Graphical Password-Based User Authentication With Free-Form Doodles

Marcos Martinez-Diaz, Julian Fierrez, and Javier Galbally

***Abstract*—User authentication using simple gestures is now common in portable devices. In this work, authentication with free-form sketches is studied. Verification systems using dynamic time warping and Gaussian mixture models are proposed, based on dynamic signature verification approaches. The most discriminant features are studied using the sequential forward floating selection algorithm. The effects of the time lapse between capture sessions and the impact of the training set size are also studied. Development and validation experiments are performed using the DooDB database, which contains passwords from 100 users captured on a smartphone touchscreen. Equal error rates between 3% and 8% are obtained against random forgeries and between 21% and 22% against skilled forgeries. High variability between capture sessions increases the error rates.**

***Index Terms*—Dynamic time warping (DTW), Gaussian mixture models (GMMs), gesture recognition, graphical passwords, mobile security.**

I. Introduction

The term "graphical password" refers to a user authentication method where pictorial information is used for validation, instead of an alphanumerical password. This method poses many challenges, such as memorability (which refers to how easy the password is to remember), usability, and security, since graphical passwords may tend to be visually simple and easily forged [1].

Graphical passwords have become popular due to the proliferation of touchscreen devices, in particular smartphones and tablets. The prevalent approaches are based on simple graphical passwords, which can be easily remembered and reproduced by potential attackers. In this work, we study user authentication based on finger-drawn doodles (i.e., free-form gestures or sequences of gestures) and on pseudosignatures, which are simplified versions of the signature drawn with the fingertip (see Fig. 1). Authentication is based on features extracted from the dynamics of the gesture drawing process (e.g., speed or acceleration). These features contain behavioral biometric information, which has been successfully used for automatic user verification based on handwritten signatures [2]. As a consequence, a potential attacker would have to copy not only *what* the user draws, but also *how* the user draws it. Unfortunately, graphical passwords tend to be much simpler than signatures and are not composed, in general, of previously learned or heavily practiced movements. This can lead to a higher intrauser variability (i.e., variations between samples produced by the same person) than in the case of signatures or may cause users to forget part of or the whole graphical password. On the other hand, while users may be concerned about their privacy when registering their signature on an authentication system, doodles can be a potential solution to overcome this concern. Doodles also have high revocability compared with signatures.

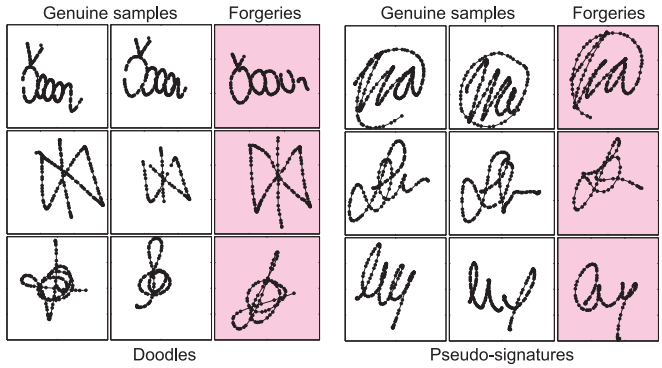

Fig. 1. Examples of doodles and pseudosignatures from the DooDB database [3].

To the extent of our knowledge, this is the first analysis of user authentication on touchscreens based on free-form gestures, using a publicly available database (DooDB Graphical Password Database [3]). The contributions of this paper are as follows.

1) Two approaches from the signature verification state of the art, namely Gaussian mixture models (GMM) and dynamic time warping (DTW), are evaluated using graphical passwords. We analyze the performance of these systems against random forgeries (when attackers claim to be another user but use their own password) and intentional forgeries (when attackers have visual access to the password being forged).
2) Feature selection identifies which features provide the highest discriminative power.
3) The effects of intersession variability (i.e., the time lapse between enrollment and authentication) are studied.
4) We study the impact of the number of available training samples during enrollment on the verification performance.
5) An improved authentication system based on the fusion of GMMs and DTW is presented.

The paper is structured as follows. In Section II, the state of the art is summarized. In Section III, the proposed verification systems are described. Experiments and results are reported in Section IV, and conclusions are drawn in Section V.

II. Related Work

Graphical passwords can be classified into three categories: 1) recall; 2) recognition; and 3) cued-recall. In recall-based systems, users have to remember a graphical password and provide it during authentication. This approach is followed in

Manuscript received January 26, 2015; revised July 2, 2015, September 18, 2015, and October 20, 2015; accepted November 22, 2015. Date of publication December 22, 2015; date of current version July 13, 2016. This work was supported by projects Contexts (S2009/TIC-1485) from CAM, Bio-Shield (TEC2012-34881) from Spanish MINECO, and BEAT (FP7-SEC-284989) from EU. This paper was recommended by Associate Editor V. Fuccella.

The authors are with the Biometric Recognition Group—ATVS, Universidad Autonoma de Madrid, 28049 Madrid, Spain (e-mail: marcos.martinez@uam.es; julian.fierrez@uam.es; javier.galbally@uam.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2015.2504101

this work. In recognition systems, graphical information is presented to the user during authentication from which the user has to perform a selection that matches a set of information previously memorized (e.g., a picture among a set of different pictures). Cued-recall systems combine the two aforementioned methods, providing graphical cues that help users recall the previously learned password (e.g., an image related to the password). Doodle-based authentication falls in the category of recall graphical passwords. A survey of graphical password authentication algorithms appears in [1].

A. Recall-Based Graphical Password Verification

A range of approaches for recall-based graphical password authentication have been evaluated using measures including resilience to forgeries, memorability, user acceptance, error rates, and time to enroll [1].

Recall-based authentication can be divided in two categories. *Exact-match* approaches assume that during authentication, a user produces exactly the same drawing provided during enrollment (e.g., [4], [8]). *Elastic* approaches allow some variability between enrollment and authentication (e.g., [6], [9]). Graphical password authentication systems can be also divided into static and dynamic approaches. *Static* or offline systems use the doodle image for authentication, while *dynamic* or online systems use time functions extracted from the doodle trajectory. Dynamic approaches have yielded better verification performance than static systems in the related field of signature verification, since more levels of information are used for authentication [2].

The Draw-A-Secret system (DAS) [4] implements a rectangular 5×5 cell grid where users trace their graphical password. The cell sequence that the users follow is stored as a password. The Background Draw-a-Secret (BDAS) [7] shows a background image behind the cell grid. A higher complexity in the password choice and better memorability were reported. With the Pass-Go authentication scheme, a variation of DAS [8], the graphical password is defined by a sequence of grid intersections instead of grid cells, overcoming the limitation of the DAS scheme, where strokes too close to adjacent cell edges could be incorrectly assigned to multiple cells.

The term "passdoodle" [5] refers to a free-form drawing. In [5], the memorability of doodles for user authentication was studied, as well as the user preference towards alphanumeric passwords or doodles. The passdoodle verification system proposed in [6] uses spatial distribution and speed for verification.

A doodle authentication system that uses DTW for matching is described in [9]. The trajectory coordinates (x, y) and their first- and second-order derivatives are used as features to characterize each doodle. Recognition performance results are provided using Tamil characters, instead of doodles. In [10], a static authentication method where free-form sketches are stored as a sequence of cell relative positions is presented. The Levenshtein distance is used to compute distances between sequences. With the Scribble-A-Secret (SAS) scheme [11], the edge orientation patterns of the doodle static image are used as features. The PassShapes approach considers graphical passwords as sequences of straight strokes following eight possible directions, at 45° angles [12].

A verification scheme based on predefined visual shapes is described in [13]. The system presents a set of cues to the users (common shapes, e.g., squares, triangles), which the users can follow to define their own free-form password. Cryptographic keys are then generated from the passwords. A graphical password verification system based on a set of predefined symbols is proposed in [14]. During enrollment, the user first selects a set of predefined symbols (at least 3) and then draws them. The set of symbols constitutes the user password.

The multitouch sketch-based authentication approach in [15] uses gestures drawn with several fingers at the same time. Since the proposed gestures are produced with all fingers, information from the hand geometry is also captured. The GEAT scheme [19] allows the user to draw a password composed of many multitouch gestures based on a set of ten predefined symbols. Support vector machines (SVM) are used for classification. In [18], an authentication scheme based on continuous touchscreen input, instead of specific gestures, is presented. SVMs and k-nearest neighbor (k-NN) classifiers are used.

Two graphical password approaches have gained popularity: the pattern lock on the android operating system and the picture password on Windows 8 devices. The pattern lock method displays a square grid of 3×3 points on the screen, and users trace a pattern connecting them. Other approaches that use dynamic information from the pattern lock drawing process have been proposed [16], [17]. In the Windows 8 picture password method, a background image is shown, and users trace on it a password composed of symbols. A summary of the proposed methods is presented in Table I.

B. Attacks to Recall-Based Graphical Passwords

Several types of attacks against graphical password authentication systems have been studied. *Smudge attacks* occur when an attacker follows the finger grease path left by the user on the screen [20]. *Shoulder-surfing* attacks occur when the attacker has visual access to the password drawing process. Several techniques against shoulder surfing attacks are proposed in [21], including adding fake strokes during the drawing process or removing strokes as they are drawn. An alternative to finger-drawn graphical passwords based on capturing the gaze trajectory has been proposed in [22] as a means to prevent shoulder-surfing.

In [23], *dictionary attacks* are studied against DAS-like systems. Users tend to select graphical passwords from a relatively small subspace of cell combinations. Thus, an attacker could be successful after a limited number of random attempts from that particular graphical subspace.

C. Signature Verification

There is a limited body of work related to doodle-based graphical passwords, in terms of systematic performance evaluation (see Table I) as opposed to handwritten signature verification [2], a particular case of graphical passwords. Behavioral information can be extracted from doodles and signatures (e.g., gesture dynamics) for matching.

TABLE I
GRAPHICAL PASSWORD AUTHENTICATION WORKS, WITH VERIFICATION PERFORMANCE

Method name	Year	Features	Distance measure	Dynamic/Static	Verification performance	Participants
DAS [4]	1999	Grid cell sequence	Exact match	Static	N/A	N/A
Passdoodle [5]	2002	Geometry & color	Visual similarity	Static	N/A	N/A
Passdoodle [6]	2004	Geometry & speed	Geometric & speed similarity	Dynamic	98.5% acceptance	10
BDAS [7]	2007	Grid cell sequence	Exact match	Static	N/A	N/A
Pass-Go [8]	2008	Grid intersection sequence	Exact match	Static	78% acceptance	167
Doodles [9]	2008	Geometry, speed, acceleration	Dynamic Time Warping	Dynamic	N/A	N/A
YAGP [10]	2008	Stroke orientations	Levenshtein distance	Static	94% acceptance	18
SAS [11]	2008	Edge orientation pattern	Correlation	Static	1% EER (random forgeries)	87
PassShapes [12]	2008	Stroke orientation	Exact match	Static	94% acceptance	17
Pseudo-signatures [13]	2008	Biometric hash	Hash matching	Static	1% EER (skilled forgeries)	37
Graphical Password [14]	2011	Predefined symbols	Exact match	Static	N/A	N/A
Multi-touch [15]	2012	Distance between points	Multiple measures	Dynamic	1.58% EER (random forgeries)	34
Password pattern [16]	2012	Coordinates, pressure, speed	Dynamic Time Warping	Dynamic	77% accuracy	31
Lock pattern [17]	2012	Timing-related features	Random forest	Dynamic	10.39% avg. EER (random forgeries)	32
Touchalytics [18]	2013	30 features	k-NN and SVM	Dynamic	3% EER (random forgeries)	41
GEAT [19]	2013	Velocity, time and acceleration	SVM	Dynamic	0.7% avg. EER (skilled forgeries)	50

Similar to doodle-based graphical passwords, two main types of signature verification approaches exist: online and offline. Online or dynamic signature verification systems use discrete-time functions sampled from the pen tip motion (e.g., x and y coordinates) to perform authentication. These signals may be captured, for example, with pen tablets or touchscreens. Dynamic signature verification systems can be further classified in two main categories. *Feature-based* or global systems, which model the signature as a holistic multidimensional vector composed of global features such as average pen speed or number of pen-ups, and *Function-based* or local systems that perform signature matching using the captured discrete-time functions (pen coordinates, pressure, etc.). Feature-based systems use statistical classifiers such as Parzen-Windows or GMMs, while function-based systems traditionally use DTW, GMMs, or hidden Markov models among other techniques. See a review in [2].

III. PROPOSED VERIFICATION ALGORITHMS

In this section, the two proposed doodle verification systems are described. First, the input coordinate sequence $[\hat{x}_n, \hat{y}_n]$ is sampled from the finger-tip trajectory on a touchscreen, as well as the time interval $\hat{t}_n$ between samples. A generic architecture of a doodle verification system is shown in Fig. 2.

A. Preprocessing and Feature Extraction

The trajectory coordinate sequence $[\hat{x}_n, \hat{y}_n]$ is resampled in order to interpolate missing samples (due to sampling errors or pauses between strokes). Cubic splines are used for interpolation. The sequences are then normalized to have zero mean, resulting in $[x_n, y_n]$.

A set of 19 additional features are extracted from the $[x_n, y_n]$ coordinate sequence (see Table II). All features are normalized to have zero mean and variance equal to 1. Thus, each doodle is described by 21 discrete-time functions.

B. System 1: Gaussian Mixture Models

GMMs have been widely used for speech and handwriting recognition. One of their main features is that they do not take into account the order of the input samples. For each user u, the distribution of d features extracted from the fingertip motion is modeled by a d-dimensional GMM λ_u. GMMs are a linear combination of N Gaussian probability density functions:

$$p\left(\mathbf{x}\,|\lambda_u\right) = \sum_{i=1}^{N} \omega_i p_i\left(\mathbf{x}\right) \tag{1}$$

where

$$p_i(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\,|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_i\right)^T \boldsymbol{\Sigma}_i^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_i\right)\right\}.$$

In our work, N is chosen to be 32, and diagonal covariance matrices $\boldsymbol{\Sigma}_i$ are used, based on the benchmark results reported in [24]. The model parameters $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \omega_i\}$ $i = 1, \ldots, N$, are estimated from a training set of doodles using the expectation maximization algorithm.

During the enrollment phase, one model is created for each user, which is later used for matching. In addition, a world GMM is created, which models the whole set of users. World models are used during the matching phase and are trained using doodles from a group of users.

The match score, given a test vector $\mathbf{x}$ and a target user statistical model λ_u, can be computed as a ratio of the log-likelihood that the test vector $\mathbf{x}$ is produced by the model λ_u and the log-likelihood that the test vector has been produced by any other user, which is modeled by the world model λ_w.

A match score s is obtained as follows:

$$s = \log p\left(\mathbf{x}\,|\lambda_u\right) - \log p\left(\mathbf{x}\,|\lambda_w\right). \tag{2}$$

C. System 2: Dynamic Time Warping

DTW was originally proposed in [25]. Our implementation was one of the best performing in the BioSecure Signature Evaluation Campaign BSEC 2009 [26].

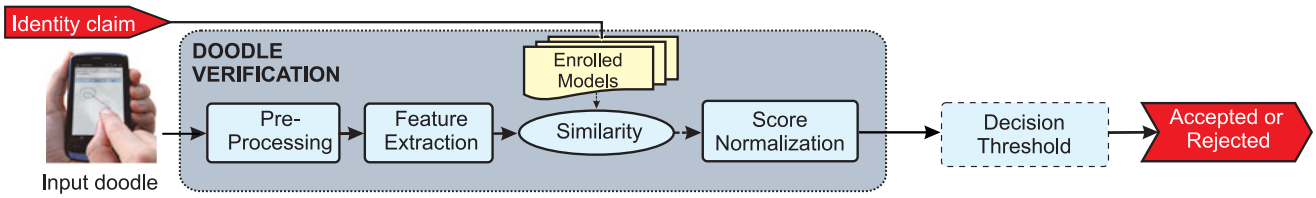

Fig. 2. Main components of a doodle verification system.

TABLE II
LOCAL FEATURE SET USED IN THIS WORK

#	Feature	Description
1	x-coordinate	x_n
2	y-coordinate	y_n
3	Path-tangent angle	$\theta_n = \arctan(\dot{y}_n / \dot{x}_n)$
4	Path velocity magnitude	$v_n = \sqrt{\dot{y}_n^2 + \dot{x}_n^2}$
5	Log curvature radius	$\rho_n = \log(1/\kappa_n) = \log(v_n / \dot{\theta}_n)$, where κ_n is the curvature of the position trajectory
6	Total acceleration magnitude	$a_n = \sqrt{t_n^2 + c_n^2} = \sqrt{\dot{v}_n^2 + v_n^2 \theta_n^2}$, where t_n and c_n are respectively the tangential and centripetal acceleration components of the pen motion.
7–12	First-order derivative of features 1–6	$\dot{x}_n, \dot{y}_n, \dot{\theta}_n, \dot{v}_n, \dot{\rho}_n, \dot{a}_n$
13, 14	Second-order derivative of features 1 and 2	$\ddot{x}_n, \ddot{y}_n$
15	Ratio of the minimum over the maximum speed over a window of five samples	$v_n^r = \min\{v_{n-4}, \ldots, v_n\} / \max\{v_{n-4}, \ldots, v_n\}$
16, 17	Angle of consecutive samples and first-order difference	$\alpha_n = \arctan(y_n - y_{n-1} / x_n - x_{n-1}) \dot{\alpha}_n$
18	Sine	$s_n = \sin(\alpha_n)$
19	Cosine	$c_n = \cos(\alpha_n)$
20	Stroke length to width ratio over a window of five samples	$r_n^5 = \frac{\sum_{k=n-4}^{k=n} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}{\max\{x_{n-4}, \ldots, x_n\} - \min\{x_{n-4}, \ldots, x_n\}}$
21	Stroke length to width ratio over a window of seven samples	$r_n^7 = \frac{\sum_{k=n-6}^{k=n} \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}{\max\{x_{n-6}, \ldots, x_n\} - \min\{x_{n-6}, \ldots, x_n\}}$

The upper dot notation (e.g., $\dot{x}_n$) indicates time derivative, and the subindexes (integers) indicate time sampling instants.

The goal of DTW is to find an elastic match among two discrete-time functions U_I and V_J of length I and J respectively. Given the two sequences, U_I and V_J, a warping path (i.e., one-to-one point correspondences) is computed so that the Euclidean distance $d(u_i, v_j)$ between corresponding samples is minimized, where u_i is the ith sample in sequence U_I and v_j the jth sample in sequence V_J. The algorithm searches for a path that minimizes the distance using a sequential procedure.

In order to limit the possible warping paths, restrictions are applied. Following a common approach in the literature, in our case, three transitions are allowed. The warping path at each point $g(i, j)$ is computed as

$$g(i,j) = \min \begin{bmatrix} g(i, j-1) + d(u_i, v_j) \\ g(i-1, j-1) + d(u_i, v_j) \\ g(i-1, j) + d(u_i, v_j) \end{bmatrix} \quad (3)$$

where $g(1,1) = d(u_1, v_1)$. The accumulated distance between the two sequences is computed as $D = g(I, J)/K$, where K is the length of the warping path. A match score is obtained as $s = \exp(-D)$.

Given a set of reference doodles provided during the enrollment phase and a test doodle, the scores between all reference doodles and the test doodle are computed, and the average is taken as the match score for that particular test sample.

IV. EXPERIMENTS

A. Database and Experimental Protocol

In the DooDB database[1] [3], the doodle dataset consists of free-form doodles, while the pseudosignature dataset is composed of simplified finger-drawn signatures. See examples in Fig. 1. The database was captured using an HTC Touch HD touchscreen mobile phone at a sampling rate of 100 Hz. Both datasets were produced by the same set of 100 users in two sessions, separated by an average of two weeks. Users held the device in their own hands while drawing. Participants were briefed to provide a graphical password that they would use as an authentication method and trained until they felt comfortable with the capture method. For each password, the $[x_n, y_n]$ coordinate sequence and the time interval between each sample are

[1] Available at: http://atvs.ii.uam.es/databases.jsp

TABLE III
FEATURE SETS SELECTED BY THE SFFS ALGORITHM ON THE DEVELOPMENT DATASETS

System	Scenario	1 x_n	2 y_n	3 θ_n	4 v_n	5 ρ_n	6 a_n	7 $\dot{x}_n$	8 $\dot{y}_n$	9 $\dot{\theta}_n$	10 v_n	11 $\dot{\rho}_n$	12 $\dot{a}_n$	13 $\ddot{x}_n$	14 $\ddot{y}_n$	15 v_n^r	16 α_n	17 $\dot{\alpha}_n$	18 s_n	19 c_n	20 r_n^5	21 r_n^7
GMM	PSEUDO-SK		✓		✓						✓	✓		✓	✓							
	DOODLE-SK		✓		✓							✓		✓	✓							
	PSEUDO-RD		✓			✓		✓		✓	✓	✓		✓	✓							
	DOODLE-RD	✓	✓			✓						✓		✓	✓							
DTW	PSEUDO-SK										✓			✓	✓							
	DOODLE-SK								✓			✓	✓		✓							
	PSEUDO-RD		✓					✓	✓	✓						✓		✓				
	DOODLE-RD		✓								✓	✓			✓	✓				✓		

captured. The time interval is in general constant, except in the transitions between consecutive strokes.

During each session, each user provided 15 genuine samples of each type (doodle and pseudosignature) and ten forgeries. To increase the quality of forgeries, the system replayed the target sample drawing process.

In the experiments, the first 50 users of the database are selected as the development set for feature selection, while the remaining are used for validation. In the development experiments, the GMM world models are estimated using the genuine samples from the validation set and *vice versa*. User enrollment is done with the first five genuine samples from Session 1. Unless stated otherwise, genuine scores are obtained with the 15 genuine doodles from Session 2, to take into account intersession variability.

Two types of forgeries are considered. *Skilled forgery* scores are obtained using the 20 available forgeries per user. *Random forgery* scores are computed for each user by comparing the user reference set (DTW system) or model (GMM system) to one sample from each of the other users. Random forgeries represent the situation where a forger claims to be a different user but provides his or her own doodle or pseudo-signature. Following this protocol, the equal error rate (EER) is used as an authentication performance measure in the experiments. $\mathrm{EER}_{\mathrm{sk}}$ and $\mathrm{EER}_{\mathrm{rd}}$ refer to the EER against skilled and random forgeries respectively. The *subjects subsets bootstrap approach* [27] is applied to estimate the EER 95% confidence intervals, performing 1000 bootstrap iterations. In each bootstrap sample, scores from 50 users are drawn with replacement from the validation set. The whole set of genuine and forgery (random and skilled) scores are drawn for each user in the bootstrap sample.

B. Experiment 1: Feature Selection

First, we analyze which are the most discriminative features for each verification system. Feature selection is carried out on the local 21-feature set using the sequential forward floating search (SFFS) algorithm [28]. The algorithm is used to find a feature set that minimizes the system EER on the development datasets.

For each dataset (doodles and pseudo-signature), feature selection is performed in two different scenarios:

1) *PSEUDO-SK & DOODLE-SK:* minimize the system EER against skilled forgeries.
2) *PSEUDO-RD & DOODLE-RD:* minimize the system EER against random forgeries.

In all cases, the 15 doodles and pseudosignatures from Session 2 are used for genuine score computation, while the first five signatures from Session 1 are used for enrollment. Thus, intersession variability is taken into account.

The best performing feature sets selected by the SFFS algorithm for each dataset and optimization scenario are shown in Table III. Feature $\ddot{y}_n$ (vertical acceleration) is present in seven of the eight sets, and features $\dot{y}_n$ (vertical speed) and $\dot{\rho}_n$ (variation of log curvature radius) are present in six of the eight sets. This indicates that vertical dynamic features may be more stable than horizontal features. However, feature $\ddot{x}$ is present in the four GMM optimal feature sets. This implies that GMMs may be more robust to users that change the usual left-to-right drawing order of their sketches (GMMs, contrary to DTW, do not consider the temporal order of time series for matching).

The performance in terms of EER against random ($\mathrm{EER}_{\mathrm{rd}}$) and skilled ($\mathrm{EER}_{\mathrm{sk}}$) forgeries of the previously computed feature sets is shown in Table IV, both on the development and validation datasets. The average of the user-specific EERs (referred to as aEER) is also reported. It is computed by averaging the individual user EERs that are obtained with user-specific decision thresholds. This represents the best EER that can be obtained if user scores were optimally normalized. The verification performance on the development and on the validation set is similar in general.

The GMM system has lower EERs against skilled forgeries than the DTW system, while the DTW system has significatively lower error rates against random forgeries. The error rates against skilled forgeries are higher for doodles, contrary to the case of random forgeries, where doodles have better performance. This may imply that pseudosignatures are harder to imitate but are more similar between them than doodles.

For the GMM system, the EER for random and skilled forgeries does not vary independently of whether the system is optimized for either of the two forgery types (i.e., the EERs and confidence intervals of the PSEUDO-SK and PSEUDO-RD scenarios are similar, and the same happens for doodles) (see Table IV). This is not the case for the DTW system and random forgeries, where the EERs vary significantly between the two

TABLE IV
VERIFICATION PERFORMANCE IN TERMS OF EER AND AVERAGE INDIVIDUAL EER (aEER) USING THE FEATURE SETS SELECTED BY THE SFFS ALGORITHM

System	Dataset	Development subset				Validation subset			
		$EER_{sk}(\%)$	$EER_{rd}(\%)$	$aEER_{sk}(\%)$	$aEER_{rd}(\%)$	$EER_{sk}(\%)$	$EER_{rd}(\%)$	$aEER_{sk}(\%)$	$aEER_{rd}(\%)$
GMM	PSEUDO-SK	**17.2** [14.5, 22.5]	12.9 [8.7, 17.6]	13.5 [10.1, 17.6]	7.6 [4.1, 9.9]	**20.9** [16.7, 24.9]	12.0 [8.8, 15.4]	14.9 [11.5, 18.3]	6.8 [4.5, 9.4]
	DOODLE-SK	**24.3** [19.4, 28.7]	9.2 [6.0, 11.4]	18.5 [14.5, 23.3]	4.9 [2.8, 5.7]	**23.0** [18.6, 26.9]	7.9 [5.1, 10.8]	17.8 [14.0, 21.6]	4.1 [2.2, 6.6]
	PSEUDO-RD	18.6 [16.1, 23.7]	**9.5** [7.0, 13.8]	14.8 [11.9, 19.6]	4.8 [2.7, 7.3]	23.1 [18.7, 27.5]	**12.9** [9.2, 16.7]	17.2 [12.9, 21.6]	6.4 [3.9, 9.5]
	DOODLE-RD	24.6 [19.7, 30.9]	**7.2** [3.9, 9.1]	20.4 [16.2, 26.3]	2.9 [1.6, 3.9]	23.7 [19.1, 27.2]	**6.7** [4.3, 9.7]	17.2 [13.5, 20.9]	3.4 [1.6, 5.5]
DTW	PSEUDO-SK	**21.6** [16.5, 26.4]	5.2 [2.0, 8.7]	15.4 [10.2, 21.1]	1.1 [0.3, 2.1]	**29.0** [24.0, 34.1]	2.7 [2.0, 3.6]	19.5 [14.6, 25.2]	0.9 [0.4, 1.4]
	DOODLE-SK	**31.9** [27.2, 36.1]	4.1 [1.4, 6.8]	24.8 [20.0, 30.0]	0.9 [0.3, 1.9]	**33.0** [28.2, 38.2]	5.2 [2.8, 6.7]	29.0 [23.3, 34.7]	1.3 [0.6, 2.1]
	PSEUDO-RD	29.1 [24.5, 34.0]	**2.0** [0.5, 6.0]	23.2 [17.9, 29.2]	0.7 [0.0, 2.2]	33.6 [28.6, 34.7]	**1.3** [0.7, 2.1]	21.0 [16.3, 26.3]	0.4 [0.2, 0.7]
	DOODLE-RD	36.7 [31.6, 41.8]	**1.6** [0.5, 3.2]	26.5 [20.6, 32.5]	0.3 [0.0, 0.6]	32.7 [26.7, 38.4]	**1.4** [0.7, 2.0]	27.3 [21.2, 33.3]	0.3 [0.0, 0.5]

Results on the development (left) and validation (right) datasets are shown. Bootstrap 95% confidence intervals are provided using the following notation: [lower bound, upper bound].

TABLE V
VERIFICATION PERFORMANCE USING SAMPLES FROM SESSION 1 BOTH FOR ENROLLMENT AND TESTING

System	Dataset	Development subset				Validation subset			
		$EER_{sk}(\%)$	$EER_{rd}(\%)$	$aEER_{sk}(\%)$	$aEER_{rd}(\%)$	$EER_{sk}(\%)$	$EER_{rd}(\%)$	$aEER_{sk}(\%)$	$aEER_{rd}(\%)$
GMM	PSEUDO-SK	**11.5** [9.0, 16.1]	7.3 [5.4, 10.4]	8.3 [5.1, 10.7]	3.3 [1.5, 4.6]	**16.2** [11.8, 19.1]	8.8 [6.0, 10.9]	11.0 [8.1, 14.1]	4.0 [2.5, 6.4]
	DOODLE-SK	**15.5** [12.1, 19.7]	5.1 [2.8, 6.7]	10.7 [9.7, 15.6]	2.1 [0.6, 2.9]	**14.4** [12.0, 18.5]	3.6 [2.8, 6.2]	10.4 [8.5, 14.3]	1.5 [0.9, 4.3]
	PSEUDO-RD	12.4 [10.6, 17.6]	**5.9** [3.2, 8.0]	8.2 [5.8, 11.6]	3.3 [1.2, 4.7]	16.4 [12.5, 19.8]	**7.5** [6.0, 11.3]	12.5 [9.6, 16.2]	3.2 [2.0, 5.5]
	DOODLE-RD	14.6 [12.8, 20.2]	**2.2** [1.0, 3.4]	11.3 [9.3, 15.3]	0.8 [0.2, 1.9]	13.5 [10.9, 16.5]	**3.3** [1.4, 5.8]	9.2 [6.4, 11.6]	1.0 [0.4, 3.2]
DTW	PSEUDO-SK	**15.2** [11.7, 19.0]	1.4 [0.4, 2.7]	8.4 [5.3, 11.9]	0.3 [0.0, 0.7]	**22.8** [18.7, 28.9]	2.2 [1.1, 3.5]	12.8 [9.1, 16.9]	1.1 [0.4, 1.9]
	DOODLE-SK	**25.2** [21.0, 29.4]	1.2 [0.5, 1.8]	15.6 [12.1, 19.3]	0.1 [0.0, 0.3]	**26.1** [22.2, 32.2]	3.3 [1.4, 4.8]	17.5 [13.9, 22.1]	1.1 [0.4, 1.9]
	PSEUDO-RD	20.2 [16.5, 24.4]	**0.6** [0.0, 1.0]	10.8 [8.0, 14.2]	0.0 [0.0, 0.1]	27.0 [22.8, 31.8]	**0.8** [0.4, 1.2]	15.3 [11.5, 19.6]	0.2 [0.0, 0.6]
	DOODLE-RD	29.3 [24.1, 33.7]	**0.4** [0.0, 1.2]	16.2 [11.8, 21.1]	0.2 [0.0, 0.5]	23.7 [18.8, 28.5]	**1.4** [0.3, 2.8]	15.5 [11.4, 19.9]	0.3 [0.0, 0.7]

The feature sets described in Table III are considered. Bootstrap 95% confidence intervals are provided using the following notation: [lower bound, upper bound].

optimization scenarios. This may reveal that for DTW-based doodle authentication, different feature sets should be used for random and skilled forgeries, respectively. That behavior is corroborated by the results of the BSEC 2009 signature verification competition, where DTW systems tuned separately for random or skilled forgeries reached top performance against each kind of forgery [26].

C. Experiment 2: Intersession Variability

Using the feature sets obtained in Experiment 1 (see Table III), we analyze the impact in the verification performance of using samples from Session 1 for authentication (instead of samples from Session 2). Consequently, user models are trained with the first five samples from Session 1, and genuine scores are computed using the ten remaining samples of Session 1. Table V shows that the EER improves in all scenarios, compared with the previous experiment (where all test samples were taken from Session 2). This reflects a high intersession variability, which may be due to the limited training period that users had while defining their own graphical password and the fact that they did not use their graphical password on a daily basis between acquisition sessions.

Comparing Table V with Table IV, the EER improvement for the GMM system is homogeneous in relative terms (around 35–45%), except in the case of doodle random forgeries. An improvement of nearly 70% in the EER against random forgeries is observed (from 7.2% to 2.2% in the development subsets). The high drop of performance in Session 2 against random forgeries corroborates that users may be failing to reproduce accurately their own doodle. Regarding the DTW system, the EER improvement against skilled forgeries is around 20–30% in relative terms, while against random forgeries, it is around 70% in most cases. This reinforces the previous observations about a high intersession variability. It is worth noting that the DTW system reaches remarkably low EERs, below 1%, and average EERs near 0%.

D. Experiment 3: Training Set Size

We investigate the effect of the number of training samples during enrollment (in Experiments 1 and 2, the systems were always trained with five samples). Keeping the previously computed optimal feature sets (see Table III), and following the same experimental protocol, the EER is obtained on each scenario using iteratively from 1 to 15 samples from Session 1 for training.

In Fig. 3, the EER evolution with respect to the number of graphical samples used for training is shown. As might be expected, the EER decreases in general when more training samples are available. However, this is not the case for the DTW system against random forgeries on both datasets. The EER does not vary when additional samples are available. In the rest of the cases, the EER starts to stabilize at six to seven training samples.

E. Experiment 4: Fusion

The verification performance combining the best systems of Experiment 1 is studied by applying score fusion. Thus, the

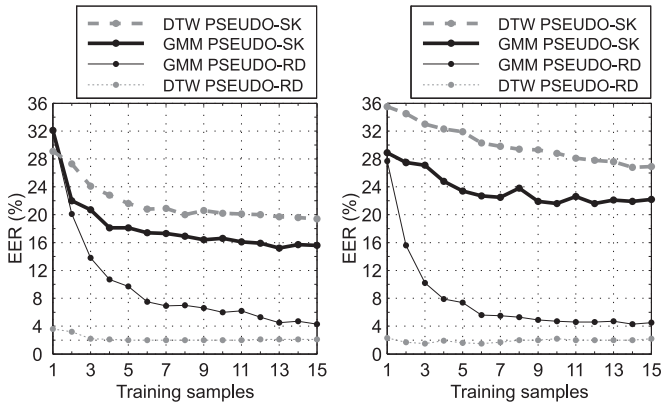

Fig. 3. Evolution of the EER in each scenario in terms of the number of training samples.

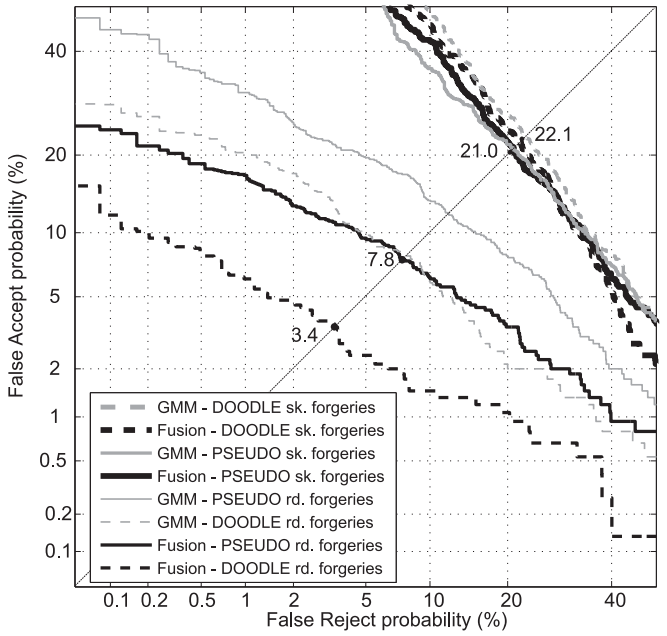

Fig. 4. Verification performance (DET graph and EERs) applying score fusion.

GMM system optimized against skilled forgeries and the DTW system optimized against random forgeries are combined, and results are computed for both datasets (doodles and pseudosignatures)

A fusion scheme based on score weighted sum is used. This approach has shown better performance over other techniques [29]. The fusion score is computed as $s = (1-k)s_D + ks_G$, where s_D and s_G are the DTW and GMM system scores, respectively, and k is the fusion weighting factor. The optimal value of k is estimated heuristically on the development dataset performing iterative experiments with different values (from 0 to 1). The optimal value is equal to 0.5 on both datasets.

The detection error tradeoff (DET) plots of both resulting fused systems on the validation datasets are shown in Fig. 4, as well as the DET plots of the GMM systems optimized against skilled forgeries (Experiment 1). The EER of the resulting systems applying score fusion is shown in Table VI.

TABLE VI
EER APPLYING SCORE FUSION

Scenario	$\text{EER}_{\text{sk}}(\%)$	$\text{EER}_{\text{rd}}(\%)$
Pseudo-signatures	21.0 [16.8, 24.6]	7.8 [5.3, 10.0]
Doodles	22.1 [17.7, 26.1]	3.4 [1.9, 4.7]

Bootstrap 95% confidence intervals are provided using the following notation: [lower bound, upper bound].

Comparing the results with the ones in Table IV, score fusion enhances the error rates against random forgeries: from 12.0 to 7.8 using doodles and 7.9 to 3.4 using pseudosignatures. The error rates against skilled forgeries do not improve significantly. These results outperform the baseline performance against skilled forgeries presented in [3] (approximately, 34% EER).

V. CONCLUSION

Two different algorithms have been analyzed for the problem of free-form graphical password verification, and the effects of feature selection, intersession variability, and training set size have been studied. Vertical features tend to be more prevalent than horizontal ones in the optimal feature sets, indicating a higher consistency.

Session intervariability negatively impacts in verification performance, as already observed in [3], probably due to users that fail to reproduce correctly their own graphical passwords. Although the GMM systems may partially overcome this issue (since they do not take into account the stroke order), verification performance is still considerably degraded. The optimal enrollment set size is around seven samples, a bit higher than the common trend in the signature verification literature (five samples) [2].

Depending on the optimization scenario (skilled or random forgeries), different optimal feature sets are selected by the SFFS algorithm. In addition, the GMM system has better performance against skilled forgeries, while the DTW system has better performance against random forgeries. This suggests that random and skilled forgeries may be a different problem from a pattern recognition point of view. This corroborates results already observed in the signature verification field, namely in the BioSecure Signature Evaluation Campaign 2009 [26], where verification systems from many international research groups were compared. The best performing systems against random and skilled forgeries were tuned for each scenario respectively, and fusion of both systems provided an overall good performance in both scenarios. In our case, score fusion also provides better results than individual systems.

REFERENCES

[1] R. Biddle, S. Chiasson, and P. Van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 19:1–19:41, 2012.

[2] J. Fierrez and J. Ortega-Garcia, "On-line signature verification," in *Handbook of Biometrics*. A. K. Jain and A. Ross, and P. Flynn, Eds. New York, NY, USA: Springer, 2008, pp. 189–209.

[3] M. Martinez-Diaz, J. Fierrez, and J. Galbally, "The DooDB graphical password database: Data analysis and benchmark results," *IEEE Access*, vol. 1, pp. 596–605, 2013.
[4] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," in *Proc. 8th USENIX Security Symp.*, 1999, p. 1.
[5] J. Goldberg, J. Hagman, and V. Sazawal, "Doodling our way to better authentication," in *Proc. Extended Abstracts Human Factors Comput. Syst.*, 2002, pp. 868–869.
[6] C. Varenhorst, "Passdoodles; a lightweight authentication method," Res. Sci. Inst., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., 2004.
[7] P. Dunphy and J. Yan, "Do background images improve "draw a secret" graphical passwords?" in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 36–47.
[8] H. Tao and C. Adams, "Pass-go: A proposal to improve the usability of graphical passwords." *Int. J. Netw. Security*, vol. 7, no. 2, pp. 273–292, 2008.
[9] N. S. Govindarajulu and S. Madhvanath, "Password management using doodles," in *Proc. 9th Intl. Conf. Multimodal Interfaces*, 2007, pp. 236–239.
[10] H. Gao, X. Guo, X. Chen, L. Wang, and X. Liu, "YAGP: Yet another graphical password strategy," in *Proc. Ann. Comput. Security Appl. Conf.*, 2008, pp. 121–129.
[11] M. Oka, K. Kato, X. Yingqing, L. Liang, and F. Wen, "Scribble-a-secret: Similarity-based password authentication using sketches," in *Proc. Int. Conf. Pattern Recog.*, 2008, pp. 1–4.
[12] R. Weiss and A. D. Luca, "PassShapes: Utilizing stroke based authentication to increase password memorability," in *Proc. 5th Nordic Conf. Human-Comput. Interaction: Building Bridges*, 2008, pp. 383–392.
[13] J. Chen, D. Lopresti, and F. Monrose, "Toward resisting forgery attacks via pseudo-signatures," in *Proc. 10th Int. Conf. Document Anal. Recog.*, 2009, pp. 51–55.
[14] W. Zada Khan, M. Y. Aalsalem, and Y. Xiang, "A graphical password based system for small mobile devices," *Int. J. Comput. Sci. Issues*, vol. 8, no. 5, pp. 145–154, 2011.
[15] N. Sae-Bae, N. Memon, K. Isbister, and K. Ahmed, "Multitouch gesture-based authentication," *IEEE Trans. Informat. Forensics Security*, vol. 9, no. 4, pp. 568–582, Apr. 2014.
[16] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you!: Implicit authentication based on touch screen patterns," in *Proc. ACM Ann. Conf. Human Factors Comput. Syst.*, 2012, pp. 987–996.
[17] J. Angulo and E. Waestlund, "Exploring touch-screen biometrics for user identification on smart phones," in *Privacy and Identity Management for Life* (ser. IFIP Advances in Information and Communication Technology), vol. 375. Berlin, Germany: Springer, 2012, pp. 130–143.
[18] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Trans. Informat. Forensics Security*, vol. 8, no. 1, pp. 136–148, Jan. 2013.
[19] M. Shahzad, A. X. Liu, and A. Samuel, "Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it," in *Proc. 19th Ann. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 39–50.
[20] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proc. 4th USENIX Conf. Offensive Technol.*, 2010, pp. 1–7.
[21] N. H. Zakaria, D. Griffiths, S. Brostoff, and J. Yan, "Shoulder surfing defence for recall-based graphical passwords," in *Proc. 7th Symp. Usable Privacy Security*, 2011, pp. 6:1–6:12.
[22] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd, "Reducing shoulder-surfing by using gaze-based password entry," in *Proc. 3rd Symp. Usable Privacy Security*, 2007, pp. 13–19.
[23] P. C. V. Oorschot and J. Thorpe, "On predictive models and user-drawn graphical passwords," *ACM Trans. Inf. Syst. Security*, vol. 10, no. 4, pp. 1–33, 2008.
[24] J. Richiardi and A. Drygajlo, "Gaussian mixture models for on-line signature verification," in *Proc. ACM SIGMM Workshop Biometric Methods Appl.*, 2003, pp. 115–122.
[25] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.
[26] N. Houmani, A. Mayoue, and S. G.-S., *et al.*, "BioSecure Signature Evaluation Campaign (BSEC'2009): Evaluating online signature algorithms depending on the quality of signatures," *Pattern Recog.*, vol. 45, no. 3, pp. 993–1003, 2012.
[27] R. M. Bolle, N. K. Ratha, and S. Pankanti, "Error analysis of pattern recognition systems—The subsets bootstrap," *Comput. Vis. Image Understanding*, vol. 93, no. 1, pp. 1–33, 2004.
[28] A. K. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997.
[29] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.