Evaluating the Security of Handwriting Biometrics

Lucas Ballard, Daniel Lopresti, Fabian Monrose

Evaluating the Security of Handwriting Biometrics

Lucas Ballard
Johns Hopkins University
Baltimore, MD, USA
lucas@cs.jhu.edu

Daniel Lopresti
Lehigh University
Bethlehem, PA, USA
lopresti@cse.lehigh.edu

Fabian Monrose
Johns Hopkins University
Baltimore, MD, USA
fabian@cs.jhu.edu

Abstract

Ongoing interest in biometric security has resulted in much work on systems that exploit the individuality of human behavior. In this paper, we study the use of handwritten passphrases in the context of authentication or cryptographic key generation. We demonstrate that accurate generative models for a targeted user's handwriting can be developed based only on captured static (offline) samples combined with pen-stroke dynamics learned from general population statistics. Our work suggests that such automated attacks are nearly as effective as skilled human forgers and hence deserve serious consideration when evaluating the security of systems that use handwriting as a biometric.

Keywords: biometric security, online handwriting, performance evaluation, forgery threat models.

1. Introduction

Ongoing interest in biometric security and related topics has resulted in much work on systems that exploit the individuality of human behavior. Signature verification, for example, has had a long, rich history, with hundreds of papers written on the subject (see, e.g., [9, 16]).

The use of signatures has some well known advantages: they are a natural and familiar way of confirming identity, have already achieved acceptance for legal purposes, and their capture is less invasive than most other biometric schemes [5]. Still, each individual has only one true signature — a severe limitation when it comes to certain security applications. As a result, researchers have recently begun to examine using arbitrary handwritten phrases, recasting the problem as one of computing cryptographic keys or biometric hashes (*e.g.*, [6, 10, 19, 20]).

In our work, we adopt such a paradigm to support the study of threat models we believe have received insufficient attention in the literature. While some of the attacks we envision are more realistic than others, all are designed to "stress" handwriting biometrics in a variety of new ways, with our ultimate goal being to build confidence in the inherent security of a given measure (or, conversely, to demonstrate that the measure suffers from flaws that should be addressed before it is put into practice). Our earlier research ([1, 13, 14]) shows that such assessments can shed new light on the effectiveness of certain biometrics, particularly when considered as an authentication mechanism [17], or for cryptographic key generation [6, 10, 19, 20].

In addition to studying the abilities of human forgers when presented with varying degrees of knowledge regarding the targeted writing, we ask whether an adversary can be successful in developing an accurate *generative model* for the user in question when given only limited information. In particular, as we shall demonstrate, such a model, trained only on captured static (offline) samples combined with pen-stroke dynamics learned from general population statistics, is nearly as effective as the best human forgers we have encountered, and substantially better than our "average" forger. This surprising result suggests that automated attacks should receive serious attention in the evaluation of security systems that use handwriting as a biometric.

2 Evaluating Handwriting Biometrics

In this section, we begin by describing the prior work that is most closely related to our own. We then provide a broad overview of our approach to data collection, followed by the evaluation of a specific set of handwriting biometrics under more stringent adversarial conditions than are typically assumed in the biometric literature.

2.1 Related Work

Particularly relevant to the research we are pursuing are a series of recent papers that use online handwriting for the computation of cryptographic keys. Feng and Wah, for example, describe a scheme for generating keys from handwritten signatures using an initial filtering based on dynamic time warping followed by the extraction of 43 features yielding an average key length of 40 bits [6]. The authors claim an Equal Error Rate (EER) of 8%, and mention that their test database contains forgeries, but unfortunately provide no details on how these were produced or their quality.

Kuan, *et al.* present a method to generate cryptographic keys from online signatures [10]. Their approach was evaluated on the SVC dataset [22] and achieved EERs of between 6% and 14% given access to a stolen token.

Vielhauer, *et al.* present a biometric hash based on 24

integer-valued features extracted from an online handwriting signal [20]. Fourteen of the features are global, while the remaining ten features are derived from segmented portions of the input obtained by partitioning the bounding box surrounding the ink into five equal-sized regions in the x- and y-dimensions. Forgeries based on an offline image of the target signature were collected. The authors report achieving a False Accept Rate (FAR) of 0% at a False Reject Rate (FRR) of 7% in their studies, but only 10 subjects were used in the testing. A later paper discusses feature correlation and stability for a much larger number of features; however, the same number of test subjects is used [19]. This follow-on work differs in that the new system is more robust to natural variation, but the authors do not report a FAR.

As can be seen, performance figures (*i.e.*, EER) are difficult to compare directly as the sample sizes are often small and test conditions quite dissimilar [4]. Furthermore, even when forgers are employed in such experiments, there is usually no indication of their proficiency.

We note that the production of "skilled" forgeries for the SVC dataset [22] resembles the methodology we have used in our own studies, although the definition of "skilled" in that work is closer to "knowledgeable" than it is to "talented." For the competition, a database incorporating 100 sets of signatures was created, with 20 genuine signatures and 20 forgeries penned by at least four subjects comprising each set. In the case of attempted forgeries, users could replay the dynamic sequence of the targeted handwriting on the screen of a software viewer before attempting to forge it. However, there was no attempt to distinguish effective vs. ineffective forgers (or hard-to-forge vs. easy-to-forge writers), or to measure the performance of forgeries based only on static (offline) data.

The first serious attempt we are aware of to provide a tool for training forgers to explore the limits of their abilities is the work by Zöbisch and Vielhauer [21]. In a small preliminary study involving four users, they found that showing an image of the target signature increased false accepts, and showing a dynamic replay doubled the susceptibility to forgeries yet again. However, since the verification algorithm used was simplistic and they do not report false reject rates, it is difficult to draw more general conclusions.

Lopresti and Raim reported the results of a simple concatenative-style attack in a small-scale study involving two test subjects writing four passwords 20 or more times each [13]. A parallel corpus of unrelated writing was also collected and labeled at the unigram level. This was then used to attempt to break the original biometric hash proposed by Vielhauer, *et al.* [20]. After a feature-space search of at most one minute on a Pentium-class PC, the attack was found to be successful 49% of the time.

More recently, the authors of the present paper conducted a detailed analysis which examined the skill levels of human forgers along with their ability to improve via training [1]. This earlier work also included a more comprehensive study involving a particular kind of concatenative attack based on the rather strong assumption that the adversary has access to online samples of the targeted user's handwriting. At that time, we found that our most successful forgers were able to achieve significantly higher success rates than the average "random" test subject and, furthermore, they were able to increase their chances of breaking the biometric hash with a reasonable amount of practice. We also noted that the generative attack was even more successful than the human forgers. Our conclusion, then, was that such worst-case scenarios deserve much more attention than they have been receiving in the evaluation of biometric systems.

The current paper builds on our earlier work in [1] by relaxing a key assumption: here we only allow the attacker access to offline samples of the user's handwriting, as might be obtained through discarded correspondence or scraps of paper, for example. The temporal characteristics needed to recreate the writing accurately enough to defeat the biometric system are computed from general population statistics.

Other researchers have also attempted to infer dynamics from a static image of handwriting, or to synthesize realistic-looking writing using a model. The former problem arises in offline instances of handwriting recognition and signature verification, where one promising approach has been to try to extract both offline and online features from a scanned image of the writing (*e.g.*, [2, 3, 11, 12]), although largely this work has focused on recovering stroke-order and not velocities or accelerations.

From the broadest perspective, our research relates to the topic of generative models for handwriting, which has received much attention over the years (*e.g.*, [7, 15]). This work is often oriented toward automating the production of training data to improve recognition algorithms, but is clearly relevant to the attack scenarios we have in mind. Particularly intriguing is recent research that combines the development of generative models with techniques for learning the dynamics of handwriting (*e.g.*, [8, 18]). Our near-term plans are to study the threat posed by such models in the context of biometric security.

2.2 Experimental Setup

We are interested in comparing and contrasting two fundamentally different scenarios. In the first, we wish to understand how well a determined and talented human forger can perform when working from increasing degrees of knowledge about the targeted user's handwriting. In the second, we want to quantify the security of handwritten passphrases in the face of automated attacks using generative models trained on offline samples of the user's writing (captured, perhaps, from pages of notes the user has discarded) combined with pen-stroke dynamics learned from general population statistics.

Our data collection efforts are supported by several graphical tools we have developed in the Tcl/Tk language. The handwriting of test subjects is captured using one version (*icapture*) that is driven by scripts describing a

given experiment, in most cases specialized to the particular user. Another, more sophisticated tool (*iedit*) supports browsing and editing ink files as well as annotation of pen-strokes (which are stored as point sequences) at the n-gram level. Our tools are portable and run under both Linux and MS Windows operating systems.

To study the forging "talents" of our subjects, *icapture* supports collecting forgery attempts under a variety of circumstances. In addition to traditional naïve forgeries, in some tests we show the user a static image of the targeted writing, or a dynamic real-time replay of the writing as it was originally captured. The former scenario corresponds to an adversary stealing the user's passphrase as written on, say, a piece of paper, while the latter represents a "shoulder-surfing" type of attack (the adversary watching the user write her passphrase surreptitiously).

Users were allowed an unlimited number of attempts to forge the writing before moving on to the next sample. They were also allowed to view an unlimited number of replays in the case of the online writing tests. A substantial amount of effort was spent ensuring that the replays seen by test subjects were high-fidelity, matching the timing of the original writing. We are particularly interested in identifying skilled forgers (sometimes called "wolves" in computer security contexts), as well as hard-to-forge and easy-to-forge writers (the latter are often known as "sheep") [1]. Doing so allows us to explore more completely the space of potential threats to biometric security as well as potential ways of addressing the flaws that might be uncovered.

In addition to sample passphrases and human-generated forgery attempts, we also collected a parallel corpus of unrelated writing from each test subject. This data is employed in a variety of potential *generative* attacks, modeling, for example, the scenario where a targeted user's PDA or pen computer is captured by an adversary, or offline samples of the user's writing are scanned or traced off of hardcopy. We designed this dataset to provide complete coverage of the passphrases at the unigram and bigram levels, but, of course, the passphrases themselves appear nowhere in this corpus.

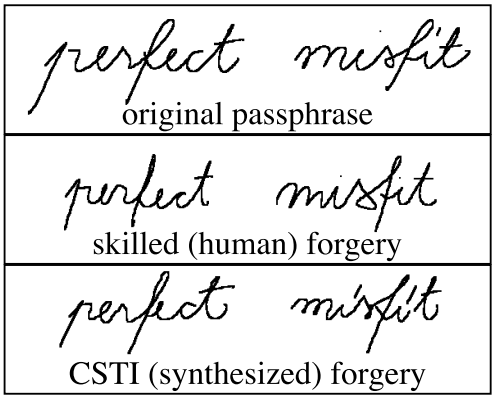

Figure 1. Example forgeries created by a skilled human forger and our algorithm.

2.3 Synthesizing Handwriting

We group users into one of three categories based on the predominate style of their handwriting: cursive, block, or mixed. To generate forgeries, we use an algorithm that we call *Concatenative-Synthesis with Temporal-Inference* (CSTI). At a high level, the algorithm proceeds as follows. When attempting to forge a target writer with passphrase p, the adversary first computes some general statistics over the (annotated) parallel corpora of writers with the same writing style. Such statistics include inter-character spacing and timing, as well as stroke velocity.

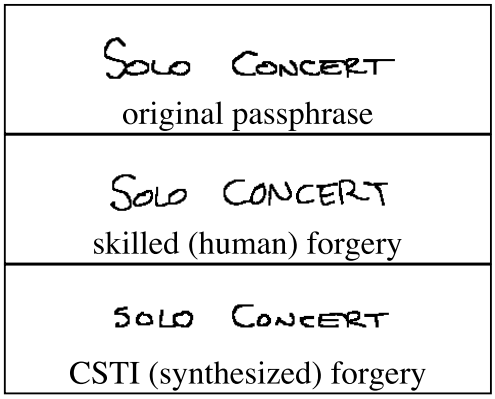

Figure 2. Example forgeries created by a skilled human forger and our algorithm.

The adversary then traces the phrases in the target user's parallel corpus to recover the overall shape of the writing, as well as a "best-guess" estimate of stroke direction and order. The tracer is provided with an offline image of the writing and allowed unlimited attempts to replicate the strokes. However, there is no attempt to reproduce the dynamics of the writing; that information will be synthesized using a procedure to be described shortly.

A search of the (traced) parallel corpus is then performed to extract n-grams that comprise the passphrase p and combine the $x(t)$ and $y(t)$ signals of each n-gram to form a static rendering of p (see [1] for a more detailed discussion of this process). Finally, using the temporal-inference algorithm discussed below, the elapsed time between each point in the synthesized passphrase is estimated, reconstructing the dynamics. Figures 1 and 2 show examples of a forgery generated in this manner.

2.4 Inferring Velocity

To aid in the generation of forgeries, we first attempt to derive a set of measures that can be used to infer the pen-tip velocities for a user's writing. The hope is that these measures can be used to compute a model at the character level that also reflects the subtleties of the underlying strokes. To do so, we assume access to the online corpus of handwriting for users with the same writing style as the target. For each of the annotated samples we apply linear re-sampling to ensure the Euclidean distance between

each point within a stroke is separated by d units.[1]

Next, we examine each stroke within each instance of a given letter in the corpus. For example, suppose that we are examining the t^{th} stroke of character c, which itself was rendered using T strokes in total. Let us denote this stroke as S. S consists of a series of points $p_1, \ldots, p_n$. We now apply a sliding window, w, of length ℓ along the stroke to acquire a set of points, with p_i being the first point in w. For an arbitrary collection of adjacent points, let $\Delta(\cdot)$ be the sum of the Euclidean distance between consecutive points in the collection, and likewise, $\delta(\cdot)$ the Euclidean distance between the first and last point.

For pedagogical purposes, assume we are examining the j^{th} such window, w_j. From the points in w_j we compute four functions to aid in generating velocity profiles, namely straightness (σ), the offset of the window within the stroke (o), direction (θ) and the number of extrema within the window (e). In particular, let: $\sigma(w_j) = \delta(w_j)/\Delta(w_j)$, $o(w_j) = \Delta(\{p_1, \ldots, p_i\})/\Delta(S)$, $\theta(w_j) = \operatorname{atan}\left(\frac{y_{i+\ell} - y_i}{x_{i+\ell} - x_i}\right)$, and $e(w_j)$ be the number of local extrema in both vertical and horizontal directions. For a given c, j, t, T tuple, let $\gamma_j^{c,t,T} = \langle \sigma(w_j), o(w_j), \theta(w_j), e(w_j) \rangle$. Next, we take all instances of the stroke under consideration to create the set $G^{c,t,T} = \bigcup_j \{\gamma_j^{c,t,T}\}$. To partition this set into groupings with similar characteristics, we apply k-means to find representative clusters ($\{G_1^{c,t,T}, \ldots, G_K^{c,t,T}\}$) with accompanying velocities $\{v_1^{c,t,T}, \ldots, v_K^{c,t,T}\}$. To infer pen-up time, we apply a similar concept except that γ is now only a function of the distance between pen-up and pen-down points. This process is applied to all characters in the corpus.

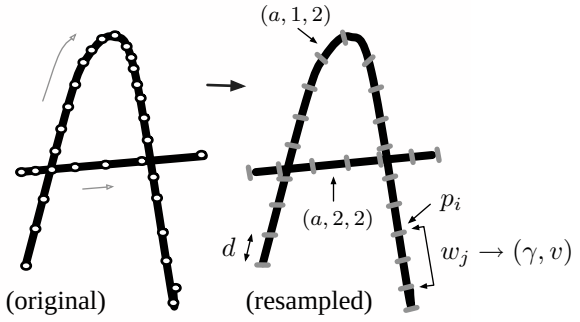

Figure 3. Applying a per-stroke sliding window to generate velocity profiles.

2.5 Using Velocity Profiles in Forgeries

After using Concatenative-Synthesis [1, 13] to create a static rendering of a forgery, the adversary uses the recorded velocity profiles to infer the elapsed time between each point in the forgery. Each stroke of each character in the static rendering of the target passphrase p is processed in this fashion. For the t^{th} stroke in letter c – again, rendered using T strokes in total – the adversary slides a window (w') of length ℓ over the stroke and computes $\gamma' = \langle \sigma(w'), o(w'), \theta(w'), e(w') \rangle$. Using a k nearest-neighbors approach, she then determines which cluster $G_i^{c,t,T}$ the vector γ' is most similar to. Once found, the velocity $v_i^{c,t,T}$ is used as an estimate of the pen speed for the point at the center of w'.

[1] The one exception is that we preserve the end points of each stroke.

3 Data Collection

Our results are based on 11,038 handwriting samples collected over several weeks from 50 users at our two universities. We used NEC VersaLite Pad and HP Compaq TC1100 pen computers as our writing platforms. To motivate our participants, several incentives were awarded during each round of data collection.[2]

First, users were asked to write five passphrases, ten times each. These passphrases were two-word sayings that are easy to recall ("perfect misfit," "solo concert," "crisis management," "least favorite," and "graphic language"). During that same session a second (disjoint) set of 65 phrases was also collected for our parallel corpus. Approximately two weeks later, participants re-wrote the original five passphrases ten times each. Next, users were asked to forge representative samples (selected based on writing style, gender, and handedness) collected in the earlier round. The first challenge was to produce forgeries after seeing only an offline representation of the writing; later the test was repeated, using real-time renderings of the target passphrases.

One week later, we singled out nine "skilled" (but untrained) forgers; three forgers for each of the three writing styles. We explained the types of features that are generally used in online handwriting systems. These forgers were then asked to forge 15 writing samples, with 60% of the samples coming from the weakest ten targets, and the other 40% chosen at random.

A point worth emphasizing is that our "skilled" forgers exhibited a high degree of self-motivation. In one instance, a forger wrote a *single* passphrase 106 times before being satisfied that what he had created depicted an accurate reproduction (the forgers were not provided with any direct feedback on the effectiveness of their attempts). Another replayed the target phrase 40 times. On average, each forger made 14 attempts and redrew the target five times, taking roughly 1-2 hours to forge the 15 passphrases. Clearly, such dedication plays an important role when evaluating the security of biometric systems, but this point has been explored only to a limited degree in the literature to date. One can expect determined adversaries to expend at least as much effort as the forgers in our study.

Finally, we had four test subjects trace the parallel corpora to recover the overall shape, stroke-order, and direction of the writing. These users were never asked to trace their own handwriting. Velocity information was discarded, to be re-synthesized as described earlier.

[2] *E.g.*, snacks, gift certificates, and special prizes for most consistent writers, best forgers, etc.

4 Experimental Results

To study the effectiveness of the aforementioned approach, we empirically evaluate the technique in the context of a biometric handwriting authentication system. In particular, we adapt the biometric hash of Vielhauer, *et al.* [19, 20] and evaluate the accuracy of the forgeries generated by Concatenative-Synthesis in which temporal information is inferred. To establish a baseline for comparison, we also report the accuracy of the forgeries produced by adversaries with varying levels of knowledge [1].

To ensure that our results do indeed reflect a meaningful assessment, we devoted some effort to studying the robustness of the underlying authentication system. In addition to testing the system as originally proposed in [19] we also examined several enhancements. First, we performed an empirical analysis of a wide range of features to find those that appear to be the most secure [1]. Specifically, we analyzed 145 state-of-the-art features and measured the security of each by noting the difference between the proportion of times legitimate users and forgers fail to reproduce a given feature. We then chose the features with the smallest such difference and used the resulting 36 online and offline features to drive the system. Second, we use smaller tolerance values than those described by Vielhauer *et al.* [19]. The approach by Vielhauer, *et al.* assumes that each value in the biometric hash must be recovered perfectly. We, however, assume that it will be possible to correct errors in a small number of hash positions, either through a search process or error-correcting code. For example, depending on the range of values in the positions in questions, correcting 5-10 errors is feasible in a few seconds of computation time on modern PC's. The number of positions in which errors have been corrected provides the x-axis in our performance graphs.

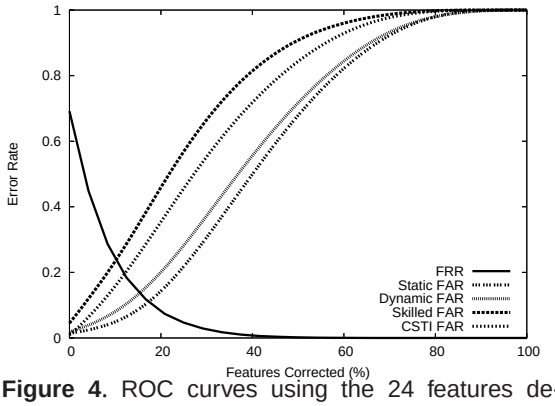

Figure 4. ROC curves using the 24 features described by Vielhauer *et al.* [20].

We used the 20 renderings of the five passphrases from the first two rounds of data collection (see Section 3) to generate a biometric template (*i.e.*, the "interval matrix" in [19]) for each user and passphrase. After omitting data from users who failed to enroll, we computed the FRR by generating templates using 75% of a user's writing, and attempted to authenticate the remaining 25%. The FAR was computed using templates generated from the writings of other users for the passphrase in question. The reported results are averaged values across 25 random partitions of the data.

In the subsequent discussion we report results based on four distinct FARs: the `Static` and `Dynamic` cases are computed using forgeries from a group of non-trained forgers given access to offline and online renderings of the target passphrase, respectively. The `Skilled` case reflects forgeries from our trained forgers. Lastly, we report the FAR for the concatenative-synthesis approach with inferred temporal information.

Figure 4 gives the results when using the features described by Vielhauer, *et al.* [20]. The `Static` and `Dynamic` forgeries exhibit an EER of 8.6% and 12.8% at 5 and 4 errors corrected. Not surprisingly, the impact of `Skilled` forgeries is dramatic, and results in an EER of 23.2% at 2 errors corrected. Finally, the EER for forgeries generated using concatenative-synthesis with temporal-inference is 18.6% at 3 errors corrected on average. This rate is much higher than both `Static` and `Dynamic` forgeries, which is particularly alarming given that we infer online information from general population statistics.

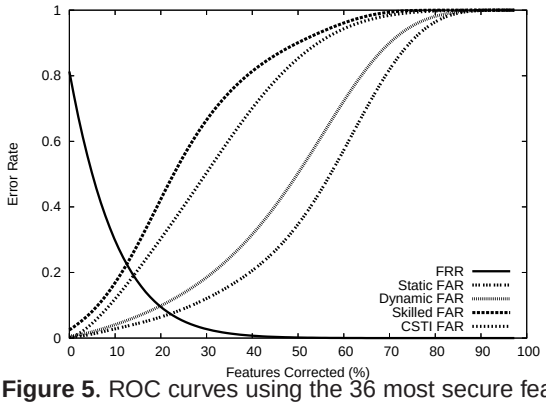

Figure 5. ROC curves using the 36 most secure features out of a set of 145.

The performance against the new, stronger, feature set is shown in Figure 5. The `Static` and `Dynamic` forgeries exhibit an EER of 6.8% and 8.2% at 8 and 7 errors corrected, respectively. The EER for `Skilled` forgeries is 20.1% at 4 errors corrected. The EER for forgeries generated using our generative approach is 17.2% at 5 errors corrected, which is close to the rate of skilled forgers. As expected, the choice of a strengthened feature set reduced the EERs in each case. Nonetheless, forgeries generated using this method performed almost as well as skilled forgers, and were much more successful than forgeries generated under traditional assumptions.

These results are of significant practical value as we assume access to only a limited number of offline samples of the target's writings (on average, 6.9 samples of length 1.83 characters for each forgery attempt), each written in a context outside its use for security. Again, recall that the individual pieces have simply been traced (to infer stroke order) as a determined adversary would likely do. Therefore, stroke order, direction, and other user-specific spatial

idiosyncrasies could be incorrectly reproduced. Nonetheless, our attack outperforms average forgers even with access to real-time representations of the passphrase as it was originally rendered.

It is interesting to note that the handwriting of several users was especially susceptible to our attack (e.g., the target writer in Figure 1), whereas other users produced writings that were quite resilient to our forgeries (e.g., the target writer in Figure 2). This multi-modal behavior might be explained by the fact that our final velocity profiles represent "average" velocity profiles. It seems likely that those writers who are close to the average are more susceptible than those with unique writing habits. Nevertheless, in the context of cryptographic key-generation or authentication, the ability to accurately forge any writer can have significant ramifications as an adversary often only needs to compromise the weakest link in the system.

5 Conclusions

In this paper, we have we presented a generative attack against an online handwriting biometric using only a small number of offline samples of the user's writing in concert with general population statistics. This approach achieves success rates close to our best human forgers.models. Our conclusions concerning this threat apply to any attempt to extract distinguishing features from a user's normal style of handwriting. As it stands, however, they do not directly apply to more stylized writing such as a true signature. For the SVC competition [22], users were asked to create a "new" signature and "practice" it. It would be interesting to know whether such pseudo-signatures might provide a solution to this problem.

In a broader sense, our work demonstrates that traditional approaches to evaluating biometric security, which are most often based on an average-case analysis, are insufficient to characterize the threats posed by determined adversaries who possess some degree of talent and/or automated attacks that exploit generative models for human behavior. Increased confidence will come only through more careful consideration of such worst-case scenarios.

6 Acknowledgments

We express our gratitude to Dishant Patel, Carolyn Buckley, Jay Zarfoss and Charles Wright for their assistance. We also thank the numerous people who participated in this study. This project is supported by NSF grant CNS-0430338.

References

[1] L. Ballard, F. Monrose, and D. Lopresti. Biometric authentication revisited: Understanding the impact of wolves in sheep's clothing. To appear in *Proceedings of the 15^{th} Annual USENIX Security Symposium*, Vancouver, BC, Canada, August, 2006.

[2] G. Boccignone, A. Chianese, L. P. Cordella, and A. Marcelli. Recovering dynamic information from static handwriting. *Pattern Recognition*, 26(3):409–418, 1993.

[3] D. S. Doermann and A. Rosenfeld. Recovery of temporal information from static images of handwriting. *International Journal of Computer Vision*, 15(1-2):143–164, 1995.

[4] S. J. Elliott. Development of a biometric testing protocol for dynamic signature verification. In *Proceedings of the International Conference on Automation, Robotics, and Computer Vision*, Singapore, 2002.

[5] M. C. Fairhurst. Signature verification revisited: promoting practical exploitation of biometric technology. *Electronics & Communication Engineering Journal*, pages 273–280, December 1997.

[6] H. Feng and C. Wah. Private key generation from on-line handwritten signatures. *Information Management and Computer Security*, 10(4):159–164, 2002.

[7] I. Guyon. Handwriting synthesis from handwritten glyphs. In *Proceedings of the Fifth International Workshop on Frontiers of Handwriting Recognition*, Colchester, England, 1996.

[8] G. Hinton and V. Nair. Inferring motor programs from images of handwritten digits. In *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.

[9] A. K. Jain, F. D. Griess, and S. D. Connell. On-line signature verification. *Pattern Recognition*, 35(12):2963–2972, 2002.

[10] Y. W. Kuan, A. Goh, D. Ngo, and A. Teoh. Cryptographic keys from dynamic hand-signatures with biometric security preservation and replaceability. In *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 27–32, Los Alamitos, CA, 2005. IEEE Computer Society.

[11] P. M. Lallican, C. Viard-Gaudin, and S. Knerr. From offline to on-line handwriting recognition. In *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, page 303312, September 2000.

[12] K. K. Lau, P. C. Yuen, and Y. Y. Tang. Recovery of writing sequence of static images of handwriting using uwm. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, volume 2, pages 1123–1128, August 2003.

[13] D. P. Lopresti and J. D. Raim. The effectiveness of generative attacks on an online handwriting biometric. In *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*, pages 1090–1099. Hilton Rye Town, NY, USA, 2005.

[14] F. Monrose, M. Reiter, Q. Li, D. Lopresti, and C. Shih. Towards speech-generated cryptographic keys on resource-constrained devices. In *Proceedings of the Eleventh USENIX Security Symposium*, pages 283–296, 2002.

[15] R. Plamondon. A delta-lognormal model for handwriting generation. In *Proceedings of the Seventh Biennal Conference of the International Graphonomics Society*, pages 126–127, London, Ontario, Canada, 1995.

[16] K. Price. Bibliography: On-line signatures, March 2006. http://iris.usc.edu/Vision-Notes/bibliography/char1011.html.

[17] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: Issues and challenges. *Proceedings of the IEEE: Special Issue on Multimedia Security of Digital Rights Management*, 92(6):948–960, 2004.

[18] T. Varga, D. Kilchhofer, and H. Bunke. Template-based synthetic handwriting generation for the training of recognition systems. In *Proceedings of the 12th Conference of the International Graphonomics Society*, pages 206–211, June 2005.

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

[20] 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.

[21] C. Vielhauer and F. Zöbisch. A test tool to support brute-force online and offline signature forgery tests on mobile devices. In *Proceedings of the International Conference on Multimedia and Expo*, volume 3, pages 225–228, 2003.

[22] D.-Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll. SVC2004: First international signature verification competition. In *Proceedings of the International Conference on Biometric Authentication (ICBA)*, Hong Kong, July 2004.