The Effectiveness of Generative Attacks on an Online Handwriting Biometric

Daniel P. Lopresti and Jarret D. Raim

Department of Computer Science and Engineering
Lehigh University
Bethlehem, PA 18015, USA
{lopresti,jdr6}@cse.lehigh.edu

Abstract. The traditional approach to evaluating the performance of a behavioral biometric such as handwriting or speech is to conduct a study involving human subjects (naïve and/or skilled "forgers") and report the system's False Reject Rate (FRR) and False Accept Rate (FAR). In this paper, we examine a different and perhaps more ominous threat: the possibility that the attacker has access to a generative model for the behavior in question, along with information gleaned about the targeted user, and can employ this in a methodical search of the space of possible inputs to the system in an attempt to break the biometric. We present preliminary experimental results examining the effectiveness of this line of attack against a published technique for constructing a biometric hash based on online handwriting data. Using a concatenative approach followed by a feature space search, our attack succeeded 49% of the time.

1 Introduction

It is standard practice in biometric authentication to test a new system and report how well that system performs. In most cases, this information takes the form of FRR (False Reject Rate) and FAR (False Accept Rate) curves. Often, researchers perform studies with groups of university students and/or other volunteers playing the role of the attacker (*e.g.*, [3, 4]).

While such evaluations shed some light on the quality of the biometric, they do not always provide a full picture of the overall security provided by the system. In this paper, we examine a fundamentally different type of threat: the possibility that an attacker has access to a *generative model* for the behavior in question, *i.e.*, an algorithm which can be used to synthesize a signal that mimics true human input. Such models exist, for example, for speech and for handwriting, as well as for other physiological phenomena. By combining this with information gleaned about the targeted user (*e.g.*, samples of the user's speech or handwriting obtained surreptitiously), an adversary could conceivably conduct a methodical search of the space of possible inputs to the system in an attempt to break the biometric.

We present preliminary results for attacks on a published technique for constructing a biometric hash based on online handwriting data, and conclude by discussing possible areas for future exploration.

T. Kanade, A. Jain, and N.K. Ratha (Eds.): AVBPA 2005, LNCS 3546, pp. 1090–1099, 2005.

2 Related Work

Much work has been done in the area of testing biometric systems for security and performance. Brömme and Kronberg [1] proposed a system that integrates into state of the art operating systems such as Windows 2000 and Linux/UNIX. This framework allows the biometric system to log all information about its operation for later inspection. In this way, real-world conditions can be studied. The Brömme and Kronberg system is geared towards providing more accurate feedback to the current maintainers of an already-deployed biometric, however. It does not try to test a system from the point of view of a determined attacker, nor does it allow researchers to compare systems that have yet to be deployed.

Another technique to help researchers test biometrics based on handwriting has been developed by Vielhauer and Zoebisch [9]. This tool allows researchers to study forgeries generated by a human with access to static and dynamic representations of the true signal. The system presents the human forger with several different prompts containing increasing information about the targeted handwriting. The "attacker" first records a test sample with no information. He/she is then shown a static representation of the true writing and asked to input another test sample. Lastly, a dynamic representation of the handwriting is displayed and the user is allowed to input one more test sample. Depending on the characteristics of the test writer and the true signal, the accuracy of the forgeries will vary widely.

Another approach to more directly studying the security provided by a biometric system is presented by Monrose, *et al.* in [5]. This paper, which provides the primary motivation for our present work, describes several types of attacks against a speaker authentication system. The tested system extracts features from the user's voice, drawing entropy from both the passphrase spoken by the user and how the passphrase was spoken. These features are then used to extract a key from a data structure in which pieces of the true key are intermingled with random data. This process makes it difficult for an attacker in possession of the device to obtain any of the sensitive information stored on it.

The speech system in the above study was attacked using several methods. The first was a standard human impostor, whereby someone other than the true user tries to authenticate against the biometric. Next, a text-to-speech (TTS) system was used to generate sequences of phonemes for the passphrase, with various input parameters governing the type of speech produced. Lastly, a crude cut-and-paste attack was attempted, employing a large inventory of the true user's speech. Phonemes which had been manually labeled in the inventory were selected and concatenated to yield the targeted passphrase. Both the TTS and cut-and-paste attacks were able to out-perform random guessing, but did not work well enough to break the biometric. These results suggest, however, that as an attacker acquires more information, it becomes easier to breach the system.

In a test of another speech-based system, Masuko, *et al.* [4] attempted to use information about the pitch of voice samples to enable the system to reject synthetically created speech. They proved that current speech authentication

could be fooled over 20% of the time by using trained speech synthesis systems, and that pitch information was not useful in rejecting synthesized speech.

With generative models for handwriting appearing in the literature (*e.g.*, [2, 6]), we seek to adapt this style of investigation to the handwriting verification problem. We note that, as in [5, 8], we are not concerned with a user's one-and-only (*i.e.*, legal) signature, but rather the idiosyncratic way the user writes an arbitrary pre-selected passphrase of his/her own choosing.

3 Attack Models

To increase the amount of knowledge about the security provided by a given biometric, a model of the operation of the system is needed. This model must take into account all of the possible vulnerabilities of the system and provide ways for testing those vulnerabilities. By allowing a finer grained comparison of systems, individual components can be contrasted with one another. A system with low FRR and FAR might not be as secure as one with higher error rates, but a better-defined (more comprehensive and realistic) security model.

Because there are so many different kinds of information that could help an attacker breach a system, an exhaustive taxonomy is beyond the scope of this paper. We instead confine ourselves to exploring one line of attack, using techniques that should be generalizable to other scenarios.

For the present study, the types of handwritten inputs we consider include:

Class 1. Different User, Different Passphrase. Sometimes referred to as a "naïve forgery."
Class 2. Different User, True Passphrase. Different user writing the same passphrase as the true user.
Class 3. True User, Different Passphrase. True user writing something other than the passphrase.
Class 4. Concatenation Attack. Passphrase created from online samples of the true user writing non-passphrase material.
Class 5. True User, True Passphrase. The keying material, provided as a baseline for reference.

Certain of these input classes were chosen based on the types of attacks usually reported in the biometric literature. Class 1 is the typical brute-force type of attack while Class 2 is closer to a so-called "skilled forgery." In testing Class 3, we hope to show that even if the attacker has access to online samples of the true user's handwriting, more work must be done to use that information to reduce the possible search space.

The representative generative model in the current test is Class 4 (we plan to study other generative models in the near future). Here we employ samples of the user's handwriting collected separately from the passphrase. These samples are manually segmented into basic units, which can be individual characters, bigrams, trigrams, etc., and then labeled. The generative model accepts as input a labeled inventory and the targeted passphrase and produces a random sequence

of concatenated units that, when rendered, attempts to simulate the user writing the passphrase. Note that both the appearance of the writing as well as the dynamics are reproduced.

Lastly, Class 5 is provided as a baseline reference to contrast the other classes to the intended input. In most biometric systems, some allowances must be made to ensure that the true user is able to authenticate despite natural variations in handwriting.

As was the case in [5], our primary interest in this work is in offline attacks, a situation that might arise when the biometric is employed to generate a secure hash or cryptographic key to be used in protecting confidential information stored on a mobile device, for example. The handwritten input is provided to the system which generates a set of features as output. The range of acceptable inputs for the true user can be viewed as defining a subspace over the entire feature space. Ignoring the unlikely event of an exact match on the first attempt (a perfect forgery), the attacker's goal, then, is to explore the space around the feature vector returned by the forgery as rapidly as possible in the hopes of uncovering the correct setting. We assume, of course, that the attacker has no way of knowing whether the forgery is good enough to fall close to a true input until the match is actually found, but once that happens, the attacker is able to tell that the system has been broken. Hence, the attacker will conduct a methodical search, working outwards from the feature vector for a certain period of time before concluding that the forgery was not good enough and moving on to try another input.

While the attack models we have presented are quite simple, they are sufficient to motivate interesting tests of published biometrics, provide an indication of the associated combinatorics, and illustrate the difficulty (or ease) with which specific systems can be broken.

4 Experimental Evaluation

To examine the impact of the models described above, several example attacks were created. For testing purposes, we chose to implement the Vielhauer, *et al.* system [8] for biometric hashing. Based on a small data set, standard FRR and FAR measures were used to determine appropriate parameter settings for our later attempts at attacking the system. We then evaluated the effectiveness of each of the classes of inputs described in the previous section,

4.1 Data Sets

For our experiments, several small data sets were created. Two writers (the authors) wrote four different passphrases 20 or more times, which resulted in a total of 154 samples. The handwriting was collected using a Wacom Intuos digitizing tablet. While this data set is small in comparison to results typically reported in the literature, it is still possible to draw conclusions due to the specific nature of our study: we are not attempting to prove that a proposed

biometric is secure, rather, we are trying to examine whether attacks based on generative models can be successful. Since we are comparing the effectiveness of attack strategies and not the overall security of a system, the size of the data set is not a serious issue provided the phenomenon of interest, the breaking of the system, is seen to occur[1]. Two examples from this data set are shown in Fig. 1.

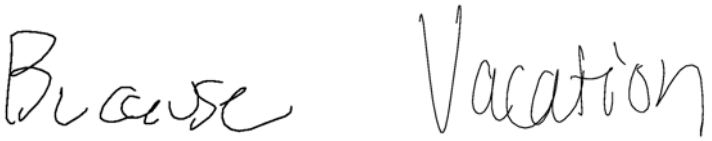
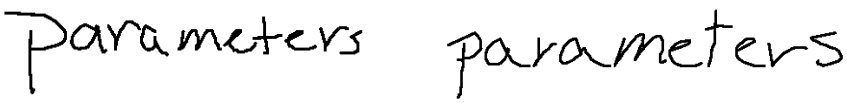


Fig. 1. Handwriting samples

As noted previously, to execute the concatenative attack (Class 4), it is assumed that the attacker has access to online handwriting samples of the targeted user as well as knowledge of the true passphrase. (It can also be assumed that the attacker has an offline image of the user's passphrase, but this was not used in our current study.) A separate set of online writing samples were labeled as to which stroke sequences corresponded to individual characters. This resulted in a corpus of possible n-gram combinations of the user's handwriting. To generate a synthetic handwritten passphrase, strokes were concatenated from the corpus to form the correct text of the passphrase. No scaling or smoothing was performed, however the individual stroke sequences were placed on a similar baseline and appropriate timestamps were recreated. An example of a passphrase synthesized using this approach appears in Fig. 2(b).



(a) Target passphrase. (b) Concatenative attack.

Fig. 2. Example of a concatenative attack

4.2 Biometric System

A detailed discussion of the Vielhauer, *et al.* biometric hash can be found in [8], but some knowledge of the system will be helpful for a greater understanding of the attacks discussed below. The system is based on 24 integer-valued features extracted from an online writing signal. The signal consists of $[x, y]$ position and timing information. Fourteen of the features are global, while the remaining

[1] A good analogy here are studies on the susceptibility of traditional password security systems to dictionary-based attacks. If a system with two passwords can be broken in such fashion, then certainly systems with larger numbers of passwords are even more susceptible. Nevertheless, we recognize the value of larger data sets and plan additional collection activities in the near future

ten features are concerned with 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. A listing of the features is provided in Table 1.

Table 1. Features employed in the Vielhauer, *et al.* biometric hash [8]

1. Number of strokes	13. Effective writing velocity in x
2. Total writing time (ms)	14. Effective writing velocity in y
3. Total number of samples (points)	15. Integrated area under x, segment 1
4. Sum of all local (x,y) minima and maxima	16. Integrated area under x, segment 2
5. Aspect ratio (x/y) * 100	17. Integrated area under x, segment 3
6. Pen-down / total writing time * 100	18. Integrated area under x, segment 4
7. Integrated area covered by x signal	19. Integrated area under x, segment 5
8. Integrated area covered by y signal	20. Integrated area under y, segment 1
9. Average writing velocity in x	21. Integrated area under y, segment 2
10. Average writing velocity in y	22. Integrated area under y, segment 3
11. Average writing acceleration in x	23. Integrated area under y, segment 4
12. Average writing acceleration in y	24. Integrated area under y, segment 5

To train the system to accept a given user, features are extracted and used to create a biometric hash where each feature generates a corresponding integer value. In addition, an interval matrix is created containing information needed for future testing. This information could be stored by the system itself or by the user in a portable format such as a USB key. A transitive enrollment system may be employed and would help in achieving a strongly-correlated set of sample data for the true user [7], but was not used for our experiments.

When a user attempts to authenticate, he/she provides a new handwriting sample and a claim to a certain identity. Features are extracted from the sample and passed through the hash generation. If a certain feature falls within the accepted range of values (plus some tolerance threshold), it generates the same integer hash. In our case, the size of the training set varied with the sample being tested, but ranged between 15 to 25 samples per class. To generate FRR and FAR curves for the system, the sample size for the training set cross-validation was varied between 5 and 10. Several potential tolerance values were also checked: 0.0, 0.01, 0.05, 0.1, 0.15, 0.2, and 0.3. The most promising graph is shown in Fig. 3, where the equal error rate tolerance threshold is found to be 0.15.

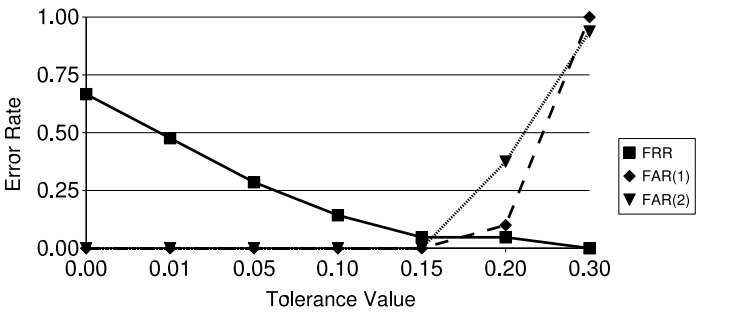


Fig. 3. Error rates for the Vielhauer, *et al.* hash on data used in our tests

As can be seen, the system performed quite well, with a FRR of 5%. In addition, two FAR curves were generated. The first is a naïve (Class 1) forgery, with one writer writing a passphrase that was used to test against the other writer writing a different passphrase. In this case, the FAR was 0%. In another test, one writer writing a passphrase was tested against the other writer writing the same passphrase (a Class 2 forgery). This case also resulted in a FAR of 0%.

To test how well each individual feature performed, a standard cross validation FAR test was run for all attack types. When a feature mismatch was detected, a note was made as to which feature failed and by what magnitude. This investigation showed that when a hash element is missed, the magnitude of the miss is generally ± 2 of the actual value. It was also interesting to see that several features yielded constant values for all of our handwriting samples. This could mean that these hash elements, and by extension the features that generated them, are not useful in the kinds of experiments we are performing. However, we believe more research is needed before drawing such conclusions.

4.3 Feature Space Search

As noted in Sect. 3, the search we performed attempts to find the true hash starting with the hash generated from the handwriting sample. The search begins with small alterations of the given hash and works outward until a predetermined time limit has been met. Our tests had a time limit of 60 seconds and were conducted on a Pentium 4 desktop PC running at 3.2 GHz with 1 GB of RAM. This machine was able to generate and check 540,000 search possibilities per second.

Results for the five different input classes are shown in Figs. 4 and 5. The first graph shows the min, mean, and max number of feature misses per sample, while the second graph shows how long it took the search process to correct the initial hash vector (when that was possible within the 60 second time limit).

It can be seen in Fig. 4 that the features used by the Vielhauer, *et al.* hash have several desirable qualities. Class 2 (Different User, Same Passphrase) and

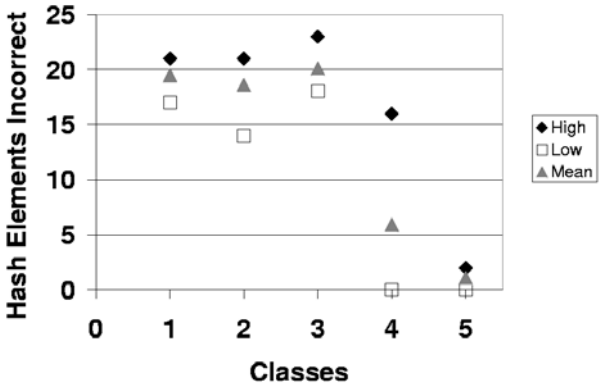


Fig. 4. Incorrect hash elements per input class

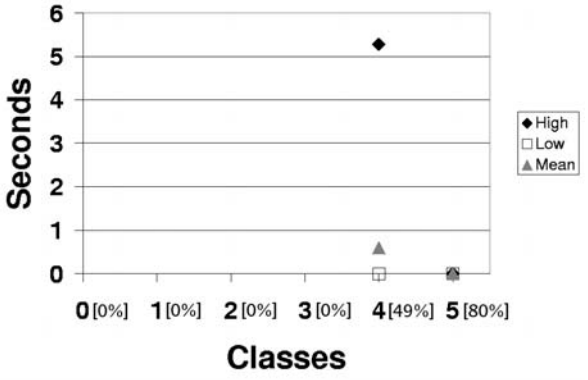


Fig. 5. Time to correct hashes per input class (with percent accepted)

Class 3 (Same User, Different Passphrase) have very similar means (18.6 vs. 20.1) and their maximum and minimum values are also comparable. This shows that some features of the biometric system are sensitive to the passphrase written, which accounts for the errors for Class 3, and some features are sensitive to the writer, which accounts for the errors seen in Class 2.

The mean incorrect number of hash elements for Class 1 is 21, which is higher than for all other classes, as would be expected. It is also of note that none of the Class 1, 2 or 3 hashes were broken in the 60 second search limit. While 60 seconds seems like a low bound, the number of possibilities for the search increases exponentially based on the number of incorrect hash elements. Still, more efficient search algorithms, faster machines, and longer runtimes could have an effect on the probability of the search finding the correct hash. As would be expected, Class 5 (keying material) was either accepted without any modifications or only required a maximum search time of 0.00019 seconds.

The concatenative attack, Class 4, displayed interesting behavior. Class 4 had the highest variance ranging from 0 to 16 hash elements incorrect, while none of the other classes had a range over 7. However, even with this distribution, the mean number of hash elements incorrect was still only 5.92, which put it below Classes 1, 2 and 3. The high variance means that many of the concatenated passphrases generated hashes with few to no hash elements incorrect. We note that 5% of Class 4 hashes were correct at the onset and required no search. When the standard 60 second search was allowed, 49% of the hashes were correctable, with an average search time of 5.28 seconds on those that were broken. Class 4 was the only class outside of the keying material (Class 5) that was able to generate hashes that required no search.

5 Conclusions

Popular measures for evaluating the performance of biometric systems may fail to capture certain kinds of threats. By limiting testing to human subjects and

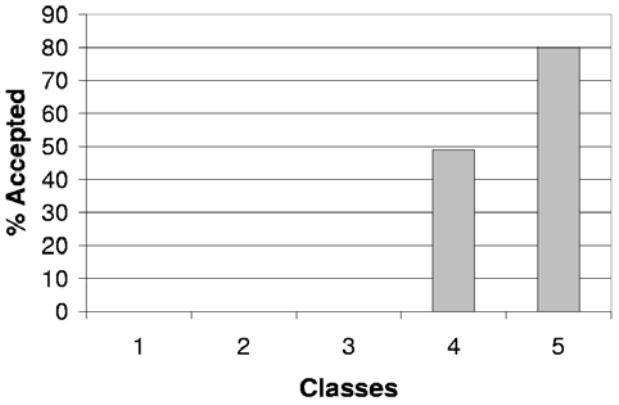


Fig. 6. Percentage of handwritten passphrases in each class accepted after search

reporting only results for FRR and FAR, the determined attacker is ignored. As more and more sensitive information is stored on portable computing devices, the incentives for breaking such systems becomes greater.

The attack models we have begun to study will help increase our understanding of potential flaws in biometric security, hopefully before they can be exploited. As can be seen in Fig. 6, we were able to achieve a 49% success rate using the concatenative attack described in this paper against a scheme for creating biometric hashes from online handwriting data[2].

The concatenative attack we have presented is only one possible avenue an adversary might take, as outlined in Sect. 3. We plan to study other forms of attack, including Plamondon's delta-log normal generative model [6] and Guyon's handwriting synthesis method [2]. These techniques will allow a full parameterization of the search space and may prove even more devastating. Other schemes for attempting to create secure hashes from a user's handwriting should likewise be evaluated in this fashion.

Testing with larger, more extensive data sets is also planned. By using tablet PC's and commercial signature capture tablets, we hope to better approximate a true distribution of users. These larger data sets will allow a more thorough examination of the feature space as well as the differences between handwritten passphrases and traditional "legal" signatures. Studying these issues in the context of other biometric measures, including speech, to build on the work that was first reported in [5], is another topic for future research.

Acknowledgments

The authors wish to thank Fabian Monrose for his helpful comments on an earlier draft of this paper.

[2] It is important to note that we are not singling out the system proposed by Vielhauer, *et al.* in [8] for criticism. The techniques presented in this paper should be applicable to many other behavioral biometrics with some amount of adaptation

References

1. A. Bromme and M. Kronberg. A conceptual framework for testing biometric algorithms within operating systems' authentication. In *Proceedings of the ACM Symposium on Applied Computing*, pages 273–280, 2002.
2. I. Guyon. Handwriting synthesis from handwritten glyphs. In *Proceedings of the Fifth International Workshop on Frontiers in Handwriting Recognition*, pages 140–153, 1996.
3. J. Lindberg and M. Blomberg. Vulnerability in speaker verification – a study of possible technical impostor techniques. In *EUROSPEECH*, pages 1211–1214, 1999.
4. T. Masuko, K. Tokuda, and T. Kobayashi. Imposture using synthetic speech against speaker verification based on spectrum and pitch. In *Proceedings of the Sixth International Conference on Spoken Language Processing*, volume 2, pages 302–305, 2000.
5. 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.
6. R. Plamondon. A delta-lognormal model for handwriting generation. In *Proceedings of the Seventh Biennal Conference of the International Graphonomics Society*, pages 126–127, 1995.
7. C. Vielhauer, R. Steinmetz, and A. Mayerhöfer. Transitivity based enrollment strategy for signature verification systems. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, volume 2, pages 1263–1266, 2001.
8. 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.
9. 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.