Toward Resisting Forgery Attacks via Pseudo-Signatures*

Jin Chen
Lehigh University
jic207@cse.lehigh.edu

Daniel Lopresti
Lehigh University
lopresti@cse.lehigh.edu

Fabian Monrose
UNC Chapel Hill
fabian@cs.unc.edu

Abstract

Recent work has shown that certain handwriting biometrics are susceptible to forgery attacks, both human- and machine-based. In this paper, we examine a new scheme for using handwritten input that attempts to address such concerns. Pseudo-signatures are intended to be easy for users to create and reproduce while being resilient to forgeries. Here we evaluate their feasibility in terms of usability and security through several user studies. Our initial experiments suggest that, when well-chosen, pseudo-signatures may prove to be an attractive biometric, although more research is required.

1. Introduction

As a potential biometric, handwriting offers an intuitive appeal. Biometric key generation (BKG) from a user's writing, as distinct from the task of signature authentication, uses error-corrected features to create a cryptographic key. Traditional handwritten signatures have a drawback for this application, however: each user has only one true signature. The use of arbitrary passphrases addresses this limitation, but as has been demonstrated by Ballard, *et al.*, population statistics can be used to break such systems [2].

To address these concerns, we have proposed the concept of a *pseudo-signature* [1, 3], which is outwardly similar to the "Draw-a-Secret" (DAS) graphical password scheme described by Jermyn, *et al.* [6]. In that work, the authors present users with a 5×5 grid of cells and ask them to create simple drawings. They then derive a password from the order in which the squares are visited by the stylus. They argue that the theoretical password space for DAS is much larger than the space for standard text-based passwords. In an experiment involving 16 test subjects, however, Nali and Thorpe found that 45% of their users chose symmetric passwords, yielding less security than expected [8].

Dunphy, *et al.* investigated the idea of incorporating background images to strengthen the DAS scheme [4]. Based on user studies, they found that people aided with background images tend to create more complicated passwords which exhibit fewer symmetry issues. Moreover, this also improves the memorability of graphical passwords.

In the present paper, we delve more deeply into the details of pseudo-signatures as first introduced elsewhere [1, 3]. After briefly reviewing the concept, we pose a series of questions regarding their usability and security. We then present the results of a series of experiments, drawing conclusions and highlighting areas for further research.

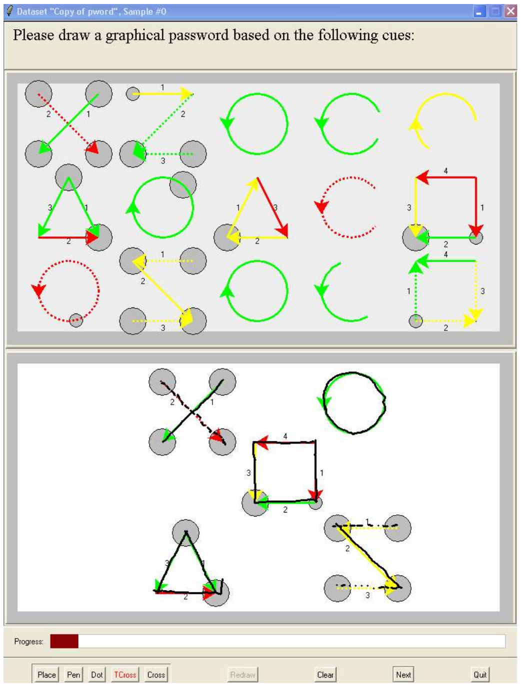
2. Pseudo-Signatures

Pseudo-signatures are sketches that a user writes solely for security purposes. Our approach differs from the original DAS scheme in several significant ways. First, we incorporate additional temporal features, such as the velocity of the pen tip and the lengths of pauses between strokes. Second, we provide users with randomly generated visual cues to help them construct better passwords. These cues include different shapes to trace, colors to suggest writing speeds, arrows to indicate directions of strokes, and locations and lengths of potential pen-tip pauses.

Figure 1 is an example of a pseudo-signature consisting of five cues. After placing her cues, a user might draw the four edges of the center square in the indicated order, with the red strokes drawn slowly, the yellow stroke somewhat faster, and the green stroke drawn quickly. She would also dwell the pen tip for a short time in the lower right corner of the square, and for a longer time in the lower left corner.

We leave it to the user's interpretation as to what constitutes "medium" velocity, or a "long" dwell time. We hope that by vaguely specifying the meanings of the visual cues, users will be able to create their own keys with sufficient *entropy* (a measure of unpredictability). Because users have the freedom to place their cues anywhere on the drawing surface, entropy should be further enhanced. Moreover, since each user is shown a different set of graphical cues based on a PIN she provides, and since these cues can be

*To be presented at the *Tenth International Conference on Document Analysis and Recognition*, July 2009, Barcelona, Spain.



Dataset "Copy of pword", Sample #0

Please draw a graphical password based on the following cues:

Figure 1. A pseudo-signature using five cues. The cue palette is displayed as a 3×5 matrix in the upper half of the display. The user places cues and writes in the lower half.

combined in arbitrary ways, the theoretical entropy available from pseudo-signatures ought to be greater than DAS and other similar schemes. In our current implementation, the cues are just a mnemonic – keys are generated solely from the handwriting – so users can, in theory, recreate their keys without using cues given enough practice.

3. Experimental Design

Our objective is to determine whether pseudo-signatures can overcome the flaws of DAS and handwritten passphrases. Several key questions come to mind:

1. Do pseudo-signatures suffer from symmetry issues?
2. Given a cue palette, do users tend to select the same set of cues and place them similarly?
3. Given a specific cue placement, do users sketch out their pseudo-signatures in the same way?
4. Can pseudo-signatures be forged by human attackers? By machine algorithms for handwriting synthesis?

A "yes" answer to any of these questions would suggest that pseudo-signatures might not serve the intended purpose.

3.1. Data Collection

Our experiment consisted of two phases: *establishment* and *forgery attempts*. We collected $2,844$ samples from 37 test subjects on a tablet PC (a NEC Versa LitePad) over a three month period (Table 1). In Round-E1, a PIN served to protect users' cue palettes. This round was repeated across five different sessions. Round-E2 and Round-E3 addressed the predictability of user-created pseudo-signatures, a critical concern.

In Round-F1, we selected five pseudo-signatures at random from Round-E1 as the target samples for forgery attempts, and then collected forgeries from 20 test subjects. Each target sample was shown first as a static image and then in a dynamic rendering. Finally, we collected forgeries from six trained forgers in Round-F2. These test subjects were first given some intuition about the features we use, then viewed playbacks of the target pseudo-signature and their earlier attempts. When finished with creating three new forgeries, the user then watched a real-time playback and selected one as her best effort.

Users generally required between 50 and 70 minutes to complete the establishment phase. For Round-F1, forgers spent between 30 and 45 minutes, while for Round-F2, they took 60 minutes on average.

3.2. Feature Selection and Biometric Hash

For key generation, we adopt the system described by Vielhauer, *et al.* [11]. The basic idea is to build a function that maps an input sample to an error-corrected key. However, since the feature selection in that earlier work has been shown to be weak [10], we borrow the approach of [2] for selecting stronger features. For each feature f_i, let r_i and a_i be the percentage that legitimate users and forgers fail to replicate, given access to the dynamic information of the target sample. Then the quality function is defined as $Q_i = (a_i - r_i + 1)/2$, and so the range of Q is $[0.0, 1.0]$. If a feature's Q-value is 0, then the feature is useless.

Having computed the quality values for 121 features [5, 7, 9, 10], we divided them into spatial and temporal categories and empirically filtered out those of low quality to yield a final set of 24 features, as described in Table 2.

In the enrollment phase, each user repeated a pseudo-signature 10 times. Let $f_{i,j}$ denote the feature value of the j^{th} feature in the i^{th} sample. When a user completes m samples, the system generates a biometric template as follows. Let $l_j^{'} = minf_{i,j}$, $r_j^{'} = maxf_{i,j}$, and $\Delta I_j = maxf_{i,j} - minf_{i,j} + 1$. Set $l_j = l_j^{'} - \Delta I_j \times \epsilon_j$, and $r_j = r_j^{'} + \Delta I_j \times \epsilon_j$, where ϵ is the tolerance value for the corresponding feature prespecified in a *tolerance table* ($T = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_n\}$) [11]. The biometric template is then an $n \times 2$ matrix of integer values

Table 1. Specification of the data collection.

Phase	Name	Experiment Description	Samples
Establishment	Round-E1	Each user creates one pseudo-signature and repeats it 10 times. She enters a PIN, selects and places her cues, and then draws the pseudo-signature.	$1,870$
	Round-E2	Each user creates 10 pseudo-signatures based on 10 different pre-defined cue palettes. She selects and places cues, and then draws the pseudo-signature.	350
	Round-E3	Each user creates one pseudo-signature based on a fixed set of cues that are already placed. She only draws the pseudo-signature.	350
Forgery Attempts	Round-F1	Each user attempts to forge five different pseudo-signatures that are shown as static images and dynamic renderings.	190
	Round-F2	Six selected users are provided with training and asked to forge the same five pseudo-signatures as in Round-F1, repeating each three times.	84

$[(l_1, r_1), (l_2, r_2), \ldots, (l_n, r_n)]$.

To hash a new feature vector so that we can compare it to the reference template, we also need some auxiliary information. Let $\Omega_j = l_j mod \Delta I_j$ denote the offset of the hashed j^{th} feature value. In this way, when a legitimate user wants to recreate a key, the system extracts the features from her querying sample and computes the hash as $H_j = (f_{i,j} - \Omega_j)/\Delta I_j$, where i is the index of the input sample and $j = 1, 2, \ldots, n$. This scheme divides the feature space into intervals along each dimension and thus is able to map two inputs with minor differences into the same output: a biometric key.

To compare a reference sample to a query sample, we first generate a template from the reference sample, then use this template and the query to generate a key, and finally compute the total number of bit differences that need to be corrected in the key. The ROC figures we plot in this paper are based on this scheme.

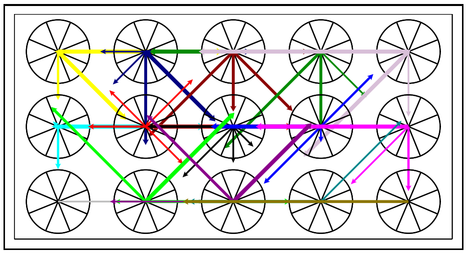


Figure 2. User adjustment of cue placement.

Table 3. Average positional shifts for cues from their initial placements. (ΔX, ΔY) denotes the horizontal and vertical shifts.

	Col 1	Col 2	Col 3	Col 4	Col 5
Row 1	(120,53)	(89,61)	(7,53)	(-158,53)	(-226,64)
Row 2	(140,-16)	(85,-7)	(-25,-22)	(-112,-21)	(-165, 4)
Row 3	(92,-81)	(85,-118)	(-13,-110)	(-95,-107)	(-157,-97)

4. Experimental Evaluation

We now consider the questions enumerated at the start of Section 3. For Question 1, we counted symmetric pseudo-signatures from Round-E1 and Round-E2 for 34 subjects, and found that 58 out of 374 samples exhibited left-right symmetry (other symmetries such as top-bottom or rotational were much less common). This percentage is 15.5%. Although hard to compare to Nali and Thorpe's findings [8], we consider this ratio to be relatively low, which is good.

As to Question 2, we found that 35 pairs of pseudo-signatures used the same set of cues. Since 35 users participated in the experiment, there are $35 \times (35 - 1)/2 = 595$ pairs of pseudo-signatures. a percentage of 5.9%. If we assume that on average three distinct cues are selected from a 3×5 palette, then in theory the expected number of distinct cue sets will be $\binom{3 \times 5}{3} = 455$. Hence, the probability of selecting the same set of cues is $1/455 = 0.22\%$. Although the actual percentage is higher than this, we note that while the cues may be the same, their placements differ by 56 pixels on average, a substantial distance in our GUI. Indeed, for increased security, users should be encouraged to move their cues from the initial placements. On the other hand, such a policy may adversely impact memorability. A study of this tradeoff is left as future work.

To examine tendencies when users move their cues, we measured the average positional shifts; these are plotted in Figure 2. Here, arrows encode both the direction of the move and the average distance from the default positions. The width of the arrow represents the number of cues moved in that direction. Table 3 also lists the average horizontal and vertical shifts. Users seem to prefer to move their cues toward the center of the writing pad, behavior that could be discouraged through feedback from the GUI.

We investigated Question 3 and Question 4 in the forgery

Table 2. Feature set used in the study.

Feature(f)	$Q(f)$	**Feature(f)**	$Q(f)$	**Feature(f)**	$Q(f)$	**Feature(f)**	$Q(f)$
Spatial Features							
# of strokes [10]	0.72	$IntegralArea(x_1)$ [10]	0.71	$\sigma(x)$ [7]	0.70	signature width [5]	0.70
$x_{end} - x_{min}$ [7]	0.69	L/A [7]	0.68	$IntegralArea(x_2)$ [10]	0.67	$Invar.Matrix(1,1)$ [5]	0.67
$PixelMatrix(3,4)$ [10]	0.66	$AveDist_{all}$ [10]	0.66	aspect ratio [10]	0.64	$AveX$ [10]	0.61
$IntegralArea(y_1)$ [10]	0.61	$IntegralArea(x_3)$ [10]	0.61	$PixelMatrix(1,1)$ [10]	0.60	$PixelMatrix(2,1)$ [10]	0.60
Temporal Features							
T(2nd pen-down) [7]	0.73	$\overrightarrow{AveV_x}$ [10]	0.70	$T(MinV_x)$ [10]	0.68	T [10]	0.68
$\overrightarrow{AveV_y}$ [10]	0.68	path length/T [10]	0.67	θ(PD(1),PU(2)) [7]	0.62	$MaxV_y - AveV_y$ [7]	0.60

experiments. We employ five forgery models from Ballard, *et al.* [2]: *naïve*, *naïve**, *static*, *dynamic*, and *trained*. To compute FRR (*False Reject Rate*) and FAR (*False Accept Rate*), we adopt the feature set we described in Section 3.2.

In the context of pseudo-signatures, *naïve* forgeries are not effective forgery attempts since our measure compares the key from a user's randomly selected pseudo-signature with the target. *Naïve** forgeries (Round-E3) share the same cue palette with the target pseudo-signature. We calculate FRR by partitioning a subject's m enrollment samples from Round-E1 into a *training* set ($3m/5$) and a *testing* set ($2m/5$), and then measure the reject rates. As to FAR, we measure the acceptance rates for forgeries based on the same training data. From our analysis, the EER (*Equal Error Rate*) for the *naïve* forgery model is 0% and for the *naïve** model it is about 1%.

To simulate a "shoulder-surfing" attack, we employed the offline, online, and trained forgery models. As can be seen in Figure 3, forgers are not likely to replicate the target sample if they only have access to the static image; the EER is approximately 1%. Given a dynamic rendering, forgers did better, which is not a surprise. However, we note that their overall success rate is still quite small; the EER increases to 3% in this particular case. For comparison purposes, we note that the EER for trained forgers in our earlier work on handwritten passphrases was 20.6% [2].

The "FAR Offline" and "FAR Online" curves in Figure 3 are plotted across the entire forger set (size = 20). This provides a snapshot of the original pseudo-signature's security. It is also important to know if there are any particularly good forgers in this group. This point may not be reflected in the ROC curves if the number of good forgers is too small to impact the averages. In plotting the distribution of each forgery sample based on the number of bit differences in the biometric key it generates, we find that all are located in the interval $[7, 19]$ (for trained forgeries, the interval is $[8, 17]$), meaning that no forgeries are particularly close to their targets. This suggests that pseudo-signatures may be resilient to forgeries, and that the average EERs we report reflect actual security levels. More testing is needed.

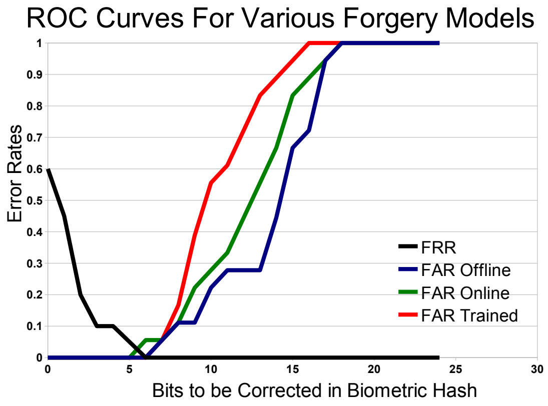


Figure 3. ROC curves for human forgeries.

Table 4. EERs for each pseudo-signature.

Attack Model	Pseudo-Signature Sample				
	#1	#2	#3	#4	#5
Offline	4%	8%	18%	1%	1%
Online	10%	8%	18%	4%	5%
Trained	18%	22%	40%	1%	5%

Computer security is typically based on a worst-case analysis, since a weak password for just one user can lead to the compromise of the entire system. We note that a few pseudo-signatures in our test set have high EERs (Table 4). One example is Pseudo-Signature $\#3$, which consists of three cues and only four strokes. Moreover, two out of the three cues the user chose are interpreted as dashed curves, which are more difficult to reproduce reliably than solid lines. Hence, the intervals in the template for this particular test subject are unusually loose, making it easier for forgers to succeed in their attacks.

Pseudo-Signatures $\#1$, $\#2$, and $\#5$ also consist of three cues, but employ more strokes. This contributes to the somewhat higher EERs for the first two. In the case of Pseudo-Signature $\#5$, which achieves a much better EER, we believe this is because the user employed a range of writing speeds, including distinct low, medium, and high velocity segments. This creates a challenge even for trained forg-

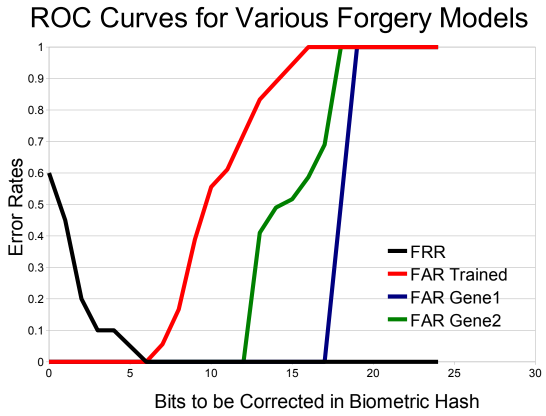


Figure 4. ROC curves for machine forgeries. Gen1 **assumes the attacker knows the cue palette, while** Gen2 **is a worst-case analysis.**

ers who are not able to reproduce the original sample even though its composition is fairly simple. This observation was confirmed by a post-study survey of the forgers.

As we know from past work [2], another significant threat exists: machine-based attacks that employ generative algorithms for handwriting synthesis (Question 4). In Figure 4, we present two preliminary attempts at testing such attacks and compare them with the trained human forgers. Gen1 is a shoulder-surfing scenario where we assume the attacker knows the cue palette but nothing else. Gen2 serves as a worst-case analysis because in addition to the cue placement, it assumes that the cue order, stroke order, and stroke directions are known in advance. It then attempts to guess the target's *speed profile* (slow, medium, fast) based on population statistics. At this early stage, the generative attacks are even less effective than the trained forgers.

5. Conclusions

The work we have presented here builds on our earlier proposals for pseudo-signatures [1, 3]. In this paper, we evaluated their feasibility from the standpoint of usability and security. Research is still ongoing, but it appears that when well-chosen, pseudo-signatures may prove to be an attractive biometric. In the near future, we plan to examine in more detail the vulnerabilities of pseudo-signatures to machine-based attacks. We are also contemplating the design of a feedback mechanism in our GUI which will evaluate the quality of pseudo-signatures during the enrollment phase and encourage users, when necessary, to create stronger examples for their security applications.

6. Acknowledgments

The authors wish to thank Lucas Ballard for his valuable contributions as well as the anonymous test subjects who participated in oue data collection. This work was supported by NSF under grants CNS-0430178 (Lehigh) and CNS-0430338 (Johns Hopkins). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

References

[1] L. Ballard, J. Chen, D. Lopresti, and F. Monrose. Biometric key generation using pseudo-signatures. In *Proceedings of The 11th International Conference on Frontiers in Handwriting Recognition*, Montreal, Canada, August 2008.

[2] L. Ballard, D. Lopresti, and F. Monrose. Forgery quality and its implications for biometric security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Special Issue)*, 37(5):1107–1118, October 2007.

[3] J. Chen, D. Lopresti, L. Ballard, and F. Monrose. Pseudo-signature as a biometric. *In the Proceedings of IEEE 2nd International Conference on Biometrics Theory, Applications and Systems*, 2008. Arlington, VA, USA.

[4] P. Dunphy and J. Yan. Do background images improve “draw a secret” graphical password? In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 36–47, Alexandria, VA, USA, October 2007.

[5] R. Guest. The repeatability of signatures. In *Proceedings of 9th International Workshop of Frontiers Handwriting Recognition*, pages 492–497, 2004.

[6] I. Jermyn, A. Mayer, F. Monrose, M. Reiter, and A. Rubin. The design and analysis of graphical passwords. In *Proceedings of the Eighth USENIX Security Symposium*, August 1999.

[7] L. Lee, T. Berger, and E. Aviczer. Reliable on-line human signature verification systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):643–647, 1996.

[8] D. Nali and J. Thorpe. Analyzing user choice in graphical passwords. Technical report, School of Information Technology and Engineering, University of Ottawa, May 27 2004.

[9] W. Nelson and E. Kishon. Use of dynamic features for signature verification. In *Proceedings of IEEE International Conference on System, Man, and Cybernetics*, pages 1504–1510, 1991.

[10] C. Vielhauer and R. Steinmetz. Handwriting: Feature correlation analysis for biometric hashes. *EURASIP Journal on Applied Signal Processing*, 4:542–558, 2004.

[11] C. Vielhauer, R. Steinmetz, and A. Mayerhofer. Biometric hash based on statistical features of online signatures. In *Proceedings of the Sixteenth International Conference on Pattern Recognition*, volume 1, pages 123–126, 2002.