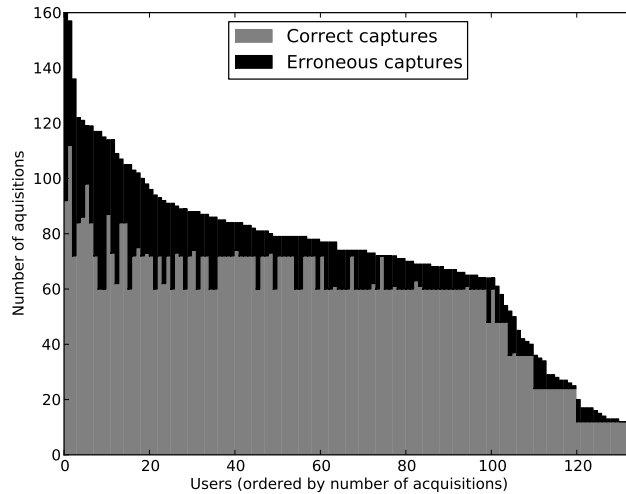


Figure 10: Number of acquisitions for each user. Correct and erroneous acquisitions are both represented

602 These mistakes are due to several reasons: (i) the password is quite long to
 603 type (16 characters) and it is known that typing mistakes increase if more than
 604 height characters are used [27], (ii) users may be not used to the keyboard and
 605 may hesitate a lot while typing (some participants do not often use computers
 606 and are not very familiar with keyboard usage, while others are perturbed by the
 607 use of a passphrase in English), (iii) users want to type faster than they are able
 608 to do, (iv) users forget the password (sessions were separated by one or more
 609 weeks and some users also participated in the creation of other benchmarks with
 610 different passwords), (v) users are disturbed by the environment (*e.g.*, discussion

611 with a colleague, noisy background, ...), (vi) users have to type a predefined
 612 password, (vii) knowing that their typing times are saved disturbed some users.

613 Usually, we type our own passwords faster than an imposed one. We have
 614 tested if this error rate of acquiring process is dependant on the user's typing
 615 speed, but it seems that there is no significant correlation (The Pearson correla-
 616 tion factor between typing speed and acquisition error rate is -0.28). Figure 11
 617 represents the acquisition error rates (during the acquisition of the database)
 618 depending on the mean typing speed of users. As can be seen, the experiment
 619 reveals no dependency between these factors. In all intervals, we have high error
 rates.

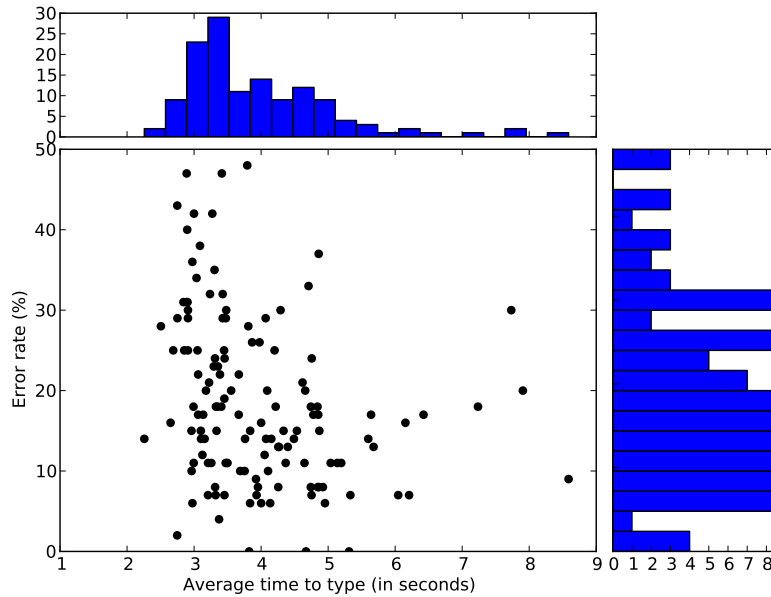


Figure 11: Acquisition error rate depending on the mean typing speed of each user.

620

621 4.3.2. Independence of the keyboard

622 Table 7 represents the *EER* values depending on the keyboard used for
 623 enrollment and verification using the proposed method against six from the
 624 literature? The *EER* value of each method was computed by keeping the first

625 ten samples for enrollment, and the others for the verification process. We did
 626 not use any update mechanism in this experiment and the decision threshold
 627 is the same for all the individuals. When the keyboards used for enrollment
 628 and verification are different, the computation is done several times by selecting
 629 enrolled vectors randomly and averaging the results.

Table 7: Error(%) rates of methods depending on the keyboard configuration. "EERnm" means captures from keyboard "n" for enrollment and captures from keyboard "m" for verification, where "1", "2", "a" stands for keyboard 1, keyboard 2 and no distinction of keyboard respectively. The best *EER* value of each method is presented in italic, while the best *EER* of each configuration is presented in bold.

Method	EER11	EER22	EER12	EER21	EERaa
STAT1	24.91%	23.96%	24.73%	<i>23.51%</i>	25.50%
STAT2	17.68%	<i>16.55%</i>	17.10%	16.65%	17.58%
STAT3	15.10%	13.81%	14.68%	<i>13.22%</i>	15.43%
DIST	27.01%	26.00%	26.46%	<i>25.07%</i>	27.56%
RHYTHM	19.40%	20.09%	<i>19.25%</i>	19.50%	19.78%
NEURAL	12.65%	12.03%	12.15%	<i>11.21%</i>	13.62%
CONTRIB	10.68%	10.37%	<i>10.30%</i>	11.76%	11.96%
Mean	18.20%	17.54%	17.81%	17.27%	18.77%

630 Columns *EER11* and *EER22* represent *EER* values when enrolled and tested
 631 samples belong only to keyboard 1 and keyboard 2 respectively. Column *EER12*
 632 represents the *EER* value computed by using keyboard 1 for enrollment and
 633 keyboard 2 for verification (and vice versa for the column *EER21*). In the
 634 column *EERaa*, samples were used without distinguishing of their origin for
 635 enrollment and verification.

636 We can see that results are rather different, depending on keyboard configu-
 637 ration. Curiously, six times out of seven, the best results are obtained when the
 638 test and enrolled keyboards are different, whereas the best performances were
 639 expected when using the same keyboard for enrollment and verification. Five
 640 times out of seven, the best results are obtained when the enrolled keyboard
 641 is keyboard 2 (the USB keyboard). This can be interpreted as the fact that
 642 templates are more precise when using a classical keyboard instead of a laptop

643 keyboard. The worst results are obtained when using both keyboards for en-
644 rollment and verification. The proposed method outperforms all the other ones
645 for most of the configurations.

646 Using Kruskal-Wallis test on the 5 vectors (EER_{11} , EER_{22} , EER_{12} , EER_{21}
647 and EER_{aa}), we have a p-value equal to 0.9673. This p-value shows that there
648 is no significant difference between the two keyboard during the enrollment and
649 verification phases.

650 We have also tested if it was possible to recognize the keyboard which was
651 used to type the password. By using an SVM with a 10-fold-cross-validation and
652 repeating the process 50 times, we obtain a keyboard recognition accuracy of
653 61.48% with a standard deviation of 0.17. These results are not sufficiently dif-
654 ferent from a random choice to argue that we are able to recognize the keyboard
655 and explains the differences.

656 4.3.3. *Number of Enrolled Templates*

657 An interesting point is that the trend of EER values depends on the number
658 of captures used to create the biometric reference for an individual because in
659 most studies, this number differs. The performance of the algorithms varies
660 depending on the number of samples used to create templates. Most studies
661 used more than twenty captures in order to create the template, whereas we
662 think five samples per user is really the maximum for usability reasons (espe-
663 cially when considering that users can practice at typing the password before
664 saving these samples). Figure 12 represents the EER value of different tested
665 algorithms depending on the number of enrolled patterns. It is clear that the
666 performance increases with the number of enrolled samples in the template. For
667 all the methods, less than ten captures give very bad results. In order to obtain
668 the best results, the required number of enrolled samples seems to be around
669 forty captures (but, in this case, the number of patterns used to test the per-
670 formance is very small and the results are less significant). For some methods,
671 the performance decreases when using more than fifty captures, but it can be
672 due to the fact that not enough samples are provided for the comparison and

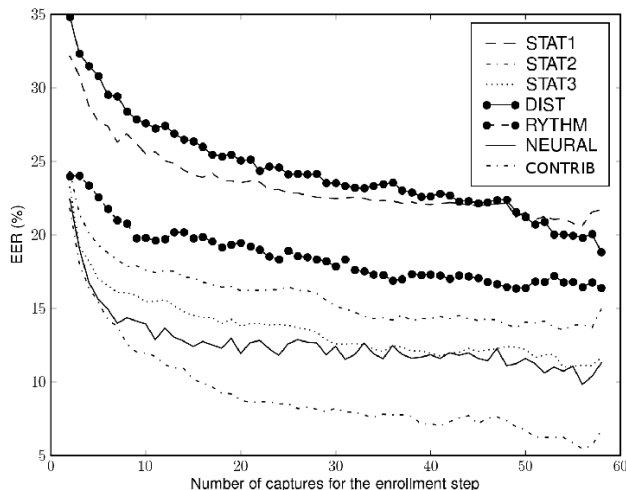


Figure 12: Evolution of the EER of the tested algorithms by considering the number of patterns used to create the template.

673 these results are not statistically relevant. Once again, the proposed method
 674 gives the best results when using more than ten captures. Even if with less than
 675 ten captures, the performances are degraded, our method outperforms most of
 676 the others and can be used in a non critical environment. Therefore, it is not
 677 absurd to use only five captures (see Section 5.1 for more information on the
 678 statistical analysis). It is up to the authentication system to be able to update
 679 the reference on the fly in order to improve the recognition rate.

680 4.3.4. Update of the biometric reference

681 Keystroke dynamics is a behavioral modality and is subject to high intraclass
 682 variability. That is why an *adaptation* (or *update*) mechanism can be applied
 683 in order to improve its performance [28]. This way, the biometric reference
 684 evolves depending on the evolution of user’s manner of typing. We compared the
 685 performance of different algorithms detailed in Section 3.3 with the classic one
 686 (no incremental mechanism). Table 8 presents the EER values of the described
 687 methods obtained by using different incremental mechanisms. These values were

688 computed using five enrolled vectors, the captured data from both keyboards
689 without any distinction and using a global threshold. It is also important to
690 note that the aim of this test is to show if there is an evolution in the way
691 the user types. It is not our objective to decide which is the best method to
692 use to update the template (adaptation is done after the verification of the test
693 vector against all the templates, only with the template of the owner of the test
694 pattern, even if the verification test fails). The presentation of the test vector is
695 as followed: for each user, we test the first test vector against all the templates
696 (we then update the templates). Then, we test the second test vector and so
697 on until having tested all the test patterns. A more operational and realistic
698 method would be to try to adapt the template of the user, if the verification
699 is successful, with all the test vectors (even if this success is an error). This
700 implies setting a decision threshold to obtain the *FAR* and *FRR* values, which
701 requires a lot of computations³.

Table 8: EER(%) values for different incremental mechanisms with five captures for the enrollment step on both keyboards. The best *EER* value for each adaptation method is presented in bold. The templates cannot be adapted with impostors patterns.

Method	Classic	Progressive	Adaptive	Average	Correlation
STAT1	27.7%	21.24%	23%	<i>20.94%</i>	21.67%
STAT2	19.29%	15.09%	11.71%	10.75%	<i>10.39%</i>
STAT3	17.02%	12.57%	9.78%	<i>8.64%</i>	9.21%
DIST	30.81%	23.75%	25.7%	<i>24.65%</i>	24.99%
RHYTHM	22.56%	15.49%	14.36%	13.21%	<i>13.19%</i>
NEURAL	15.79%	<i>8.43%</i>	10.03%	8.75%	10.39%
CONTRIB	15.28%	6.69%	9.21%	6.96%	7.88%

702 We can see in Table 8 that using a template update mechanism improves the
703 performance of the system. For most algorithms, the best update mechanism is
704 the *average* one, even if it can use fewer samples to create the template than the
705 progressive one (where overfitting can occur). Therefore, filtering the captures

³For each of our cases, for each interval of threshold of each method, compute the confusion matrix and get its *FAR* and *FRR*.

706 before adding them to the template improves the performance by reducing the
 707 *EER* by approximately 8%. Our method gives once again the best results and
 708 provides the minimal *EER* value of less than 7% for the progressive mode.

709 *4.3.5. Independence of the Threshold*

710 Using individual thresholds instead of global ones is supposed to improve
 711 the performance of algorithms. Table 9 presents the improvements in term of
 712 *EER* when using individual thresholds. The *EER* were computed using: five
 713 captures to compute the template, the average update method and data from
 714 both keyboards.

715

Table 9: *EER*(%) value for each method when using global and individual thresholds, by using data of both keyboards and an incremental mechanism. The best *EER* value of each method is presented in bold.

Method	EER(global)	EER(individual)	Gains
STAT1	20.94%	<i>19.54%</i>	1.4%
STAT2	10.75%	<i>9.22%</i>	1.53%
STAT3	9.78%	<i>8.64%</i>	1.14%
DIST	24.65%	<i>21.53%</i>	3.12%
RHYTHM	13.21%	<i>10.02%</i>	3.18%
NEURAL	10.3%	<i>8.75%</i>	1.55%
CONTRIB	6.96%	6.95%	0.01%
Mean	13.8%	12.1%	1.7%

716 Automatically configuring the individual threshold with a system using a
 717 shared secret is possible, but it cannot be applied in the case of using a differ-
 718 ent password for each user (nobody would agree to give his own password to
 719 impostors in order to get their samples as attack). A solution to this problem
 720 is presented in [38] where users are classified in different groups depending on
 721 various parameters. These groups are created thanks to a training database,
 722 and each group shares the same parameters of the method computed with the
 723 training databases. Using the Kruskal-Wallis test, we obtain a *p-value* of 0.2774,
 724 which indicates that the gain of using individual thresholds is acceptable but

725 not significant in comparison to the global threshold approach. Nevertheless,
 726 in general, the individual thresholds approach leads to better results (which is
 727 also clearly shown in Table 9).

728 4.3.6. Computation Time

729 The computation time taken to verify a pattern against the template is quite
 730 similar for all the methods, but, it is not the case for the template creation.
 731 Computation times for template creation of all the users (including database
 732 reading) are presented in Table 10. The timings were computed when using 5
 733 and 10 captures to create the template with a python script on a Linux PC
 734 Desktop with an Intel Pentium IV processor with a speed of 3GHz and 1Gb of
 735 RAM.

736 We can see that template creation is quite fast for STAT1, STAT2, STAT3,
 737 DIST, RHYTHM and the timings are not really dependant on the number of
 738 patterns used to create the template. Computation time is higher for CONTRIB
 739 and NEURAL, but CONTRIB remains much faster than NEURAL (almost
 740 seven times faster). All the scripts were written in the Python language (both
 741 the algorithms and evaluation scripts) using the psyco [45] module which speeds
 742 up the execution of Python code (by using mechanisms similar to JIT compiler).
 743 We used the ffnnet [46] library for the neural network and libsvm [47] for the SVM.

Table 10: Computation time involved in biometric reference of all the users creation for each method, when 5 and 10 captures are required to create the template.

Nb	STAT1	STAT2	STAT3	DIST	RHYTHM	NEURAL	CONTRIB
5	3s	3s	3s	3s	4s	5m 55s	54s
10	3s	3s	3s	3s	4s	30m 4s	4m 24s

744 5. Discussion

745 5.1. Confidence intervals

746 The performance difference between each methods could be very small which
 747 implies that these methods are not statistically different. In order to compare

Table 11: Confidence intervals of the EER when using a confidence of 95%.

Method	EER min	EER max	Interval width
STAT1	27.09%	28.42%	1.33%
STAT2	18.69%	19.85%	1.16%
STAT3	16.64%	17.71%	1.07%
DISTANCE	30.12%	31.49%	1.37%
RHYTHME	21.70%	22.95%	1.25%
NEURAL	15.32%	16.40%	1.08%
CONTRIB	14.62%	15.69%	1.07%

748 the algorithms more easily, we can use hypothesis tests or confidence intervals.

749 We computed the confidence interval of the EER when using five samples
 750 for the enrollment, no adaptation scheme and a global threshold. We applied
 751 the method presented in [48]⁴ and obtained the results presented in Table 11.

752 When using no adaptation and only five samples to build the template, our
 753 contribution performs better 90% of the time with the *STAT1*, *STAT2*, *STAT3*,
 754 *DISTANCE* and *RHYTHME* methods (we have 5% of EER outside of the
 755 confidence interval for both methods). There is a small overlap between the
 756 *NEURAL* and *CONTRIB* methods. The proposed method is slightly better in
 757 term of error rate than the *NEURAL* method, but this is not statistically sig-
 758 nificant. The method remains more interesting because template computation
 759 takes less time.

760 5.2. Detector Variability

761 Another new consequent database is available in the keystroke dynamics
 762 research area [33] ; table 12 summarises this database. This database and ours
 763 were constructed with the same objectives, but we some differences are present:

- 764 • we have twice as many users and can obtain results on a higher population
 765 of individuals or can split it in two datasets: one for configuration and one
 766 for validation;

⁴by using a confidence of 95% instead of 90%

- 767 • we obtain more intraclass variability because:
 - 768 – our sessions are more spaced (one week instead of one day) for the
 - 769 majority of users⁵;
 - 770 – there is a break before the user retypes the password. The user does
 - 771 not type a password many times in one shot.
 - 772 – we use two keyboards
- 773 • nevertheless we have less sessions.

774 The authors at [33] tested 14 different anomaly detectors on this database
775 (refer to this work for more information). They presented their results differently
776 from us: a ROC curve is computed for each user and its EER is extracted,
777 then the mean and standard deviation of the EER computed for each user is
778 presented.

779 We used the same protocol in order to observe the behavior of our method on
780 this database (which contains a lot more scores per user). With a global thresh-
781 old, we obtain an EER of 10.63%, while with individual threshold, we obtain an
782 averaged EER of 9.39% (with a standard deviation of 6.72). This would place
783 our method at the first place of their Table 2 (which presents methods ordered
784 by performance) because their best method (Manhattan scaled) gives an EER
785 value of 9.6% (with a standard deviation of 6.9). Our results are also better
786 than their one-class SVM application. These better results can be explained by
787 the fact that we use impostor samples in our method, instead of the anomaly
788 detector which only uses genuine samples in its template.

789 6. Conclusion and Perspectives

790 The keystroke dynamics authentication is an interesting biometric modality
791 as it does not require any additional sensor and is well-accepted by users [11].
792 The performance of such systems for authentication purposes is sufficiently high.

⁵the timestamp of each capture is saved in the database

Table 12: Summary of the information provided in the database presented in [33].

Information	Description
Users	51 users
Database sample size	20400 passphrases (50*8 samples for each user)
Data sample length	10 characters ('.tie5Roanl')
Typing error	not allowed
Controlled acquisition	yes
Age range	between 18 and 70
Gender	30 males & 21 females
PC usage frequency	unknown
User profession	unknown
Keyboard	QWERTY keyboards (laptop)
Acquisition platform	Windows
Timing accuracy	200 microseconds with an external clock

793 The proposed method in this work outperforms all methods in the literature in
 794 deployment conditions (*i.e.*, if the number of captures for the enrollment is
 795 limited to 5) even though the computation time for enrollment remains higher.
 796 We can argue that this method is efficient when users type the same shared
 797 secret to authenticate themselves, and even if the template creation can take
 798 more time, the authentication process is as fast as in the other methods.

799 In order to compare the performance of the proposed method with that of
 800 the other ones, we have created a large database [36] with more than 100 users
 801 with at least 5 sessions for the acquisition phase. This database is available
 802 for the research community (some databases were used in several works [49, 25]
 803 but not used by other researchers or were not made publicly available) and has
 804 allowed us to answer multiple questions.

805 We saw that using individual thresholds could improve the performance of
 806 the system. One of our future works will involve identifying a method allowing
 807 a quick and easy configuration of individual thresholds without impostors' data.
 808 Good robustness was shown for these algorithms for different keyboards. The
 809 benefit of supervised template update mechanisms of the biometric reference
 810 was also demonstrated.

811

812 Several factors have to be tested in the keystroke dynamics domain. This
813 often implies creating a new database especially designed for the corresponding
814 tests (*i.e.*, dependency on the keyboard, computer operating systems, knowledge
815 of the password, size of the password, content of the password). These databases
816 can be created by merging different databases from different researchers or by
817 creating new ones with the help of *GREYC-Keystroke* software.

818 A security analysis of keystroke dynamics will also be an interesting point to
819 explore in the future (*i.e.*, analysis of security problems inherent to the modality
820 or its implementations).

821 **7. Acknowledgments**

822 The authors would like to thank the “Région Basse-Normandie” and the
823 French Research Ministry for their financial support for this work. We would
824 also like to thank all the individuals who have participated to the definition
825 of the keystroke dynamics database, as well as the authors of *libsvm* [47] and
826 *ffnet* [46], the two libraries used during this project. Finally, we would like to
827 thank the reviewers whose suggestions have helped improve the content of this
828 paper significantly.

829 **References**

- 830 [1] A. Conklin, G. Dietrich, D. Walz, Password-based authentication: A sys-
831 tem perspective, in: Proceedings of the 37th Hawaii International Confer-
832 ence on System Sciences, Hawaii, 2004, p. 10.
- 833 [2] A. Peacock, X. Ke, M. Wilkerson, Typing patterns: A key to user identifi-
834 cation, *IEEE Security & Privacy* (2004) 40–47.
- 835 [3] D. Gunetti, C. Picardi, Keystroke analysis of free text, *ACM Transactions*
836 *on Information and System Security (TISSEC)* 8 (3) (2005) 312–347.

- 837 [4] S. Hocquet, J.-Y. Ramel, H. Cardot, User classification for keystroke dy-
838 namics authentication, in: The Sixth International Conference on Biomet-
839 rics (ICB2007), 2007, pp. 531–539.
- 840 [5] M. Obaidat, B. Sadoun, Verification of computer users using keystroke
841 dynamics, Systems, Man and Cybernetics, Part B, IEEE Transactions on
842 27 (1997) 261–269.
- 843 [6] V. Vapnik, Statistical learning theory, NY Wiley, 1998.
- 844 [7] M. Theofanos, B. Stanton, C. A. Wolfson, Usability & Biometrics: En-
845 suring Successful Biometric Systems, National Institute of Standards and
846 Technology (NIST), 2008.
- 847 [8] ISO, Biometric performance testing and reporting, Tech. rep., ISO/IEC
848 19795-1:2006(E) (2006).
- 849 [9] A. Mansfield, J. Wayman, Best practices in testing and reporting perfor-
850 mance of biometric devices, NPL Report CMSC 14 (02).
- 851 [10] J. Bhatnagar, A. Kumar, On estimating performance indices for biometric
852 identification, Pattern Recognition 42 (2009) 1803 – 1815.
- 853 [11] M. El-Abed, R. Giot, B. Hemery, C. Rosenberger, A study of users’ accep-
854 tance and satisfaction of biometric systems, in: 44th IEEE International
855 Carnahan Conference on Security Technology (ICCST’10), San Jose, Cali-
856 fornia, USA, 2010, pp. 1–10.
- 857 [12] ISO, Information technology - security techniques - security evaluation of
858 biometrics, Tech. rep., ISO/IEC 19792-1:2008(E) (2008).
- 859 [13] F. Cherifi, B. Hemery, R. Giot, M. Pasquet, C. Rosenberger, Behavioral
860 Biometrics for Human Identification: Intelligent Applications, IGI Global,
861 2009, Ch. Performance Evaluation Of Behavioral Biometric Systems, pp.
862 57–74.

- 863 [14] M. Sasse, S. Brostoff, D. Weirich, Transforming the 'weakest link' – hu-
864 man/computer interaction approach to usable and effective security, *BT*
865 *Technology Journal* 19 (2001) 122–131.
- 866 [15] I. Winkler, B. Dealy, Information Security Technology?... Don't Rely on It
867 A Case Study in Social Engineering, in: *Proceedings of the Fifth USENIX*
868 *UNIX Security Symposium, 1995*, p. 6.
- 869 [16] R. Gaines, W. Lisowski, S. Press, N. Shapiro, Authentication by keystroke
870 timing: some preliminary results, *Tech. rep.*, Rand Corporation (1980).
- 871 [17] J. D. Garcia, Personal identification apparatus, patent (November 1986).
- 872 [18] V. V. Phoaha, S. Phoha, A. Ray, S. S. Joshi, S. K. Vuyyuru, Hidden markov
873 model (hmm)-based user authentication using keystroke dynamics, patent
874 (2009).
- 875 [19] S. Bleha, C. Slivinsky, B. Hussien, Computer-access security systems using
876 keystroke dynamics, *IEEE Transactions On Pattern Analysis And Machine*
877 *Intelligence* 12 (1990) 1216–1222.
- 878 [20] M. Brown, S. J. Rogers, User identification via keystroke characteristics
879 of typed names using neural networks, *Int. J. Man-Mach. Stud.* 39 (1993)
880 994–1014.
- 881 [21] F. Monrose, Rubin, Authentication via keystroke dynamics, in: *Proceed-*
882 *ings of the 4th ACM conference on Computer and communications security,*
883 1997, pp. 48–56.
- 884 [22] S. Cho, S. Hwang, Artificial rhythms and cues for keystroke dynamics based
885 authentication, in: *IAPR International Conference on Biometrics, Vol. 5,*
886 2006, pp. 626–632.
- 887 [23] R. Rodrigues, G. Yared, C. do NCosta, J. Yabu-Uti, F. Violaro, L. Ling,
888 Biometric access control through numerical keyboards based on keystroke
889 dynamics, *Lecture notes in computer science* 3832 (2006) 640.

- 890 [24] Y. Sang, H. Shen, P. Fan, *Parallel and Distributed Computing: Applications and Technologies*, Springer, 2005, Ch. Novel impostors detection in
891 keystroke dynamics by support vector machine, pp. 666–669.
- 893 [25] E. Yu, S. Cho, Keystroke dynamics identity verification, its problems and
894 practical solutions, *Computers & Security* 23 (2004) 428–440.
- 895 [26] K. Revett, S. de Magalhaes, H. Santos, On the use of rough sets for user
896 authentication via keystroke dynamics, *Lecture notes in computer science*
897 4874 (2007) 145.
- 898 [27] D. Hosseinzadeh, S. Krishnan, Gaussian mixture modeling of keystroke
899 patterns for biometric applications, *Systems, Man, and Cybernetics, Part*
900 *C: Applications and Reviews*, *IEEE Transactions on* 38 (2008) 816–826.
- 901 [28] P. Kang, S.-s. Hwang, S. Cho, Continual retraining of keystroke dynam-
902 ics based authenticator, in: S.-W. Lee, S. Li (Eds.), *Proceedings of ICB*
903 *2007*, Vol. 4642 of *Lecture Notes in Computer Science*, Springer Berlin /
904 Heidelberg, 2007, pp. 1203–1211.
- 905 [29] J. Lee, S. Choi, B. Moon, An evolutionary keystroke authentication based
906 on ellipsoidal hypothesis space, in: *Proceedings of the 9th annual conference*
907 *on Genetic and evolutionary computation*, 2007, pp. 2090–2097.
- 908 [30] K. Revett, A bioinformatics based approach to user authentication via key-
909 stroke dynamics, *International Journal of Control, Automation and Sys-*
910 *tems* 7 (1) (2009) 7–15.
- 911 [31] M. Karnan, M. Akila, N. Krishnaraj, Biometric personal authentication
912 using keystroke dynamics: A review, *Applied Soft Computing* In Press,
913 Corrected Proof (2010) –. doi:DOI: 10.1016/j.asoc.2010.08.003.
- 914 [32] R. Giot, M. El-Abed, C. Rosenberger, Keystroke dynamics with low con-
915 straints svm based passphrase enrollment, in: *IEEE International Confer-*
916 *ence on Biometrics: Theory, Applications and Systems (BTAS 2009)*, 2009,
917 p. 9.

- 918 [33] K. S. Killourhy, R. A. Maxion, Comparing anomaly-detection algorithms
919 for keystroke dynamics, in: 39th Annual International Conference on De-
920 pendable Systems and Networks (DSN-2009), 2009, pp. 125–134.
- 921 [34] K. Revett, S. de Magalhães, H. Santos, Enhancing login security through
922 the use of keystroke input dynamics, *Lecture notes in computer science*
923 3832 (2006) 661.
- 924 [35] E. P. Kukula, C. R. Blomeke, S. K. Modi, S. J. Elliott, Effect of human-
925 biometric sensor interaction on fingerprint matching performance, image
926 quality and minutiae count, *IJCAT* 34 (4) (2009) 270–277.
- 927 [36] R. Giot, M. El-Abed, C. Rosenberger, Greyc keystroke: a benchmark for
928 keystroke dynamics biometric systems, in: *IEEE International Conference*
929 *on Biometrics: Theory, Applications and Systems (BTAS 2009)*, 2009, p. 9.
- 930 [37] B. Scholkopf, A. Smola, *Learning with kernels: Support vector machines,*
931 *regularization, Optimization, and Beyond.* MIT Press 1 (2002) 2.
- 932 [38] S. Hocquet, J.-Y. Ramel, H. Cardot, Estimation of user specific parameters
933 in one-class problems, in: *ICPR '06: Proceedings of the 18th International*
934 *Conference on Pattern Recognition*, 2006, pp. 449–452.
- 935 [39] N. Grabham, N. White, Use of a novel keypad biometric for enhanced
936 user identity verification, in: *Instrumentation and Measurement Technol-*
937 *ogy Conference Proceedings*, 2008. *IMTC 2008. IEEE*, 2008, pp. 12–16.
- 938 [40] R. Giot, M. El-Abed, C. Rosenberger, Keystroke dynamics authen-
939 tication for collaborative systems, in: *The IEEE International Sym-*
940 *posium on Collaborative Technologies and Systems (CTS)*, *IEEE*
941 *Computer Society*, Baltimore, Maryland, USA, 2009, pp. 172–179.
942 doi:10.1109/CTS.2009.5067478.
- 943 [41] C.-W. Hsu, C.-C. Chang, C.-J. Lin, *A practical guide to support vector*
944 *classification* (2010).
945 URL <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

- 946 [42] Y. Sheng, V. Phoha, S. Rovnyak, A parallel decision tree-based method
947 for user authentication based on keystroke patterns, *IEEE Transactions on*
948 *Systems, Man, and Cybernetics, Part B: Cybernetics* 35 (4) (2005) 826–833.
- 949 [43] A. Anagun, Development of committee neural network for computer access
950 security system, *Lecture Notes in Computer Science* 3982 (2006) 11.
- 951 [44] S. Cho, D. H. Han, Chigeun Han, H.-I. Kim, Web-based keystroke dynamics
952 identity verification using neural network, *Journal of organizational com-*
953 *puting and electronic commerce* 10 (2000) 295–307.
- 954 [45] A. Rigo, *Psyco - home page* (2010).
955 URL <http://psyco.sourceforge.net/>
- 956 [46] M. Wojciechowski, Feed-Forward neural network for python, Technical Uni-
957 versity of Lodz (Poland), Departement of Civil Engineering, Architecture
958 and Environmental Engineering, <http://ffnet.sourceforge.net/> (2007).
- 959 [47] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector ma-
960 chines, software available at [http://www.csie.ntu.edu.tw/~cjlin/](http://www.csie.ntu.edu.tw/~cjlin/libsvm)
961 [libsvm](http://www.csie.ntu.edu.tw/~cjlin/libsvm) (2001).
- 962 [48] A. Mayoue, Biosecure tool - performance evaluation of a biometric verifi-
963 cation system (2007).
964 URL [http://svnext.it-sudparis.eu/svnview2-eph/ref_syst/Tools/](http://svnext.it-sudparis.eu/svnview2-eph/ref_syst/Tools/PerformanceEvaluation/doc/howTo.pdf)
965 [PerformanceEvaluation/doc/howTo.pdf](http://svnext.it-sudparis.eu/svnview2-eph/ref_syst/Tools/PerformanceEvaluation/doc/howTo.pdf)
- 966 [49] J. R. M. Filho, E. O. Freire, On the equalization of keystroke timing his-
967 tograms, *Pattern Recognition Letters* 27 (2006) 1440–1446.