Analysis of Keystroke Dynamics For the Generation of Synthetic Datasets

Denis Migdal, Christophe Rosenberger

Analysis of Keystroke Dynamics For the Generation of Synthetic Datasets

Denis Migdal, Christophe Rosenberger
Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC,
14000 Caen, France
{denis.migdal, christophe.rosenberger}@ensicaen.fr

October 17, 2018

Abstract

Biometrics is an emerging technology more and more present in our daily life. However, building biometric systems requires a large amount of data that may be difficult to collect. Collecting such sensitive data is also very time consuming and constrained, s.a. GDPR legislation. In the case of keystroke dynamics, existing databases have less than 200 users. For these reasons, we aim at generating a keystroke dynamics synthetic dataset. This paper presents the generation of keystroke data from known users as a first step towards the generation of synthetic datasets, and could also be used to impersonate users' identity.

Keystroke; Synthetic dataset; Data Analysis;

1 Introduction

Keystroke dynamics (KD) [1] is a behavioral biometric modality that allows the authentication of individuals through their way of typing a password or a free text on a keyboard. It is a behavioral biometrics which has the advantage of not requiring additional sensor than the keyboard. This biometric modality also allows continuous authentication through time [2, 3].

User authentication with keystroke dynamics is generally done in real time (*i.e.*, online) in a real world system. Scientists working on keystroke dynamics do not analyze the performance of their system in an online way (*i.e*, by asking users to authenticate themselves in real time and to impersonate other users). Indeed, they work in an offline context by using samples previously collected by other researchers, and stored in a benchmark dataset. A complete list of available keystroke dynamics datasets has been made in [4, 5]. As it can be seen, most of datasets have less than 200 individuals and few samples for each user. The collection of such datasets is very time consuming, this is the main reason why there is not more very large datasets like for the face modality as for example [6].

In this paper, we present a study whose objective is to model real KD data in order to generate synthetic KD datasets. This approach has been used for the digital fingerprint modality with the SFINGE software [7] as their collection and distribution are regulated in many countries. We believe the KD model will be able to help the research community to create a new dataset of higher quality than the existing ones. We think this work is important, because it is known that KD studies are not fair as (i) acquisition protocols are different between studies [8]; (ii) there is not always a comparative study [9] when authors propose new algorithms; and (iii) there are not always a valuable statistical evaluation [9]. Our work contributes to solve these problems.

The paper is organized as follows. Section II is dedicated to some background information on Keystroke dynamics and existing studies for this biometric

modality. We present in section III the definitions and the components of the KD model we propose in this study. Section IV concerns the evaluation of the proposed KD model on two real KD datasets and 4 matching methods. We analyze in section V the keystroke dynamics data of these two datasets to set the parameters of the proposed KD model. Section VI is dedicated to the validation process of the KD model showing the capability to generate similar keystroke dynamics data. Last, section VII concludes this work and gives some perspectives.

2 Related works

A keystroke dynamic system (KDS) is composed of two main modules: the enrollment and the verification modules. Each user must enroll himself/herself in the KDS which computes a biometric reference given multiple samples (*i.e.*, several inputs of the password) acquired during the enrollment step. For each input, a sequence of timing information is captured (*i.e.*, time when each key is pressed or released) from which some features are extracted (*i.e.*, latencies and durations) and used to learn the model which characterizes each user. During a verification request, the claimant types his/her password. The system extracts the features and compares them to the biometric reference of the claimant. If the obtained distance is below a threshold, the user is accepted, otherwise he/she is rejected.

First works on KD have been done in the eighties [10], although the idea of using a keyboard to automatically identify individuals has first been presented in 1975 [11]. In the preliminary report of Gaines *et al.* [10], seven secretaries typed several paragraphs of text and researchers showed that it is possible to differentiate users with their typing patterns. Since then, several studies have been done, allowing to decrease the quantity of information needed to build the biometric reference, while improving the performances [8, 12–15]. However, most studies are not comparable because they use different datasets or protocols [8, 9].

3 Keystroke dynamics generative model

First, we define many terms to build the proposed model:

- **Digraph:** $D = [C_0, C_1]$, array of two characters.
- **DigraphTime:** $DT_D = [d_0, d_1, d_2, d_3, d_4, d_5]$, as shown in Figure 2, array of 6 durations from 4 times corresponding to the pressure (P) and release (R) times of each character of a Digraph D. A DigraphTime DT_D is defined as partially consistent if the following equations are verified, consistent if the following equations and inequalities are verified, and inconsistent otherwise:
  - $d_0 = d_2 - d_4$;
  - $d_0 = d_1 - d_3$;
  - $d_1 = d_2 - d_5$;
  - $d_3 = d_4 - d_5$;
  - $d_0 \geq 0$
  - $d_1 \geq 0$
  - $d_5 \geq 0$
- **Text:** $T_n = \{D_i\}_{i \in [\![0,n[\![}$, an array of n Digraphs D_i. A text T_n is said consistent if $\forall i \in [\![0, n[\![, D_{i-1}[1] = D_i[0]$.
- **Keystroke dynamics:** $K = [\{DT_i\}_{i \in [\![0,n[\![}, T_n]$, an array of n DigraphTime DT_i associated to the Digraph $T_n[i]$. Keystroke is said consistent (or partially consistent) if T_n, and all DT_i are consistent (or partially consistent), and if $\forall i \in [\![0, n[\![, DT_{i-1}[5] = DT_i[0]$.

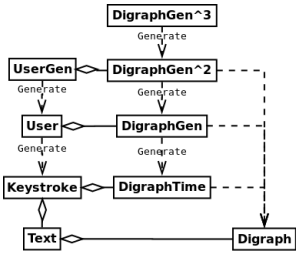

Figure 1: KD Generative model

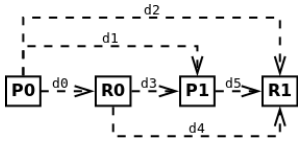

Figure 2: DigraphTime

We propose in this paper a generative keystroke dynamics model. We explain its different components (see also Figure 1 and 2):

- **DigraphGen:** $DG_D() = DT$, generates a DigraphTime for a given Digraph.
- **User:** $U(T_n) = K$, generates a Keystroke dynamics from a given Text. A User is composed of a set of DigraphGen.
- **DigraphGen^2:** $DG^2_D() = DG_D$, generates a DigraphGen for a given Digraph.
- **UserGen:** $UG() = U$, generates a User. A UserGen is composed of a set of DigraphGen^2.
- **DigraphGen^3:** $DG^3(D) = DG^2_D$, generates a DigraphGen^2 for a given digraph.

4 Evaluation

In the scope of this study, two datasets are used : Data1 (GREYC-KeyStroke Dataset [16]) & Data2 (GREYC-Web based KeyStroke dynamics dataset [17]). These datasets are listed in [4] and are enough representative of real world keystroke dynamics data. From these two datasets, the following sub-datasets are created:

- *Data{1,2}D:* DigraphTimes for each User and Digraph extracted from Data{1,2}.
- *Data{1,2}K:* a fixed Text's Keystrokes for each User, extracted from Data{1,2}. Two fixed Texts being present in Data2, extracted sub-datasets are thus Data2K1 and Data2K2.

The comparison between keystroke dynamics samples is evaluated in this paper through 4 distance functions:

We suppose to compute a distance between two templates K_A and K_B, these functions are defined as follows:

- Blesha [18]: We suppose that the template K_A is associated by μ the average value of biometric samples:

$$STAT1 = \frac{(K_B - \mu)^t (K_B - \mu)}{||K_B||.||\mu||} \quad (1)$$

- Hocquet [19]: We suppose that the template K_A is associated by μ and σ the average value of biometric samples and the standard deviation:

$$STAT2 = 1 - \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{|K_B(i)-\mu_i|}{\sigma_i}} \quad (2)$$

- Monrose [20]: The function is given as follows:

$$STAT3 = \sqrt{\sum_{i=1}^{n}(K_B(i) - K_A(i))^2} \quad (3)$$

- BioHashing: This algorithm is a template protection scheme [21] where the biometric template is projected given a key and quantized. The comparison is realized with the Hamming distance. We apply this protection scheme and compare the templates in the transformed domain.

In the scope of this paper, the BioHashing and Monrose distances between a template (sample) and a set of templates (references) are computed as the minimal distance of the sample with each template in the reference gallery.

5 Analysis of keystroke dynamics data

As previously seen, generating a keystroke dynamics template from a given Text T_n consists in generating an array of DigraphTime, i.e. generating $6 * n$ durations. To be able to generate a keystroke dynamics similar to that one user could type, these $6 * n$ durations have to be transformed into a set of assumed independent variables which laws and parameters can then be estimated for a user, enabling their random generation, and thus the computation of a synthetic keystroke dynamics corresponding to a given user. In the scope of this paper, only the linear (in)dependency of variable is considered.

5.1 Variables (in)dependency

Linearly correlated variables can be transformed into a set of non-linearly correlated variables, through

PCA (Principal component analysis), first introduced by Pearson in 1901 [22]. However, we show that durations are not strongly correlated between them, and thus, in the scope of this article, assume them to be independent. Even if the usage of PCA is irrelevant in such a case, its first step enables the computation of the inter-correlations of two variables by the computation of a correlation matrix. In a correlation matrix $C = \{C_{i,j}\}_{\{i,j\}\in[\![0,n[\![^2}$, $C_{i,j}$ is the linear correlation between the variables i and j. A correlation matrix $C = \{C_{i,j}\}_{\{i,j\}\in[\![0,n[\![^2}$, with $C_{i,j}$ the linear correlation between the variables i and j, is computed as follows:

1. Given a matrix $M = \{M_k\}_{k\in[\![0,K[\![}$ of K entries $M_k = \{M_{k,i}\}_{i\in[\![0,n[\![}$, with $M_{k,i}$ the realization of the variable i for the entry k.
2. $\bar{M} = \{\frac{M_{k,i}-\mu_i}{\sigma_i}\}_{i\in[\![0,n[\![,k\in[\![0,K[\![}$ where μ_i is the mean of $\{M_{k,i}\}_{k\in[\![0,K[\![}$, and σ_i, its standard deviation.
3. $C = 1/K * \bar{M}^T * \bar{M}$

In the scope of this paper, we arbitrary consider that sets with less than a certain number of elements cannot provide pertinent results, and are thus discarded. We used 23 as arbitrary value in this paper (cf section 5.2).

To qualify presence of specific correlations between two variables i, j inside m subsets of entries, m correlations matrix $C^l, l \in [\![0,m[\![$ are computed from such subsets. Each element $C_{i,j}$ of the final correlation matrix C is then computed as the mean of each $C^l_{i,j}$: $C_{i,j} = \frac{1}{m}\Sigma_{l=0}^{m-1}C^l_{i,j}$. If each subset corresponds to, e.g. a User, M will be said, in this paper, "splitted by User", and C will qualify the presence of User-specifics correlations across all Users.

To qualify presence of the same correlations between two sets of variables $\{i_x\}_{x\in[\![0,m[\![}, \{j_x\}_{x\in[\![0,m[\![}$, of length m, entries are splitted in m sub-entries $M'_{m*k+x} = \{M_{k,o_x}\}_{o\in\{i,j\}}$. The correlation matrix C is then computed from M'. If each x corresponds to, e.g. a Digraph, M will be said, in this paper, "merged by Digraph", and C will qualify the presence of non-Digraphs-specifics correlations across all Digraphs.

5.2 Laws followed by Variables

Once the variables assumed independent, or transformed in such a way, laws followed by each variable are searched through the following process:

1. Given the realizations of a variable X, and a law law_p with unknown parameters p;
2. Estimate p from the median, mean, min, max, or/and standard deviation of X;
3. Compute the $\chi^2(X, law, p)$ score qualifying the capacity of X to match the values that would be expected if X follows law_p through a χ^2 test.

The χ^2 test qualifies the capacity of a set of observed values to match a set of expected values. The χ^2 test returns $\chi^2(X, law, p) = 1 - \alpha$, in which α is the p-value, i.e. the probability to obtain the same $1 - \alpha$ score if X follows law_p. If the p-value is below an arbitrary threshold (s.a. 0.05), the hypothesis "X follows law_p" can then be rejected.

However, in the scope of this paper, we do not aim to reject the hypothesis, but to select laws that seem to best represent X. The $\chi^2(X, law, p)$ score can then be seen as a score of distance between observed values of X, and the expected values. We compute $\chi^2(X, law, p)$ as follows:

1. Let $Card(S)$ be the cardinal of S;
2. Let $a\%b$ be the rest of the division of a by b;
3. $\mathbb{R}$ is divided in $n = \lceil Card(X)/5 \rceil$ subspaces $E_i, i \in [\![0,n[\![$, each expected to contain 5 elements of X. E_{n-1} is expected to contain $Card(X)\%5$ elements of X if $5 \nmid Card(X)$;
4. Let $X_i = X \cap E_i$;
5. Let $Card(E_i) = 5$, and $Card(E_{n-1}) = Card(X)\%5$ if $5 \nmid Card(X)$;
6. Let $Sum = \Sigma_{i=0}^{n-1}(Card(E_i) - Card(X_i))^2/Card(E_i)$.
7. Let cdf_f be the cumulative distribution function of the law χ^2 of freedom f;
8. $\chi^2(X, law, p) = cdf_{n-Card(p)-1}(Sum)$.

In the scope of this paper, we arbitrary consider that sets with less than a certain number of elements cannot provide pertinent results, and are thus discarded. We used 23 as arbitrary value in this paper, thus having at least 5 subspaces and 100 users.

From tested laws, the best 3 are selected i.e. the 3 laws that minimize the score of distance $\chi^2(X, law, p)$. These laws are tested with and without exclusion of aberrant values (here, X values that differ from $\pm 3\sigma$ from the median value of X). When the parameters p have different estimations, only the one that minimizes the score of distance is kept. A set of 19 laws have been tested in this paper:

- arcsine
- beta
- betaprime
- chi
- chisquare
- cosine
- erlang
- exponential
- f
- gamma
- gumbel
- laplace
- logistic
- lognormal
- uniform
- normal
- rayleigh
- t
- triangular

To qualify the capacity of n subsets of X, $X_i, i \in [\![0, n[\![$, to follow a same law law, but each with different parameters p_i, the score of distance $\chi^2(X, law)$ is computed as the mean of the χ^2 test applied on each X_i: $\frac{1}{n}\Sigma_{i=0}^{n-1}\chi^2(X_i, law, p_i)$.

6 Experimental observations

In this section, we analyze the statistics of real keystroke dynamics from the datasets presented in section IV.

6.1 Durations correlations

First, diagonals of correlation matrix are discarded. Correlations between two durations $DT_{D_i}[5]$, and $DT_{D_j}[0]$ are discarded if $j = i + 1$ in Data{1,2}K datasets, or if $D_1[1] = D_2[0]$ in Data{1,2}D datasets, as they are in fact, or might be, the same duration. Digraph are considered equal:

- in Data{1,2}K datasets, if their positions in the Keystroke are equals.
- in Data{1,2}D datasets, if their content are equals.

As shown in Table I, no strong stable correlation has been found between durations of DigraphTime from different Digraph, (Out: Data{1,2}K, Out_U: Data{1,2}K splitted by User). DigraphTime will be thus assumed independent. Also, no strong stable correlation implying durations d_0 and d_5 of a same DigraphTime has been found (05_K: Data{1,2}K splitted by User, 05_D: Data{1,2}D merged and splitted by Digraph 05: Data{1,2}D merged by Digraph).

Stable correlations have been detected between durations d_1, d_2, d_3, d_4 of a same DigraphTime ($\overline{05}$: Data{1,2}D merged by Digraph). This may be due to the fact each of theses durations can be written as $d_x = d_3 + k_x * d_0 + l_x * d_5$ with $l_x \in \{0, 1\}$, $k_x \in \{0, 1\}$, and $\sigma(d_3) \approx 3 * \sigma(d_0 + d_5)$. In the scope of this paper, DigraphTime is assumed to be computable from 3 independent durations (ideally d_0, d_5, d_3).

6.2 Durations laws

For the 6 DigraphTime durations $DT_D[i]$, $i \in [\![0, 6[\![$, the 5 best laws that minimize $\chi^2(DT_D[i], law)$, with parameters depending on the Digraph and User, are presented in Table 2. DigraphTime durations will then be assumed to best follow either a gumbel, normal, or logistic law, which parameters depends on the User and Digraph. In order to reduce the number of possible combinations, if a DigraphTime duration is generated with a given law, all other durations will be generated by the same law, but with different parameters.

We can see clearly in Table 2 that the estimated laws and parameters for all DigraphTime durations are quite similar for the two datasets we used in this study. Thanks to these statistical observations, we propose a generative model of keystroke dynamics data in the next section.

	05_D	05	$\overline{05}$	05_K	Out_U	Out
$Card(\{C_{i,j} \in C, \lvert C_{i,j}\rvert \geq 0.95\})$	0/0	0/0	**12/12**	0/0/0	0/0/0	0/0/0
$Card(\{C_{i,j} \in C, \lvert C_{i,j}\rvert \geq 0.75\})$	0/0	0/0	12/12	0/0/0	0/0/0	0/0/0
$Card(\{C_{i,j} \in C, \lvert C_{i,j}\rvert \geq 0.50\})$	0/0	0/0	12/12	0/4/0	0/4/0	42/42/12
$Card(\{C_{i,j} \in C\})$	18/18	18/18	12/12	270/288/90	7532/8610/712	7532/8610/712
$\max(\{\lvert C_{i,j}\rvert, C_{i,j} \in C\})$	0.4/0.39	0.38/0.39	**1/1**	0.31/0.54/0.34	0.31/0.54/0.66	0.6/0.72/0.66

Table 1: Correlations found in Data1D/Data2D, and in Data1K/Data2K1/Data2K2 datasets.

Datasets	Rank	d_0	χ^2	d_1	χ^2	d_2	χ^2	d_3	χ^2	d_4	χ^2	d_5	χ^2
Data1D	1	cosine (3σ)	0.964	**gumbel** (3σ)	0.808	**gumbel** (3σ)	0.795	**gumbel** (3σ)	0.820	**gumbel** (3σ)	0.820	cosine (3σ)	0.965
	2	**normal** (3σ)	0.966	**normal** (3σ)	0.822	**normal** (3σ)	0.809	**normal** (3σ)	0.834	**normal** (3σ)	0.833	cosine	0.968
	3	cosine	0.967	**logistic** (3σ)	0.831	**logistic** (3σ)	0.812	cosine (3σ)	0.842	**logistic** (3σ)	0.841	**normal** (3σ)	0.969
	4	**logistic**	0.967	gamma (3σ)	0.836	betaprime (3σ)	0.820	**logistic** (3σ)	0.842	cosine (3σ)	0.844	**normal**	0.969
	5	**normal**	0.967	cosine (3σ)	0.839	cosine (3σ)	0.820	gamma (3σ)	0.863	gamma (3σ)	0.863	**logistic**	0.970
Data2D	1	**normal** (3σ)	0.869	**gumbel** (3σ)	0.810	**gumbel** (3σ)	0.781	**gumbel** (3σ)	0.835	**gumbel** (3σ)	0.82	**normal** (3σ)	0.874
	2	cosine (3σ)	0.872	**normal** (3σ)	0.820	**normal** (3σ)	0.794	**normal** (3σ)	0.838	**normal** (3σ)	0.832	**logistic** (3σ)	0.881
	3	**logistic** (3σ)	0.880	**logistic** (3σ)	0.880	betaprime (3σ)	0.800	**logistic** (3σ)	0.848	betaprime (3σ)	0.835	cosine (3σ)	0.884
	4	gamma (3σ)	0.893	gamma (3σ)	0.834	gamma (3σ)	0.804	cosine (3σ)	0.858	gamma (3σ)	0.836	gamma (3σ)	0.900
	5	**gumbel** (3σ)	0.893	betaprime (3σ)	0.834	**logistic** (3σ)	0.807	betaprime (3σ)	0.890	**logistic** (3σ)	0.839	betaprime (3σ)	0.879

Table 2: Top 5 results of χ^2 test with 19 laws, with (3σ) and without exclusion of abherant values

7 Keystroke dynamics generative model

7.1 Principles

As seen in the previous sections, DigraphTime durations follow either a gumbel, a normal, or a logistic law which parameters can be estimated for each known User and Digraph. For a given User and Digraph, a DigraphGen can be then implemented as a set of 6 random engines generating the 6 DigraphTime durations with the chosen law and estimated parameters.

We propose 10 consistency strategies, 1 for inconsistent DigraphTime, in which all durations are randomly generated (u), and 9 for partially-consistent DigraphTime, in which 3 durations are computed from the 3 others. The durations to compute can be chosen among the 8 following lists, and be used for all Digraph and User, or be randomly chosen (n) for each new DigraphTime to generate:

- 0: $d_3d_4d_5$
- 1: $d_2d_3d_5$
- 2: $d_2d_3d_4$
- 3: $d_1d_4d_5$
- 4: $d_1d_3d_4$
- 5: $d_1d_2d_5$
- 6: $d_1d_2d_4$
- 7: $d_2d_1d_3$

Once the DigraphGen created for a given User, the keystroke dynamics of a given Text T_n is generated through the following process:

1. $K[1] = T_n$
2. $\forall i \in [\![0, n[\![, K[0][i] = DT_{T_n[i]} = DG_{T_n[i]}()$.
   Before the consistency strategy application, and if Keystroke is expected to be consistent (or partially consistent), the DigraphTime first duration $K[0][i][0]$ is settled, if exists (i.e. if $i > 0$), to the last duration of the previous DigraphTime $K[0][i-1][5]$.

7.2 Results analysis protocol

Synthetic datasets, SData{1,2}K_L_CS, are generated, for each law L, and consistency strategy CS, from each dataset Data{1,2}K. For each User of Data{1,2}K, the same number of entries (Keystroke dynamics) it has in Data{1,2} are generated (as seen in the previous section) and inserted into SData{1,2}K_L_CS. For each User, the first 10 entries are used as reference templates, the others as

samples. User with less than 25 entries are discarded (i.e. with less than 15 samples).

For each dataset SData{1,2}K_L_CS, and distance function DistFct, 3 datasets are computed:

- *DataSU:* to qualify the capacity of synthetic Keystroke dynamics to be indistinguishable from real Keystroke dynamics;
- *DataU:* to qualify the KDS performance with real Keystroke dynamics data;
- *DataS:* to qualify, in comparison with DataU, the capacity of synthetic datasets to match the KDS performance that would be expected with real Keystroke dynamics data.

These datasets are composed of legitimate and impostor scores, computed with the distance function DistFct. Legitimate scores are obtained by comparing the reference template with samples from the same user. Impostors scores are obtained by comparing the reference template of users with samples from other users. DataU is computed from Data{1,2}K, and DataS, from SData{1,2}K_L_CS. In DataSU, legitimate scores are legitimate scores of DataU, and impostors scores are the distance, for each User, between real user templates, and its synthetic samples.

We consider the False Acceptance Rate (FAR) describing the ratio of accepted impostor data, the False Rejection Rate (FRR) describing the ratio of falsely rejected legitimate users. The Equal Error Rate (EER) corresponds to configuration of the biometric system when FAR equals FRR. The FAR/FRR points, and the EER are computed from EvaBio [23] considering 25,000 thresholds. Each final indicator is computed by averaging the computed indicators of each Data{1,2}K datasets.

7.3 Usurpation of keystroke dynamics

The EER value computed from DataSU is used to qualify the capacity of synthetic Keystroke dynamics data to be indistinguishable from real Keystroke dynamics data. An EER of x% means that it is not possible to choose a threshold, such as rejecting less than x% genuine users, without accepting less than x% EER impostors. Thus, an EER of 50% means that, for this threshold, the choice to accept or reject a user is not better than random. An EER $> 50\%$ means that, for this threshold, more impostors will be accepted than genuine users.

BioHashing	**Blesha**	**Hocquet**	**Monrose**
gumbel6 50.4%	gumbelu 52.5%	gumbelu 54.1%	gumbel6 52.9%
normal6 49.9%	gumbel4 52.4%	logisticu 53.1%	normal6 52.2%
gumbel7 49.6%	gumbel2 52.3%	normalu 52.0%	gumbel7 51.9%
gumbel4 49.2%	gumbel6 52.3%	gumbel2 51.8%	gumbel4 51.6%
normal7 48.7%	normal6 52.3%	logistic2 50.9%	normal7 51.2%

Table 3: Mean of DataSU EER (Top 5)

As shown in table 3, synthetic Keystroke dynamics data are indistinguishable from real one ($EER \approx 50\%$), when using the chosen distance functions, and are even more, for some configurations, more accepted than real Keystrokes. The fact that synthetic data are more accepted than real data, can be explained either by a lesser intra-score variance, or by a lesser intra-score mean, for synthetic data compared to real data.

Consistency strategy seems to matter more than the used law. Strategies that computes d_5 (0, 1, 3, 5) instead of randomly generating it have a worst EER value than the strategy which randomly choose the durations to be computed, which is worst than other strategies. Results can also be divided into 4 groups in which the laws giving the greater EER values are, in order of superiority, gumbel, normal, and logistic laws.

As shown by the symmetric of the FAR/FRR curves in Figure 3, our proposed Keystroke generation method is thus able to produce synthetic samples that enable identity usurpation of a known user, by imitating its keystroke dynamics.

7.4 Scores estimations

A χ^2 test is performed, in a similar way than previously, between the legitimate and the impostor scores of DataSU, to qualify the capacity of synthetic samples of having the same scores than real samples. However, as it can be seen in Figure 4, synthetic sam-

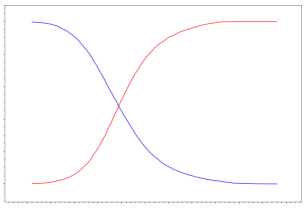

Figure 3: FAR/FRR curves of synthetic samples against real samples compared to real references with Hocquet distance (DataSU from SData2K1_gumbel_6)

Figure 4: Hocquet distances of real (grey) and synthetic (black) Keystrokes samples with real templates (DataSU from SData2K2_gumbel_6)

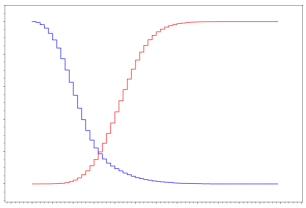

Figure 5: FAR/FRR curves for a real dataset with BioHashing distances (DataU from Data2K1)

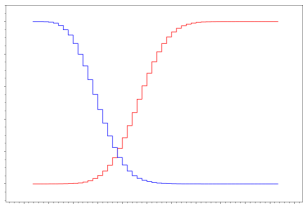

Figure 6: FAR/FRR curves for a synthetic dataset with BioHashing distances (DataS from SData2K1_logistic_0)

ples scores are close to real samples scores, but not enough for a χ^2 test to return a score different from 1.

We thus demonstrate that even being close to real samples scores, our Keystroke generation method is not able yet to produce synthetic samples which scores would match real samples.

7.5 EER and FAR/FRR estimations

The absolute and relative distances between the EER values, computed from the synthetic (DataS) and real (DataU) entries, qualify the capacity of the synthetic datasets to estimate the EER value of the real one.

As shown in Table 4, the EER value can be estimated with a variable accuracy (errors from 0.34 to 28.43). The logic found in previous sections is not respected, and even inverted. Indeed, consistency strategies 0, 1, 3, 5 and n, best estimate the EER value than strategies 7, 6, 4, and 2, the worst strategy being u.

As shown in Figures 5 and 6, our proposed Keystroke generation is thus able to estimate EER values of real dataset. However, as shown in Table 4, the best consistency strategies to estimate EER value are the worst to generate synthetic samples scores close to real samples scores.

8 Conclusion

In this paper, we presented a method that enables the generation of synthetic keystroke dynamics data from known Users, to either usurp real user KD, or to estimate EER value of a KDS. These methods have been tested on fixed text, but could be as well applied to free text.

This work constitutes a first step towards the generation of large synthetic Keystroke dynamics datasets. The following step would be the generation of keystroke dynamics data for an unknown user. Such large synthetic Keystroke dynamics datasets could then be used to fairly compare KDS performances, as well as to improve learning-based KDS' performances.

We also aim to improve the current method with a better estimation of laws parameters, either through an non-User specific PCA on DigraphTime durations, a better estimation of statistics, or the usage of several laws (e.g. cosine for d_0, d_5, and gumbel for d_3).

References

[1] R. Giot, M. El-Abed, and C. Rosenberger, "Keystroke dynamics overview," in *Biometrics / Book 1*, D. J. Yang, Ed. InTech, Jul. 2011, vol. 1, ch. 8, pp. 157–182. [Online]. Available: http://www.intechopen.com/articles/show/title/keystroke-dynamics-overview

Rank	**BioHashing** 19.14/29.28%			**Blesha** 31.67/50.00%			**Hocquet** 15.92/33.76%			**Monrose** 21.44/27.35%		
1	logistic0	0.34	1.44%	logistic1	6.82	17.96%	normal0	3.16	14.31%	gumbeln	1.84	7.39%
2	gumbel0	0.93	4.18%	normal3	7.09	18.70%	normal5	3.61	16.18%	normaln	2.04	8.29%
3	normal0	1.28	5.81%	logistic3	7.11	18.75%	logistic0	4.01	18.02%	logisticn	2.10	8.49%
4	logistic5	1.89	7.85%	gumbel3	7.50	19.51%	logistic5	4.45	19.12%	logistic0	2.69	10.78%
5	gumbeln	1.82	8.58%	gumbel1	7.68	20.03%	gumbel0	5.12	22.98%	gumbel0	3.41	13.85%

Table 4: Absolute and relative distance between synthetic and real EER (Top 5)

[2] S. Mondal and P. Bours, "A study on continuous authentication using a combination of keystroke and mouse biometrics," *Neurocomputing*, vol. 230, pp. 1–22, 2017.

[3] B. Li, H. Sun, Y. Gao, V. V. Phoha, and Z. Jin, "Enhanced free-text keystroke continuous authentication based on dynamics of wrist motion," in *Information Forensics and Security (WIFS), 2017 IEEE Workshop on*. IEEE, 2017, pp. 1–6.

[4] V. Monaco, "Public keystroke dynamics datasets," 2018. [Online]. Available: http://www.vmonaco.com/keystroke-datasets

[5] R. Giot, B. Dorizzi, and C. Rosenberger, "A review on the public benchmark databases for static keystroke dynamics," *Computers & Security*, vol. 55, pp. 46–61, 2015.

[6] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in face detection and facial image analysis*. Springer, 2016, pp. 189–248.

[7] R. Cappelli, D. Maio, and D. Maltoni, "Sfinge: an approach to synthetic fingerprint generation," in *International Workshop on Biometric Technologies (BT2004)*, 2004, pp. 147–154.

[8] R. Giot, M. El-Abed, B. Hemery, and C. Rosenberger, "Unconstrained keystroke dynamics authentication with shared secret," *Computers & Security*, vol. 30, no. 6-7, pp. 427–445, Sep. 2011.

[9] K. S. Killourhy and R. A. Maxion, "Should security researchers experiment more and draw more inferences?" in *4th Workshop on Cyber Security Experimentation and Test (CSET'11)*, Aug. 2011, pp. 1–8.

[10] R. Gaines, W. Lisowski, S. Press, and N. Shapiro, "Authentication by keystroke timing: some preliminary results," Rand Corporation, Tech. Rep. R-2567-NSF, May 1980.

[11] R. Spillane, "Keyboard apparatus for personal identification," IBM Technical Disclosure Bulletin, Apr. 1975.

[12] D. Umphress and G. Williams, "Identity verification through keyboard characteristics," *Internat. J. Man Machine Studies*, vol. 23, pp. 263–273, 1985.

[13] F. Monrose and A. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Syststems*, vol. 16, no. 4, pp. 351–359, 2000.

[14] K. Revett, F. Gorunescu, M. Gorunescu, M. Ene, S. d. M. Tenreiro, and H. M. D. Santos, "A machine learning approach to keystroke dynamics based user authentication," *International Journal of Electronic Security and Digital Forensics*, vol. 1, pp. 55–70, 2007.

[15] H. Lee and S. Cho, "Retraining a keystroke dynamics-based authenticator with impostor patterns," *Computers & Security*, vol. 26, no. 4, pp. 300–310, 2007.

[16] R. Giot, M. El-Abed, and C. Rosenberger, "Greyc keystroke: a benchmark for keystroke dynamics biometric systems," in *IEEE International Conference on Biometrics: Theory, Ap-*

plications and Systems (BTAS 2009), 2009, pp. 1–6.

[17] ——, "Web-Based Benchmark for Keystroke Dynamics Biometric Systems: A Statistical Analysis," in *The Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2012)*, 2012.

[18] S. Bleha, C. Slivinsky, and B. Hussien, "Computer-access security systems using keystroke dynamics," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 12, pp. 1217–1222, 1990.

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

[20] F. Monrose and Rubin, "Authentication via keystroke dynamics," in *Proceedings of the 4th ACM conference on Computer and communications security*, 1997, pp. 48–56.

[21] A. Teoh, D. Ngo, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern recognition*, vol. 40, 2004.

[22] L. KPFRS, "On lines and planes of closest fit to systems of points in space," in *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (SIGMOD)*, 1901.

[23] J. Mahier, B. Hemery, M. El-Abed, M. T. El-Allam, M. Y. Bouhaddaoui, and C. Rosenberger, "Computation evabio: A tool for performance evaluation in biometrics," vol. 3, pp. 51–60, 01 2011.