

Biometric Authentication Revisited: Understanding the Impact of Wolves in Sheep's Clothing

Lucas Ballard

Fabian Monroe

Daniel Lopresti

February 2006

Technical Report LU-CSE-06-023

Department of Computer Science and Engineering
Lehigh University
Bethlehem, PA 18015 USA

<http://www.cse.lehigh.edu/>



LEHIGH
UNIVERSITY

Computer Science and Engineering

Computer Science and Engineering

Computer Science and Engineering

Computer Science and Engineering

CSE

Biometric Authentication Revisited: Understanding the Impact of Wolves in Sheep's Clothing*

*Lucas Ballard*¹ *Fabian Monroe*¹ *Daniel Lopresti*²

¹ Security and Privacy Applied Research (SPAR) Lab,
Department of Computer Science,
Johns Hopkins University,
Baltimore, MD 21218
{lucas, fabian}@cs.jhu.edu

² Department of Computer Science and Engineering,
Lehigh University,
Bethlehem, PA 18015
lopresti@cse.lehigh.edu

February 2006

Abstract

Biometric security is a topic of rapidly growing importance, especially as it applies to user authentication and key generation. In this paper, we describe our initial steps toward developing evaluation methodologies for behavioral biometrics that take into account threat models which have largely been ignored. We argue the pervasive assumption that forgers are minimally motivated (or, even worse, naïve), or that attacks can only be mounted through manual effort, is too optimistic and even dangerous. To illustrate our point, we analyze a handwriting-based system used for key generation and show that the standard approach of evaluation over-estimates the security of the system by almost 400%. Our results highlight a number of pressing concerns that must be addressed before biometric-based schemes are put into practical use. Additionally, to overcome current labor-intensive hurdles in performing more accurate assessments of system security, we present a *generative attack* model based on concatenative synthesis that can provide a rapid indication of the security afforded by the system. We show that our generative attacks match or exceed the effectiveness of forgeries rendered by skilled humans.

*A revised version of this paper will be presented at the *Fifteenth USENIX Security Symposium*, Vancouver, BC, Canada, July-August 2006.

1 Introduction

The security of many systems relies on obtaining human input that is assumed to not be readily reproducible by an attacker. Passwords are the most common example, though the assumption that these are not reproducible is sensitive to the number of guesses that an attacker is allowed. In so-called *online* attacks, the adversary must submit each request to a nonbypassable reference monitor (e.g., the login prompt), that accepts or declines the password and, in particular, permits a limited number of incorrect attempts. In contrast, an *offline* attack permits the attacker to make a number of guesses at the password that is limited only by the resources available to the attacker, i.e., time and memory. When passwords are used as cryptographic keys, they become susceptible, and sometimes succumb, to offline attacks.

An alternative form of user input that is intended to be difficult for attackers to reproduce are biometrics. Like passwords, biometrics have typically been used as a technique for a user to authenticate herself to a reference monitor that can become unresponsive after too many failed attempts. However, biometrics also have been explored as a means for generating user-specific cryptographic keys [41, 31, 32, 45]. As with password-generated keys, there is insufficient evidence, for most types of biometrics, that keys generated from biometric features alone will typically survive offline attacks. As such, an alternative that we and others have previously explored is *password hardening* whereby a cryptographic key is generated from both a password and dynamic biometric features of the user while entering it [31].

While these directions may indeed allow for the use of biometrics in a host of applications, we believe the manner in which biometric systems have been tested in the literature (including our prior work) raises some concerns. In particular, this work demonstrates the need for adopting more realistic adversarial models when performing security analyses. Indeed, as we show later, the impact of forgeries generated under such conditions help to better understand the security (or lack thereof) of certain biometric-based schemes.

Our motivation for performing this analysis is primarily to show that there exists a dramatic disconnect between realistic threats and typical “best” practices [26] for reporting biometric performance; one which requires serious rethinking as both industry and the research community gains momentum in the exploration of biometric technologies. We believe that the type of analysis presented herein (which considers far more sophisticated adversaries than usually found in the biometric literature to date), is of primary importance for the use of biometrics for authentication and cryptographic key generation (e.g., [32, 11, 14, 4, 20]), where *weakest-link* analysis is paramount. Unfortunately, none of these work address the feasibility of attacks from more than naïve impersonators.

Moreover, to raise awareness of this failing we explore a particular type of methodology in which we assume that the adversary utilizes indirect knowledge of the target user’s biometric features. That is, we presume that the attacker has observed, for the type of biometric of interest, measurements of that biometric in contexts outside its use for security. For example, if the biometric is the user’s handwriting dynamics while providing input via a stylus, then we presume the attacker has samples of the user’s handwriting in another context, captured hardcopies of the user’s writing, or writings from users of a similar style.

We argue that doing so is more reflective of the real threats to biometric security. In this paper, we explore how an attacker can use such data to build *generative models* that predict how a user would, in this case, write a text, and evaluate the significance of this to biometric authentication.

We note that our goal of examining generative attacks is not solely to demonstrate the attacks themselves, but rather to improve biometric authentication and password hardening approaches to resist them. That said, some biometrics will be intrinsically susceptible to generative attacks, static biometrics (e.g., fingerprint, retina, etc.) being prime examples: once recorded in another context, they can be trivially “generated” since they do not change (see for example [42]). For this reason, reference monitors must confirm that they are measuring a biometric feature from the human directly (e.g., by sensing blood flow) rather than a generated copy. However, there seems little that can be done to save static biometrics against generative (potentially offline) attacks. For this reason, we currently focus our attention on dynamic biometrics, and highlight what we believe are to be short-comings in current practices.

2 Biometric Authentication

Despite the obvious diversity of approaches implied by the attention that biometrics have received, from the standpoint of this investigation, several key points remain relatively constant. For instance, the traditional procedure for applying a biometric as an authentication paradigm involves sampling an input from the user, extracting an appropriate set of features, and comparing these to previously stored templates to confirm or deny the claimed identity. While a wide range of features have been investigated (and there is still ongoing debate about which are most appropriate for a given biometric), it is universally true that system designers seek features that exhibit large inter-class variability and small intra-class variability. In other words, two different users should be unlikely to generate the same input features, while a single user ought to be able to reproduce her own features accurately and repeatably.

Likewise, the evaluation of most biometric systems usually follows a standard model: enroll some number of users by collecting training samples, e.g., of their handwriting or speech. At a later time, test the rate at which users’ attempts to recreate the biometric to within a predetermined tolerance fail. This failure rate is denoted as the False Reject Rate (FRR) or Type I error. Additionally, evaluation usually involves assessing the rate at which one user’s input (i.e., the impostor) is able to fool the system when presented as coming from another user (i.e., the target). This evaluation yields the False Accept Rate (FAR) or Type II error for the system under consideration. The tolerance setting is vital in assessing the limits within which a sample will be considered as genuine, while at the same time, balancing the delicate trade-off of resistance to forgeries. Typically, one uses the equal error rate (EER)—that is, the point at which Type I and Type II errors are equal—to describe the accuracy of a given biometric system. Essentially, the lower the EER, the higher the accuracy.

Researchers also commonly distinguish between forgeries that were never intended to

defeat the system (“random” or naïve forgeries), and those created by a user who was instructed to make such an attempt given information about the targeted input (i.e., so-called “skilled” forgeries). The evaluation, however, of biometrics under such weak security assumptions can be misleading. In fact, some have argued that many pattern recognition problems (e.g., signature verification) raise numerous difficulties particularly with respect to **Type II** error evaluation — or the *real* risk of accepting forgeries. Indeed, it may even be argued that as there is no strong means by which one can define a good forger and prove her existence (or non-existence) that such analysis is theoretically impossible [40]. Nevertheless, the biometric community continues to rely on relatively simple measures of adversarial strength, and most studies to date only incorporate unskilled adversaries, and very rarely, “skilled” impersonators.

This general practice is, however, troubling as the evaluation of **Type II** errors is likely to be significantly underestimated [40, 38]. Moreover, we believe that this relatively ad-hoc approach to evaluation misses a significant threat: the use of *generative models* to create synthetic forgeries which can form the basis for sophisticated *automated* attacks on biometric security. This observation was recently reiterated in [43] where the authors conjectured that although the complexity of successful impersonations on various biometric modalities can be made formidable, biometric-based systems might be defeated using various strategies (see for example [42, 50, 13, 35, 32, 21, 24, 28]). As we show later, even rather simplistic attacks launched by successive replication of synthetic or actual samples from a representative population can have adverse effects on **Type II** errors — particularly for the weakest users (i.e., the so-called “Lambs” in the biometric jargon for a hypothetical menagerie of users [7]).

In what follows, we provide what we believe is the most in-depth study to date that emphasizes the extent of this problem. Furthermore, as a first step to provide system evaluators with a stronger methodology for quantifying performance under various threats, we describe our work on developing a prototype toolkit using handwriting dynamics as a case in point.

3 Handwriting authentication: *an exemplar*

Research on user authentication via handwriting has had a long, rich history, with hundreds of papers written on the topic. The majority of this work to date has focused on the problem of signature verification [36]. Signatures have 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 [9]. While each individual has only one true signature — a notable limitation — handwriting in general contains numerous idiosyncrasies that might allow a writer to be identified.

In considering the mathematical features that can be extracted from the incoming signal to perform authentication, it is important to distinguish between two different classes of inputs. Data captured by sampling the position of a stylus tip over time on a digitizing tablet or pen computer are referred to as *online* handwriting, whereas inputs that presented in the form of a 2-D bitmap (e.g., scanned off of a piece of paper) are referred to as *offline* handwriting. To avoid confusion with the traditional attack models in the security

community, later on in this paper we shall eschew that terminology and refer to the former as covering both temporal and spatial information, whereas the latter only covers spatial information. Features extracted from offline handwriting samples include bounding boxes and aspect ratios, stroke densities in a particular region, curvature measurements, etc. In the online case, these features are also available and, in addition, timing and stroke order information that allows the computation of pen-tip velocities, accelerations, etc. Studies on signature verification and the related topic of handwriting recognition often make use of 50 or more features and, indeed, feature selection is itself a topic for research. The features we use in our own work are representative of those commonly reported in the field [12, 44, 27, 22]. Repeatability of features over time is, of course, a key issue, and it has been found that dynamic and static features are equally repeatable [12].

In the literature, performance figures (i.e., **EER**) typically range from 2% to 10% (or higher), but are difficult to compare directly as the sample sizes are often small and test conditions quite dissimilar [8]. Unfortunately, forgers are rarely employed in such studies and, when they are, there is usually no indication of their proficiency. Attempts to model attackers with a minimal degree of knowledge have involved showing a static image of the target signature and asking the impostor to try to recreate the dynamics – see, e.g., [10, 33] for studies along these lines. The only serious attempt we are aware of, previous to our own, to provide a tool for training forgers to explore the limits of their abilities is the work by Zoebisch and Vielhauer [50]. In a small preliminary study involving four users, they found that showing an image of the target signature increased false accepts (i.e., Type II error), and showing a dynamic replay doubled the successability to forgeries yet again. However, since the verification algorithm used was extremely simplistic and they do not report false reject rates, it is difficult to draw more general conclusions.

To overcome the “one-signature-per-user” restriction, we employ more general passphrases in our research. While signatures are likely to be more user-specific than arbitrary handwriting, results from the field of forensic analysis demonstrate that writer identification from a relatively small sample set is feasible [15]. Indeed, since this field focuses on handwriting extracted from scanned page images, the problem we face is less challenging in some sense since we have access to dynamic features in addition to static. Another concern, user habituation [8], is addressed by giving each test subject enough time to become comfortable with the experimental set-up and requiring practice writing before the real samples are collected. Still, this is an issue and the repeatability of non-signature passphrases is a topic for future research.

3.1 Handwriting Authentication System

In order to have a concrete platform to test our hypothesis, we loosely adapted the system presented in [45, 44] for generation of “biometric hashes”¹. We note that our results are system-independent as we are only evaluating biometric *inputs*, for which we evaluated features that are reflective of the state of the art [22, 27, 12, 44].

¹While we find it convenient to adapt a biometric hashing paradigm motivated by the earlier work of Vielhauer, *et al.*, we are not in a position to make statements concerning the suitability of that scheme for specific security applications.

For completeness, we briefly describe relevant aspects of the system, for a more detailed description see [44]. To input a sample to the system, a human writes a passphrase on an electronic tablet. The sample is represented as three signals parameterized by time. The discrete signals $x(t)$ and $y(t)$ specify the location of the pen on the writing surface at time t , and the binary signal $p(t)$ specifies whether the pen is up or down at time t . The tablet normalizes (e.g., resamples the input) and computes a set of n statistical features (f_1, \dots, f_n) using these signals. These features comprise the actual input to the biometric authentication or key-generation system.

During an enrollment phase, each legitimate user writes a passphrase a pre-specified number (m) of times, and the feature values for each sample are saved. Let $f_{i,1}, f_{i,2}, \dots, f_{i,n}$ denote the feature values for sample i . Using the feature values from each user and passphrase, the system computes a global set of tolerance values ($T = \{\epsilon_1, \dots, \epsilon_n\}$). These tolerances are used to accommodate natural human error, and their derivation is described in more detail in §4. Once the m readings have been captured, a biometric template is generated for each user and passphrase as follows: Let $\ell'_j = \min_{i \in [1,m]} f_{i,j}$, $h'_j = \max_{i \in [1,m]} f_{i,j}$, and $\Delta_j = h'_j - \ell'_j + 1$. Set $\ell_j = \ell'_j - \Delta_j \epsilon_j$, and $h_j = h'_j + \Delta_j \epsilon_j$. The resulting template is an $n \times 2$ array of values $\{\{\ell_1, h_1\}, \dots, \{\ell_n, h_n\}\}$.

Later, when a user provides a sample with feature values f_1, \dots, f_n for authentication, the system checks whether $f_i \in [\ell_i, h_i]$ for each feature f_i . Each $f_i \notin [\ell_i, h_i]$ is deemed an error, and depending on the threshold of errors tolerated by the system, the attempt is either accepted or denied. We note that as defined here, templates are insecure since they leak information about the actual feature values. We omit discussion of securely representing biometric templates (see for example [5, 31, 19]) as this is not a primary concern of this research.

3.1.1 Feature Analysis

Clearly, the security of any biometric system is entirely based on the quality of the features (f_i) used. A thorough analysis of proposed features for handwriting verification is presented in [44], although we argue that the goals in that work sufficiently differ from our own that we believe a new feature-evaluation metric was required. In that work, the quality of a feature was measured by the “deviation” of the feature and entropy of the feature across the population. For our purposes, these evaluation metrics are not ideal: we are not concerned with the entropy of each feature, but rather how difficult the feature is to forge — which we argue is a more important criteria. When systems are evaluated using purely naïve forgeries, then entropy could be an acceptable metric. However, as we show later, evaluation under naïve forgeries is not appropriate ².

As our main goal is to highlight limitations in current practices, we needed to evaluate a robust yet usable system based on a strong feature set. To this end, we implemented 144 “state of the art” features [44, 12, 34, 22] and evaluated each based on a quality metric (Q) defined as follows. For each feature f , we compute the proportion of times that f is missed

²It is interesting to note, however, that each strong feature as defined in [44] may, at the very least, be inferred from what we deemed as the best features. We did, however, find several other features that were not included in the original work.

by legitimate users, denoted r_f , and the proportion of times that f is missed by forgers (with access to dynamic information), denoted a_f ; r_f and a_f are computed across the entire population. Then, $Q(f) = r_f - a_f$, and the range of Q is $[-1, 1]$. Intuitively, features with a quality score of 1 are completely useless—i.e., they are *never* reliably reproduced by original users and are *always* reproduced by forgers. On the other hand, features with scores closer to -1 are highly desirable when implementing biometric authentication systems.

For our evaluation, we divided our feature set into two groups covering the temporal and spatial features, and ordered each according to the quality score. We then chose the top 40 from each group, and disregarded any with a Type I error greater than 10%. Finally, we discounted any features that could be inferred from others (e.g., given the width and height of a passphrase as rendered by a user, then a feature representing the ratio between width and height is redundant). This analysis resulted in what we deem the 37 best features described in Appendix B.

3.1.2 Data Collection

For our empirical analysis, we collected data over several months at two academic institutions. The results reported herein are from 11,038 handwriting samples collected on digitized pen tablet computers from 50 users (members of Biology and Computer Science departments) during several rounds. We used NEC VersaLite Pad and HP Compaq TC1100 tablets as our writing platforms³. The specifics of each round will be addressed shortly. To ensure that the participants were well motivated and provided writing samples reflective of their natural writing (as well as forgery attempts indicative of their innate abilities), several incentives were awarded⁴.

In **round I**, we collected two distinct datasets: the first set, denoted \mathcal{B} , establishes a baseline of “typical” user writing. After habituation on the writing device, users were asked to write five different phrases—comprising two-word oxymorons—ten times each. As this data would later be used to generate biometric authentication templates, users were given specific instructions to write as naturally (and consistently) as possible. Each phrase was displayed in the top-half of the screen, and the user’s writing was collected from the bottom half. The second dataset (\mathcal{G}) represents what we call our generative corpus and consists of a set of another 65 oxymorons (*distinct from that in \mathcal{B}*). The only restrictions placed when choosing the phrases for the generative corpus was that the collection contained coverage at the bi-gram level of the phrases written in \mathcal{B} . As we show later, on average no more than 10.7% of these (i.e, seven phrases) chosen at random from \mathcal{G} were used in our generative attacks. The average elapsed time for **round I** was approximately one hour.

Round II started approximately two weeks later, where we asked the same set of users to again write the five phrases from **round I** ten times. Additionally, the users were asked to forge representative samples (based on writing style, handedness of the original writer, and gender) from **round I** to create two sets of 17 forgeries. First, users were required to forge samples after seeing *only* a static representation. Next, users were asked to forge the

³Writing samples were normalized for cross-platform testing and has no effect on the results.

⁴In addition to light snacks, each participant received gift certificates for iTunes or Amazon.com for each round; many special prizes (eg, most consistent writers, best forgers, most dedicated, etc) were also awarded.

same phrases again, but this time, upon seeing a real-time rendering of the phrase. At this stage, the users were instructed to make use of the real-time ⁵ presentation to improve their rendering of the spatial features (for example, to distinguish between one continuous stroke versus two strokes that overlap) and to replicate the temporal features of the writing (e.g., time, pen tip velocity/acceleration, etc.)

Lastly, in round III, we selected nine users in our population who appeared to have a natural tendency to produce better forgeries than the average user in our study (although we did not include all of the best forgers). This group represents the three “skilled” (but untrained) forgers for each writing styles (i.e., cursive, block, and mixed) for which they appear to have a natural ability to duplicate. Each forger was provided with a general overview and examples of the types of temporal (e.g., stroke order, acceleration, velocity) and spatial (e.g, width, slant, number of strokes) characteristics that handwriting systems typically capture. These forgers were then asked to forge 15 writing samples from a particular writing style, with 60% of the samples coming from the weakest 10 targets, and the other 40% chosen at random.

The experimental setup for these educated forgers is as follows. First, a real-time reproduction of the target sample is displayed (at the top half of the tablet) and the forger is allowed to attempt forgeries (at her own pace) with the option of saving the attempts she liked. She can also select and replay her forgeries and compare them to the target. In this way, she is able to fine-tune her attempts by comparing the two writing samples. Next, she selects the forgery she believes to be her best attempt, and proceeds to the next target. The averaged elapsed time for this round was approximately 2 hours.

4 Evaluation

4.1 Experimental Methodology

As mentioned earlier, the security of biometric systems is generally analyzed with respect to Receiver Operating Characteristic (ROC) curves that reflect the **Type I** and **Type II** errors. However, as two biometric readings are rarely exact, computing these error rates is typically done as a function of the number of errors tolerated by the system. Moreover, as erroneous readings or widely varying user-input are not uncommon, such outliers in biometric data must be removed; failure to do so could adversely affect system usability and security. To remove the outliers in our collected handwriting samples, we assume that the feature values follow a normal distribution. Consider, for example, that for sample S_i , the biometric device extracted features $f_{i,1}, f_{i,2}, \dots, f_{i,n}$. For a given user and passphrase we have m such samples. Thus, we compute μ_j and σ_j as the mean and standard deviation over $f_{1,j}, f_{2,j}, \dots, f_{m,j}$. We say that for a given sample S_i , the feature $f_{i,j}$ is a “feature outlier” if $f_{i,j} \notin [\mu_j - k\sigma_j, \mu_j + k\sigma_j]$, and that S_i is a “sample outlier” if it contains more than δ feature outliers, where k and δ are empirically derived. For the final 37 features used, we

⁵Every effort was made to ensure that the rendering was (on average) within a few *ms* of the actual writing. The success of the skilled forgeries should offer alternative confirmation that this was indeed a non-issue.

set $k = 2$ and $\delta = 3$. Adjusting these parameters beyond these values resulted in negligible change in the **Type I** and **Type II** errors.

Once marked as a sample outlier, the sample is removed and not used to compute error rates. Moreover, if more than 75% of a user’s samples for a given passphrase are classified as sample outliers, then we consider that as a Failure to Enroll (FTE) [26] for that passphrase. In this case, we discarded all of the user’s samples for that passphrase when computing error rates. The resulting FTE for the evaluation system was $\approx 8.7\%$ which is well inline with current practice and indicates the system was not suitable for a small portion of our population — the so-called “goats” [7]. After outlier removal we had access to 79.2% of the original dataset.

4.1.1 Determining Error Rates

To compute the error rates we first (1) derive the necessary tolerance values that will accommodate for human variation during authentication (see § 3.1) (2) derive the biometric templates (using the tolerances), and then (3) compute the **Type I** and **Type II** errors. Note that it would be incorrect to use the same data to both derive tolerances and compute templates, as the results would be highly correlated. Thus, since tolerances are universal parameters, we adopted the following methodology: we randomly segmented the user population into two groups of equal size, namely \mathcal{U}_l and \mathcal{U}_∞ . Then to compute the **Type I** and **Type II** errors, the users in \mathcal{U}_l are used to derive tolerances, and we create templates for users in \mathcal{U}_∞ (using the tolerances from group \mathcal{U}_l). The entire process is repeated 30 times and we report the average error rates across all runs. We elaborate on the specifics of each sub-process below.

Computing Tolerances To compute the tolerance values we adopt the approach described in [45, 44]. There, the tolerance for user u , passphrase p , and feature f_i is computed as follows. Suppose that for feature f_i , we have two sets of feature values $\langle F_{0,i}, F_{1,i} \rangle$ from the samples corresponding to u ’s writing of p . Compute $\ell_i = \min F_{0,i}$, $h_i = \max F_{0,i}$, and $d_i = h_i - \ell_i$. Then, for each $v \in F_{1,i}$ define

$$d_v = \begin{cases} (v - h_i)/d_i & \text{if } v > h_i \\ (\ell_i - v)/d_i & \text{if } v < \ell_i \\ 0 & \text{otherwise} \end{cases}$$

The tolerance value for feature f_i is then defined as the average of all $d_v \in (0, .2]$ for all values of u and p . Values of 0 are ignored because they fall within the initial range and values greater than .2 are removed as outliers [45]. We choose $F_{0,i}$ and $F_{1,i}$ as follows. Given a set of samples S for $u \in \mathcal{U}_l$ and p , set $\nu = \lfloor .75 \times |S| \rfloor$ and $\kappa = |S| - \nu$. We randomly partition S into groups S_0 of size ν and S_1 of size κ , and take as $F_{0,i}$ (resp. $F_{1,i}$) the feature values for f_i from $S_{0,i}$ (resp. $S_{1,i}$) and compute the necessary tolerance value. For completeness, this process is repeated 25 times and we take the average across all iterations for each feature as that feature’s final tolerance value.

Computing Type I Errors Similarly, for a given user $u \in \mathcal{U}_\infty$ and phrase p , we segment u 's writing of p into two sets S_0 and S_1 , of size ν and κ . We use the elements in S_0 and the tolerances derived from \mathcal{U}_l to create a biometric template and attempt to authenticate the samples in S_1 against this template—all along keeping record of how many features are missed for each sample in S_1 . This process is repeated 25 times for each passphrase p and each $u \in \mathcal{U}_\infty$ and we report what percent of the samples fail to authenticate if the system corrects up to a certain threshold of errors.

Computing Type II Errors Given a forgery of passphrase p against a user $u \in \mathcal{U}_\infty$, we use the tolerances derived from \mathcal{U}_l and each of u 's renderings of p to create a template. We then attempt to authenticate the forgery against this template and record how many features are incorrect under the forgery attempt. We then report the percentage of forgeries that successfully authenticate when correcting for up to a certain threshold of errors.

4.2 Grooming Sheep into Wolves

Our experiments were designed to illustrate the discrepancy in perceived security when considering traditional forgery paradigms and a more stringent, but realistic, security model. In particular, we assume that at the very minimum, the system will most likely be compromised by a forger who (1) attacks victims who have a writing style that the forger has a natural ability to replicate, (2) has knowledge of how biometric authentication systems operate, and (3) has a vested interest in accessing the system, and therefore is willing to devote significant effort towards these ends.

Figure 1 presents ROC curves for forgeries from impersonators with varying levels of knowledge. The plot denoted **Type II-naive*** depicts results for the traditional case of naïve or random forgeries widely used in the literature [21, 10, 40, 18]. In these cases, the impersonation attempts simply reflect taking one user's true rendering of phrase p as an impersonation attempt on the target writing p . Therefore, this classification makes no differentiation based on forger and victim's style of writing, and so may include, for example, block writers "forging" cursive writers. Arguably, such forgeries may not do as well compared to a less standard (but more reasonable) type of naïve classification (**Type II-naive**) where one only evaluates **Type II** errors for writers from similar styles. Thus, for the remaining discussions in this paper, we disregard this former classification of forgeries.

The **Type II-static** plots represent the success rate of forgers when given access to only a static rendering of the passphrase. By contrast, **Type II-dynamic** forgeries are produced after seeing (possibly many) real-time renderings of the image. One can easily consider this a realistic threat if we assume that a motivated adversary may capture the writing on camera, or more likely, may have access to data written electronically in another context. Lastly, **Type II-skilled** presents the resulting success rate of forgeries derived under our forgery model which captures a more worthy opponent. Sample forgeries from this group are shown in Figure 3 with additional examples appearing in Appendix A. Notice that when classified by writing type, the "skilled" forgers were very successful against mixed writers (Figure 2).

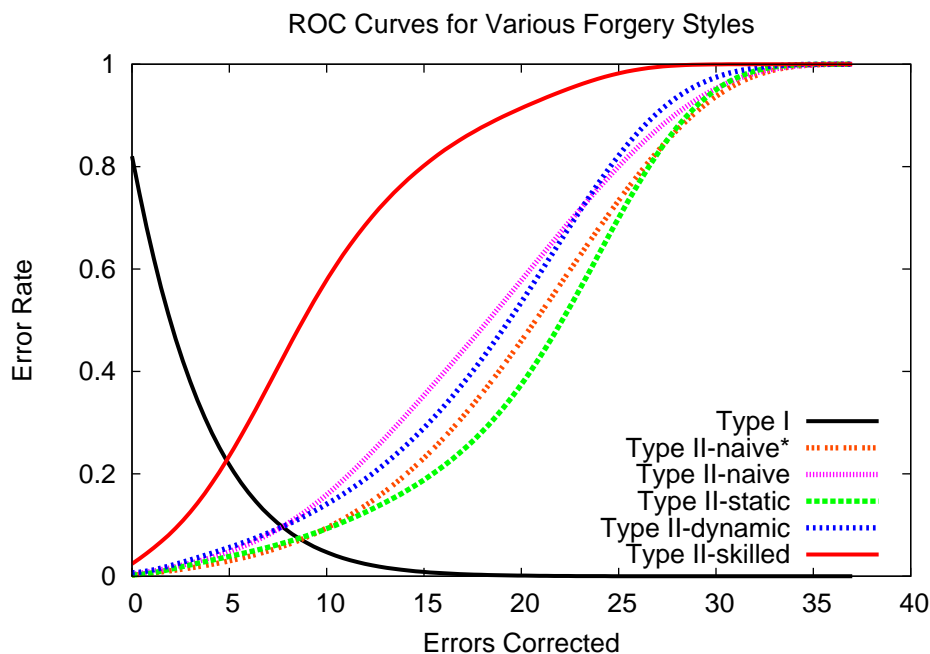


Figure 1: Overall ROC curves for naïve, static, dynamic and skilled forgers.

Intuitively, one would expect that forgers with access to dynamic and/or static representations of the target writing should be able to produce better forgeries than those produced under the naïve classification. This is not necessarily the case, as we see in Figure 1 that at some points, the naïve forgeries do better than the forgeries generated when seeing static and dynamic information. This is primarily due to the fact that the naïve classification reflects users’ normal writing (as there is really no forgery attempt here). The natural tendencies exhibited in their writings appear to produce better “forgeries” than that of static or dynamic forgers (beyond some point), who may suffer from unnatural writing characteristics as a result of focusing on the act of forging.

One of the most striking results depicted in the figures is the significant discrepancy in **Type II** error rates between standard evaluation methodologies and that of the more skilled forgeries captured under our strengthened model. While it is tempting to directly compare the results under the new model to those under the more traditional metrics (i.e., by contrasting the **Type II**-skilled error rate at the **EER** under one of the older models), such a comparison is *not* valid. This is because the forgers under the new model had already depicted some innate abilities at forging and were also more knowledgeable with respect to the intricacies of handwriting verification (see the **round III** discussion in § 3.1.2).

However, the correct comparison is to consider the **EERs** under the two models. For instance, the **EER** for this system under **Type II**-skilled forgeries is approximately 23.3% at five error corrections. However, for the more traditional metrics, one would arrive at **EERs** of

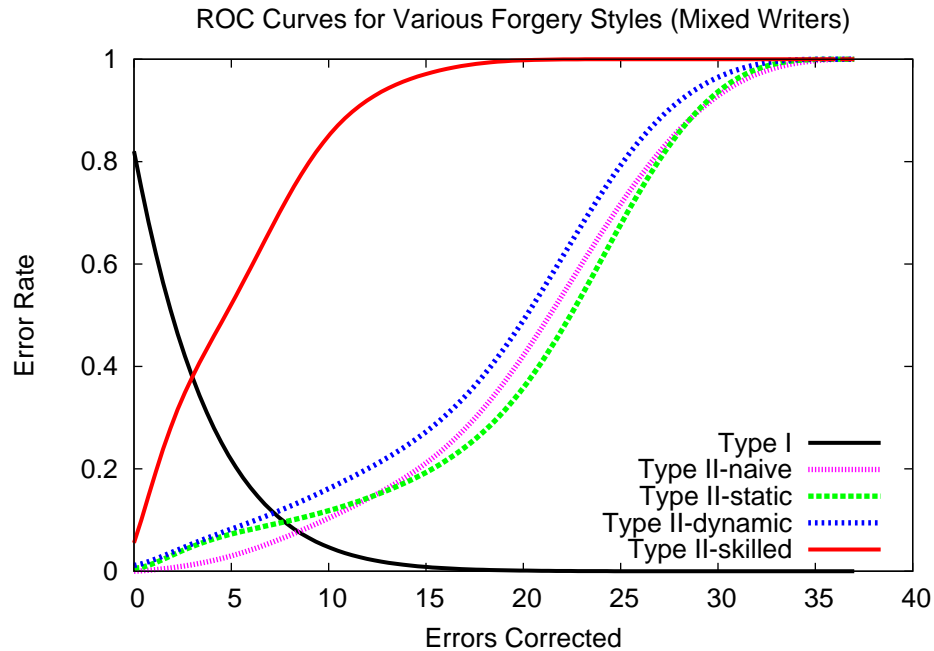


Figure 2: ROC curves against all *mixed* writers. This grouping appeared the easiest to forge by the users in our study.

7.5%, 6.2%, 5.4% under evaluations of dynamic, static and naive forgeries, respectively. Not surprisingly, such results are indeed inline with the current state of the art [21, 10, 40, 18]. Even worse, under the most widely used form of adversary considered in the literature (i.e., naive*) we see that the security of this system would be over-estimated by nearly 400%!

Forger Improvement Figure 4 should provide assurance that the increase in forgery quality is not simply a function of selecting motivated individuals from round II to participate in round III. The graph shows the improvement in Type II error rates between rounds II and III for the trained forgers. The improvement can be primarily attributed to two factors: (1) knowledge of how such systems work and (2) style-targeted victims⁶. We see that the improvement is significant, especially for the forgers who focused on mixed and block writers. Improvement of cursive forgeries was significant, although not as dramatic as the other two types. We noted that, in general, cursive writers were more difficult to forge than writers of other styles. Notice that at the EER induced by forgers with access to dynamic information (Figure 1), our trained block, cursive, and mixed forgers improved

⁶The observant reader will note that the trained forgers faced a different distribution of “easy” targets in Round III than they did in Round II. We did this so we could analyze the system at it’s weakest link. However, we found that if we normalize the results so that both rounds had the same makeup of “easy” targets, the EER presented in Figure 1 only shifts from 23.3% at five errors to 22.8% at five errors.

Target	perfect misfit
Human Forgery	perfect misfit
Target	Crisis management
Human Forgery	Crisis management

Figure 3: Examples of block forgeries from our educated forgers.

their Type II error rate by 0.33, 0.16, and 0.47, respectively. This improvement results from less than two hours of training and effort, which is likely much less than what would be exerted by a dedicated or truly skilled forger.

5 Assembly-line Wolves: Generative Models as an Evaluation Paradigm

Somewhat surprised by the results, and curious about the talent of our forgers, we decided to create a tool for a more automated method to evaluate biometric authentication systems. Indeed, we were also intrigued to learn whether our “skilled” forgers were really this good (to have such a dramatic effect on the EER), and whether we could do better using less human-intensive resources.

The reason for this should be clear once one considers the traditional approach to evaluating a proposed biometric system. Generally, evaluation involves recruiting some number of test subjects into the lab and encouraging them to try to break the system at hand. Keep in mind that the number of tests that can be performed is limited by the need to recruit human participants and collect good quality data from motivated subjects. This limitation may be a major contributing factor as to why only a handful of studies in the literature account for some form of realistic adversary. Nevertheless, the aforementioned results (§4) should make it abundantly clear that analysis under more sophisticated adversarial models must be considered.

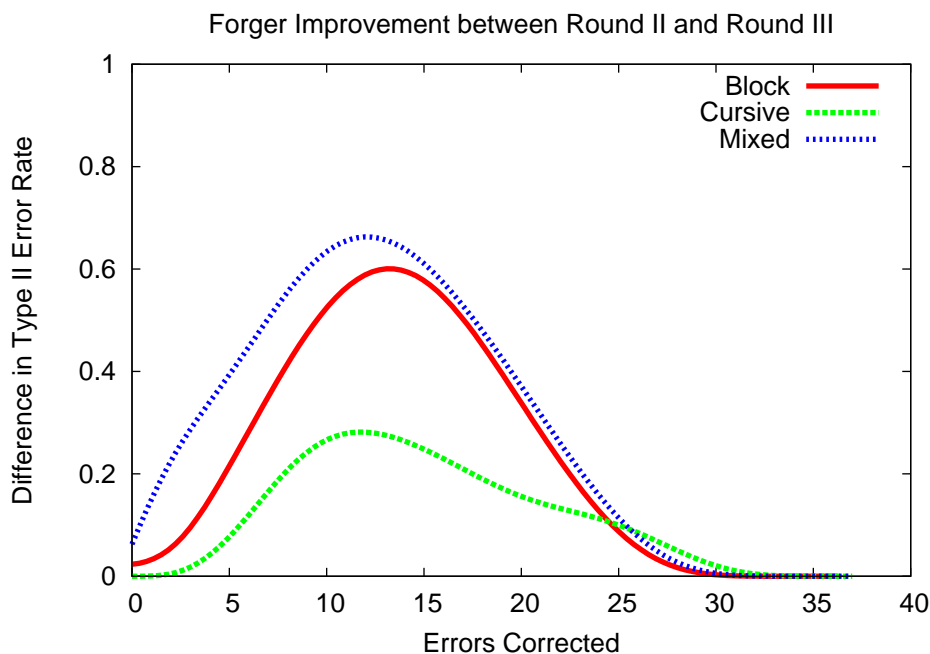


Figure 4: Forger improvement between rounds II and III.

To confront the obstacles posed by wide-scale data collection and training of good impersonators, we decided to explore the use of automated attacks using generative models as a supplementary approach for evaluating behavioral biometrics. In that regard, we were keen to see whether an automated attack, using limited writing samples from the target, could match the false accept rates observed for our skilled forgers in §4.2. In fact, we believe that such generative attacks themselves may be a far more dangerous threat that, until now, have yet to be studied in sufficient detail.

For the remaining discussion we explore a set of threats that stem from generative attacks which assume knowledge that spans the following spectrum:

- I. *General population statistics.* Such information can be gleaned in a number of ways. For example, via the open sharing of test datasets by the research community.
- II. *Statistics specific to a demographic of the targeted user.* In the case of handwriting, we assume the attacker can, for example, rely on a corpus collected from other users of a similar writing type (e.g., cursive).
- III. *Data gathered from the targeted user.* Excluding, of course, direct capture of the secret itself, one can imagine the attacker capturing copies of a user’s handwriting, either through discarded documents or by stealing a PDA, where writing unrelated to the secret is stored in plaintext.

As we explore the impact of these varying threats, a key issue that we also consider is the amount of recordings one needs to make these scenarios viable attack vectors. As we show later, the amount of data required may be surprisingly small for the case of authentication systems based on handwriting dynamics.

5.1 A generative toolkit for performance testing

One approach to synthesizing handwriting we explore here is to assemble a collection of basic units (graphemes) that can be combined in a concatenative fashion to mimic authentic handwriting. In this case, there is no underlying model of human physiology to guide the synthesis, rather, creation of the writing sample is accomplished by choosing appropriate graphemes from an inventory that may cover writing from the target user (scenario III above) as well as representative writings by other members of the population at large (scenarios I and II). The technique we apply here expands upon earlier rudimentary work [25], and is similar in flavor to approaches taken to generate synthesized speech [32] and for text-to-handwriting conversion [13].

5.1.1 Forgeries

As noted earlier, each writing sample consists of three signals parameterized by time: $x(t)$, $y(t)$ and $p(t)$. The discrete signals $x(t)$ and $y(t)$ specify the location of the pen at time t , and binary signal $p(t)$ indicates whether the pen is touching the writing surface. The goal of our generative algorithm is then to generate t , $x(t)$, $y(t)$ and $p(t)$ such that the sample is not only accepted as authentic, but also relies on acquiring as little information as possible from the target user (again, in a different security context). In particular, we assume the adversary has access to a generative corpus \mathcal{G}_u for each user u she wishes to attack, in addition to samples from users of similar writing styles \mathcal{G}_S ; where S is one of “block,” “mixed,” or “cursive.” As is the case with traditional computations of the EER we also assume that passphrase p is known.

General Knowledge Assume that the adversary wishes to forge user u with passphrase p and writing style S . Ideally, she would like to do so using a minimal amount of information directly collected from u . Fortunately, from Section 4 we see that knowledge of a user’s writing style yields a fair amount of pertinent information that can potentially be used to replicate that user’s writing. Thus, to aid in generating accurate forgeries, the adversary can make use of several statistics computed from annotated writing samples in $\mathcal{G}_S \setminus \mathcal{G}_u$. In what follows, we discuss what turns out to be some very useful measures that can likely be easily generalized for other behavioral biometrics.

Denote as $P_c(i, j, c_1, c_2)$ the probability that writers of style S render the character c_1 with j strokes and the i^{th} stroke of c_1 is connected to c_2 . Let $P_c(i, j, c_1, *)$ be the probability that these writers connect the i^{th} stroke of c_1 (again rendered with j strokes) to any adjacent letter. For example, many cursive writers will connect the first stroke of the letter ‘i’ to preceding letters; for such writers $P_c(1, 2, \text{‘i’}, *) \approx 1$. Note that in this case, the dot of the ‘i’ will be rendered after preceding letters, we call this a “delayed” stroke.

Let $\delta_w(c_1, c_2)$ denote the median space between the adjacent characters c_1 and c_2 , $\delta_w(c_1, *)$ the median distance between c_1 and any preceding character, and $\delta_w(*)$ the median distance between any two adjacent characters. Intuitively, $\delta_w(c_1, c_2) < 0$ if users tends to overlap characters and $\delta_w(c_1, c_2) \approx 0$ if u is a cursive writer. Similarly, let $\delta_t(c_1, c_2)$ denote the median time elapsed between the end of c_1 and the beginning of c_2 . Definitions of $\delta_t(c_1, *)$ and $\delta_t(*)$ are analogous to those for δ_w .

Finally, the generative algorithm clearly must also make use of a user’s pen-up velocity. This can be estimated from the population by computing the pen-up velocity for each element in \mathcal{G}_S and using the 75th percentile of these velocities. We denote this value as v_S .

Having acquired her generalized knowledge, the adversary can now select and combine her choices of n -grams that will be used for concatenative-synthesis in the following manner:

n -gram Selection At a high level, the selection of n -grams that allow for a concatenative-style rendering of p involves a search of \mathcal{G}_u for possible candidates. Let $\mathcal{G}_{u,p}$ be a set of u ’s renderings of various n -grams in p . Notice that there may be more than one element in $\mathcal{G}_{u,p}$ for each n -gram in p . The attacker then randomly ⁷ selects k renderings g_1, \dots, g_k from $\mathcal{G}_{u,p}$, such that the strings (L_i) corresponding to each g_i form p , i.e., $L_1 || L_2 || \dots || L_k = p$. We ensure that g_i and g_{i+1} do not originate from the same writing sample ⁸.

n -gram Combination Now, given the selection of n -grams (g_1, \dots, g_k) the attacker’s task is to combine them to form a good representation of p . Namely, she must adjust the signals that compose each g_i ($t_{g_i}, x_{g_i}(t_{g_i}), y_{g_i}(t_{g_i})$ and $p_{g_i}(t_{g_i})$) to create a final set of signals that authenticates to the system. The algorithm is quite simple. At a high level, it proceeds as follows: The adversary normalizes the signals $t_{g_i}, x_{g_i}(t_{g_i})$ and $y_{g_i}(t_{g_i})$ by subtracting the respective minimum values from each element in the signal. The $y_{g_i}(t_{g_i})$ are shifted so that the baselines match across g_i . To finalize the spatial transforms, the adversary horizontally shifts each g_i ; define the shift for $g_i, i > 1$ by the recurrence

$$\delta_{x,i} = \delta_{x,i-1} + \max(x_{g_{i-1}}(t_{g_{i-1}})) + \delta_w(e_{i-1}, s_i)$$

where e_i (resp. s_i) is the last (resp. first) character in g_i and $\delta_{x,1} = 0$. Once the adversary has fixed the (x, y) coordinates, she needs to fabricate t and $p(t)$ signals to complete the forgery. Modifying $p(t)$ consists of deciding whether or not to connect adjacent grams. To do this, the adversary uses knowledge derived from the population. If e_{i-1} is rendered with j' strokes, and g_i starts with s_i , the adversary connects the j^{th} stroke of e_{i-1} to s_i with probability $P_c(j, j', e_{i-1}, s_i)$. To generate a more realistic connection, the adversary smooths the last points of e_{i-1} and the first points of s_i . Additionally, all strokes that occur after stroke j are “pushed” onto a stack, which is emptied on the next generated pen-up. This

⁷Our algorithm is not completely random, it is biased towards representations of longer n -grams. However, in practice the average length of each n -gram is rather small as shorter n -grams are required to fill in the gaps between larger n -grams.

⁸We impose this restriction so we may demonstrate the feasibility of a generative attack. An actual adversary could benefit by using graphemes from the same sample as the dynamics could be more representative than those created by a generative approach.

behavior simulates a true cursive writer returning to dot ‘i’s and cross ‘t’s at the end of a word, processing characters closest to the end of the word first.

Adjusting the t signal is also straightforward. Let T be the time in $t_{g_{i-1}}$ that the first delayed stroke in e_{i-1} starts. If there are no delayed strokes in e_{i-1} , $T = \max(t_{g_{i-1}})$. Then, the adversary can simply shift t_{g_i} , $i > 1$ by $\delta_{\tau,i}$ where

$$\delta_{\tau,i} = \delta_{\tau,i-1} + T + \delta_t(e_{i-1}, s_i)$$

and $\delta_{\tau,1} = 0$.

The only other time shift occurs when delayed strokes are popped from the stack. Note that it is difficult to determine the time shift for such strokes simply by using the time as it occurred in the original context, as this is *dependent* on the number of letters in the original word and where this stroke appeared. However, one can make use of global knowledge to estimate the time delay by using v_S and the distance between the end of the previous stroke and the new stroke. Note that it is beneficial to take v_S as the 75th percentile instead of the median velocity because, for cursive writers in particular, the majority of pen-up velocities is dominated by the time between words. However, these velocities are intuitively slower as the writer is now thinking about creating a new series of grams as opposed to finishing grams that already exist.

It is also of interests to take note that if the adversary does not have access to the statistical measure $\delta_w(e_{i-1}, s_i)$, she can first base her estimate of inter-character spacing on $\delta_w(e_{i-1}, *)$, and then on $\delta_w(*)$. She proceeds similarly for the measures δ_t and P_c .

5.1.2 Results

To evaluate this concatenative approach we analyzed the quality of the generated forgeries on user u writing passphrase p . However, rather than using all the available samples from the generative corpus, we instead choose 15 samples at random from $\mathcal{G}_{u,p}$ — with the one restriction being that there must exist at least one instance of each character in p among the 15 samples. For simplicity, let’s denote this limited corpus as $\mathcal{G}'_{u,p}$. Recall that this generative corpus contains writing samples from u , but collected in a different context — i.e., did not contain writings of p . The attackers choice of n -grams g_1, \dots, g_k are chosen from this restricted set.

Additionally, we limit \mathcal{G}_S to contain only 15 randomly chosen samples from each user with a similar writing style as u . Let’s denote this set of writings as \mathcal{G}'_S . Note that we *purposefully* chose to use small (and arguably, easily obtainable) datasets to illustrate the power of this concatenative attack. Our “general knowledge” statistics are computed from \mathcal{G}'_S . An example forgery derived by this process is shown in Figure 5. Additional examples are given in Appendix A.

We generated 25 forgeries and used each as an attempt to authenticate to the biometric template corresponding to user u under passphrase p . Figure 6 depicts the average across all 25 forgery attempts. As a baseline for comparison, we replot the **Type I** and **Type II**-skilled plots from Section 4. The **Type II**-generative plot shows the results of the generative algorithm against the entire population. Observe that under these forgeries the system

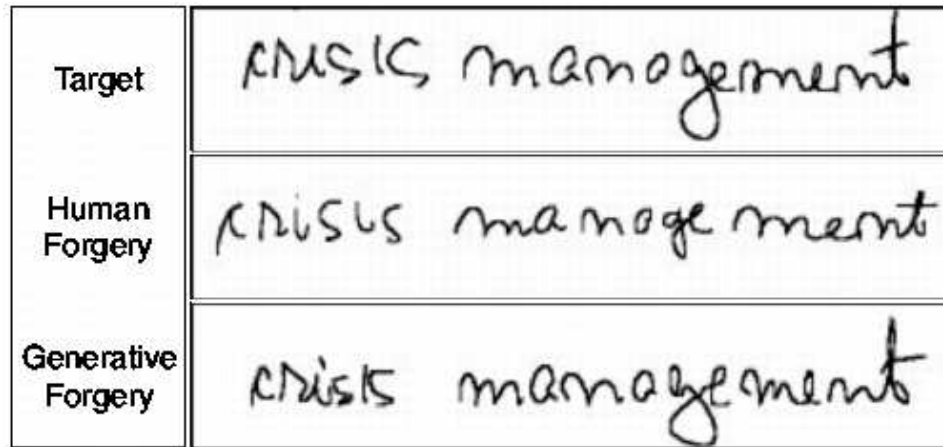


Figure 5: An example of mixed forgeries. The second rendering is a human-generated forgery of the first, and the third was created by our generative algorithm.

achieves an EER of 26% at four error correction compared to an EER of 23% at five error corrections when considering our “skilled” forgers.

Discussion We note that on average each generative attempt only used information from 6.67 of the target user’s writing samples. Moreover, the average length of an n -gram was 1.64 (and was never greater than 4). More importantly, as we make no attempt to filter the output of the generative algorithm by rank-ordering the best forgeries, the results could be much improved. That said, we believe that given the limited information assumed here, the results of this generative attack on accuracy of the system is alarming. Furthermore, we believe that this attack is feasible because annotation of the samples in $\mathcal{G}_{u,p}$, while tedious, poses only a minor barrier to any determined adversary. For instance, in our case annotation was accomplished with the aide of a annotation tool that we implemented which is fairly automated, especially for block handwriting: taking $\approx 30s$ to annotate block phrases and $\approx 1.5 mins$ for cursive phrases.

6 Other Related Work

There is, of course, a vast body of past work on the topic of signature verification (see [36] for a comprehensive if somewhat dated survey, [18] for a more up-to-date look at the field, and [39] for an extensive online listing of publications). However, to the best of our knowledge, there is relatively little work that encompass our goals and attack models described herein.

Perhaps the work closest to ours, although it predominately involves signatures, is that by Vielhauer and Steinmetz [44]. They use 50 features extracted from a handwriting sample to construct a biometric hash. While they performed some preliminary testing on PIN’s and passphrases, the bulk of their study is on signatures, where they evaluated features based

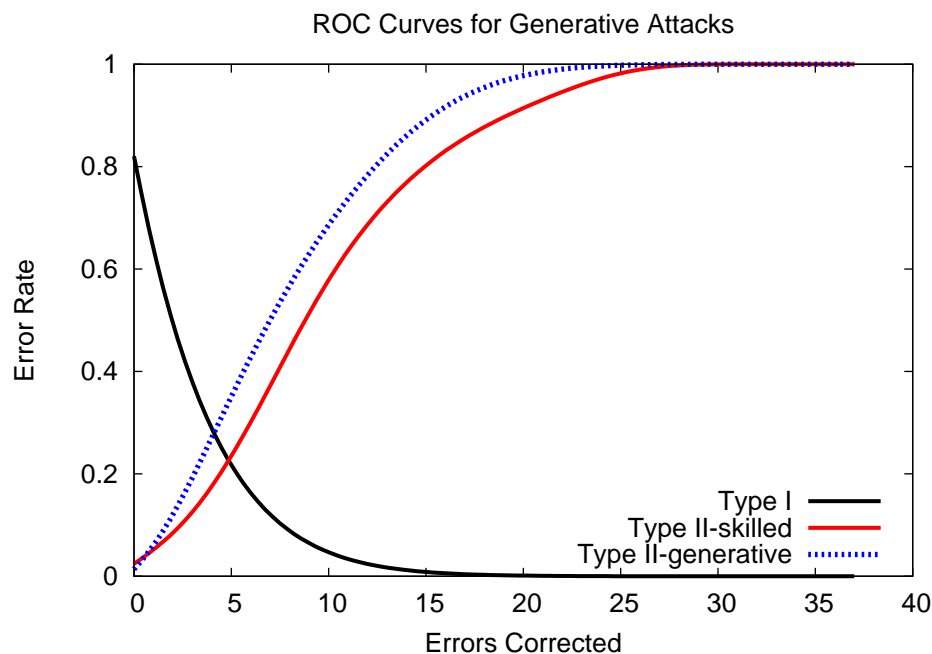


Figure 6: ROC curves for generative forgeries. Observe that even using limited information from the target and population-wide derived statistics, the forgeries out-perform that of our “skilled” forgers—shifting the EER from 23% at five errors to 26% at four errors.

on intrapersonal deviation, interpersonal entropy with respect to their hash function, and the correlation between these two values. That work however does not report any results for meaningful attempts at forgery (i.e, other than naïve attacks).

Also germane are a series of recent papers that have started to examine the use of dynamic handwriting for the generation of cryptographic keys[11, 20]. 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 [11]. They claim an EER of 8% and mention that their test database contains forgeries, but provide no details on how these were produced or their quality.

Kuan, et al. present a method based on block-cipher principles to yield cryptographic keys from dynamic signatures [20]. They test their algorithm on the standard dataset from the *First International Signature Verification Competition* and report ERRs of between 6% and 14% if the forger has access to a stolen token. The production of skilled forgeries in the SVC dataset resembles the methodology we have used in our studies [49].

Also of interest is the work of Brömme and Al-Zubi who present a scheme for authentication based on simple graphical sketches, which are described textually and the user must create (or re-create) [3]. With knowledge of the sketch – presumably via seeing a static

image – forgers were successful at an EER of 7.2%, but this experiment only involved one sketch from a single subject and two impostors. Additionally, while the order in which the components of the sketch are drawn presents a barrier for an attacker, it may also prove difficult for the user to remember.

In the realm of signature verification we also note work on an attack based on hill-climbing, but that makes the assumption that the system reveals how close of a match the input is [48]. We believe this to be clearly unrealistic, and our attack models are chosen to be more pragmatic than this. Finally, there have been a handful of works on using generative models to attack biometric authentication. However, we note there exists significant disagreement in the literature concerning the potential effectiveness of similar (but inherently simpler) attacks on speaker verification systems (e.g., [35, 32]). Lindberg and Blomberg, for example, determined that synthesized passphrases were not effective in their small-scale experiments [24], whereas Masuko et al. found that their system was easily defeated [28].

7 Conclusions

Several fundamental computer security mechanisms rest on the ability of an intended user to generate an input that an attacker is unable to reproduce (sufficiently accurately). In the biometric community, the security of biometric-based technologies hinges on this perceived inability of the attacker to reproduce the target user’s input. In particular, the evaluation of biometric technologies is usually conducted under fairly weak adversarial conditions. Unfortunately, this practice may be significantly underestimating the real risk of accepting forgeries (for a particular system) as authentic. To directly address this limitation we present an automated technique for deploying generative attacks that assists in the evaluation of biometric systems. We show that our generative attacks match or exceed the effectiveness of forgeries rendered by skilled humans.

8 Acknowledgments

The authors would like to thank Dishant Patel and Carolyn Buckley for their help in our data collection efforts. We also thank the numerous people who provided us with handwriting samples, many of whom devoted hours of their time to this work. The work is supported by NSF grant CNS-0430338.

A Example Forgeries

The phrases in our study were chosen as they were easy to remember and reasonable in length. The phrases were “crisis management”, “graphic language”, “least favorite”, “perfect misfit”, and “solo concert”. Signatures were not used due to privacy concerns and strict restrictions on research involving human-subjects. More importantly, in the context of key-generation, signatures are not a good choice for hand-writing biometric as the compromise

of keying material could prevent a user from accessing the system thereafter. Figures 7, 8, 9 and 10 provide some examples of both human and computer-generated forgeries.

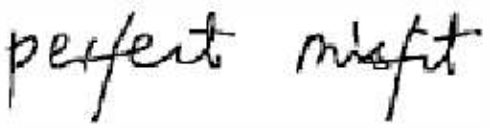

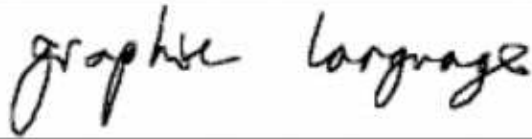
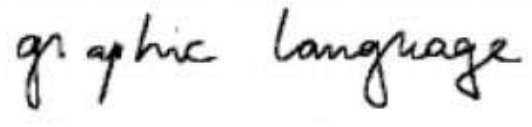
Target	
Human Forgery	
Target	
Human Forgery	

Figure 7: Examples of mixed forgeries from our trained forgers.

B Features

In Table 1 we present the resulting 37 features from our analysis of 144 possible features. We make use of the following notation: θ is the angle of a given stroke, v , v_x , v_y denote velocity, velocity in the horizontal direction, and velocity in the vertical direction, respectively.

References

- [1] J. Bentley and Colin Mallows. How much assurance does a PIN provide? In proceedings of Human Interactive Proofs (HIPS), Second International Workshop, LNCS 3517, H.S. Baird and D. P. Lopresti (Eds), 2005.
- [2] The Biometrics Consortium. <http://www.biometrics.org/>.
- [3] A. Brömme and S. Al-Zubi Multifactor Biometric Sketch Authentication. In *Proceedings of the First Conference on Biometrics and Electronic Signatures of the GI Working Group BIOSIG*, Darmstadt, Germany, pp. 81-90, July 2003.
- [4] Y. Chang and W. Zhang and T. Chen. Biometric-based cryptographic key generation. In Proceedings of IEEE Conference on Multimedia and Expo, June 2004.
- [5] G. I. Davida, Y. Frankel, and B. J. Matt. On enabling secure applications through off-line biometric identification. In *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, pages 148-157, May 1998.

Feature	Description
# of strokes	Number of strokes used to render the passphrase[44].
# of extrema	Number of local extrema in the horizontal and vertical directions[44].
Writing width	Total width of the writing[44, 12].
Writing height	Total height of the writing[44, 12].
Pen-down distance	Total distance travelled by the pen-tip while touching the screen[12].
Pen-up distance	Euclidean distance between pen-up and pen-down events.
Inv. Mom. 00	$\sum_x \sum_y f(x, y)$; $f(x, y) = 1$ if there is a point at (x, y) and 0 otherwise[12].
Inv. Mom. 10	$\sum_x \sum_y f(x, y) \cdot x$. Measures the horizontal mass of the writing[12].
Inv. Mom. 01	$\sum_x \sum_y f(x, y) \cdot y$. Measures the vertical mass of the writing[12].
Inv. Mom. 11	$\sum_x \sum_y f(x, y) \cdot xy$. Measures diagonality of the writing sample[12].
Inv. Mom. 12	$\sum_x \sum_y f(x, y) \cdot xy^2$. Measures horizontal divergence[12].
Inv. Mom. 21	$\sum_x \sum_y f(x, y) \cdot x^2y$. Measures vertical divergence[12].
Loop area	Total area enclosed within loops generated by overlapping strokes[12].
Loop y centroid	The average value of all y coordinates contained within writing loops[12].
Upper zone	Distance between upper-baseline and topline of the writing[27].
Lower zone	Distance between baseline and bottomline of the writing[27].
X-Area	Integrated area to the left of the writing[44].
Y-Area	Integrated area beneath the writing[44].
Median θ	Median stroke-slant, normalized to $\theta \in [0, \pi]$ [27].
Horiz. end dist.	Distance between the last x -coordinate and maximum x -coordinate[22].
Vert. end dist.	Distance between the last y -coordinate and maximum y -coordinate[22].
Time	Total time spent writing (measured in ms)[22].
Pen up/down ratio	Ratio time spent with the pen off and on the tablet[22].
# of samples	Number of samples recorded by the tablet[22].
Median pen velocity	Median speed of the pen-tip[22].
Time of max vel.	Time of the maximum pen-velocity[22].
Time of min v_x	Time of the minimum pen-velocity in the horizontal direction[22].
Time of max v_x	Time of the maximum pen-velocity in the horizontal direction[22].
Time of min v_y	Time of the minimum pen-velocity in the vertical direction[22].
Time of max v_y	Time of the maximum pen-velocity in the vertical direction[22].
Time of max θ	Time of maximum stroke slant.
# of times $v_x = 0$	Number of times the pen ceases to move horizontally[22].
# of times $v_y = 0$	Number of times the pen ceases to move vertically[22].
Duration $v_x > 0$	Total time the pen spends moving to the right[22].
Duration $v_x < 0$	Total time the pen spends moving to the left[22].
Duration $v_y > 0$	Total time the pen spends moving to the up[22].
Duration $v_y < 0$	Total time the pen spends moving to the down[22].

Table 1: The statistical features used to evaluate the biometric authentication system. Features were chosen based on the quality score Q defined in §3.1.1.

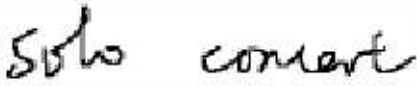


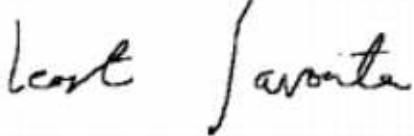
Target	
Human Forgery	
Target	
Human Forgery	

Figure 8: Examples of cursive forgeries from our trained forgers.

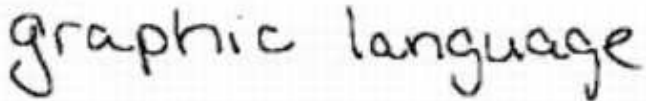
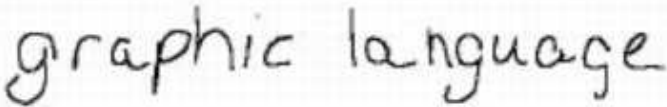
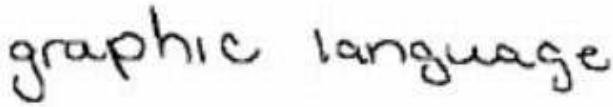
Target	
Human Forgery	
Generative Forgery	

Figure 9: An example of block forgeries. The second rendering is a human-generated forgery of the first, and the third was created by our generative algorithm.

[6] D. Davis, F. Monrose, and M. Reiter. On user choice in graphical password systems. In *Proceedings of the 13th USENIX Security Symposium*, San Diego, August, 2004.

[7] G. R. Doddington, W. Liggett, A. F. Martin, M. Przybocki, and D. A. Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition eval-

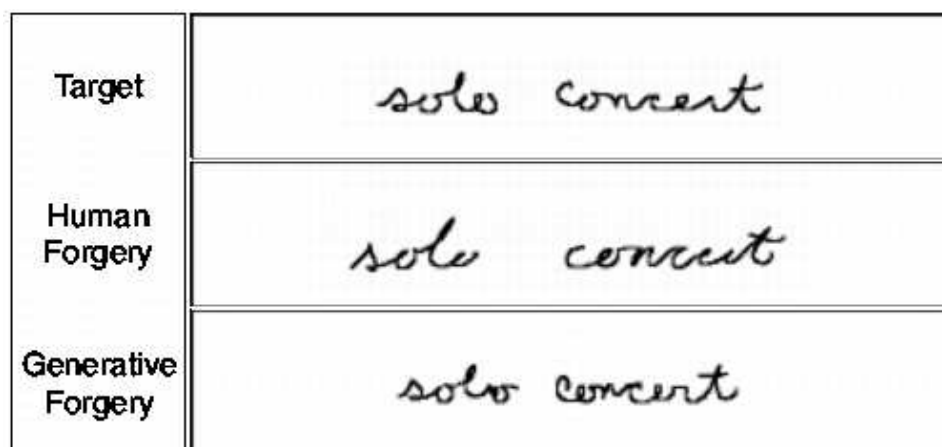


Figure 10: An example of cursive forgeries. The second rendering is a human-generated forgery of the first, and the third was created by our generative algorithm.

uation. In *Proceedings of the 5th International Conference on Spoken Language Processing*, November 1998.

- [8] S.J. Elliot. Development of a Biometric Testing Protocol for Dynamic Signature Verification. In *Proceedings of ICARCV*, 2002.
- [9] M. C. Fairhurst. Signature verification revisited: promoting practical exploitation of biometric technology. *Electronics & Communication Engineering Journal*, pp. 273-280, December 1997.
- [10] J. Fierrez-Aguilar, L. Nanni, J. Lopez-Peñalba, J. Ortega-Garcia, and D. Maltoni. An on-line signature verification system based on fusion of local and global information. In *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*, LNCS vol. 3546, pp. 523-532, 2005.
- [11] H. Feng and C. Wah. Private Key generation from on-line handwritten signatures. In *Information Management and Computer Security*, Vol. 10, No.4, pp. 159-164, 2002.
- [12] R. Guest. The Repeatability of Signatures. In *Proceedings of the IEEE 9th International Workshop on Frontiers in Handwriting Recognition*, pages 492-497 August, 2004.
- [13] I. Guyon. Handwriting synthesis from handwritten glyphs. In *Proceedings of the Fifth International Workshop on Frontiers of Handwriting Recognition*, Colchester, England, 1996.
- [14] A. Goh and D. Ngo. Computation of Cryptographic Keys from Face Biometrics. In *Proceedings of CMS 2003*, LNCS 2828, pages 1-13, 2003.
- [15] C. Hertel and H. Bunke. A set of novel features for writer identification. In *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*, Springer-Verlag Lecture Notes in Computer Science vol. 2688, pp. 679-687, 2003.
- [16] D. Hamilton, J. Whelan, A. McLaren and I MacIntyre. Low Cost Dynamic Signature Verification System. In *Proceedings of European Convention on Security and Detention*, pages 202-206, London, 1995.

- [17] K. Huang and H. Yan. Stability and style-variation modeling for on-line signature verification. In *Pattern Recognition*, vol. 36, pp. 2,253–2,270, 2003.
- [18] A. K. Jain, F. D. Griess, and S. D. Connell. On-line signature verification. In *Pattern Recognition*, vol. 35, no. 12, pp. 2,963–2,972.
- [19] Y. Dodis and L. Reyzin and A. Smith. Fuzzy Extractors: How to generate strong keys from biometrics and other noisy data. IN Proceedings of Advances in Cryptology, EUROCRYPT, pages 523–540, 2004.
- [20] W. Kuan, A. Goh, D. Ngo, A. Teoh. Cryptographic Keys from Dynamic Hand-Signatures with Biometric Secrecy Preservation and Replaceability. In *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID)*, pp 27-32, 2005.
- [21] F. Leclerc and R. Plamondon. Automatic signature verification: the state of the art, 1989–1993. In *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 8, no. 3, pp. 643-660, 1994.
- [22] L. Lee, T. Berger, and E. Aviczer. Reliable On-Line Human Signature Verification Systems. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):643–647, June, 1996.
- [23] M. Lee, D. P. Lopresti and J. P. Olive. A text-to-speech platform for variable length optimal unit searching using perceptual cost functions. In *International Journal of Speech Technology*, vol. 6, no. 4, pp. 347–356, October 2003.
- [24] J. Lindberg and M. Blomberg. Vulnerability in Speaker Verification - A Study of Technical Impostor Techniques. In *Proceedings of the European Conference on Speech Communication and Technology*, Budapest, Hungary, vol.3, pages 1211-1214, September 1999.
- [25] 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*, volume 3546, LNCS, pages 1090-1099, 2005.
- [26] A. J. Mansfield and J. L. Wayman. Best Practices in Testing and Reporting Performance of Biometric Devices. Centre for Mathematics and Scientific Computing, National Physical Laboratory, NPL Report CMSC 14/02, August, 2002.
- [27] U.-V. Marti, R. Messerli, H. Bunke, Writer Identification Using Text Line Based Features, In *Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR'01)*, pages 101–105, September, 2001.
- [28] T. Masuko, K. Tokuda and T. Kobayashi. Imposture[sic] using synthetic speech against speaker verification based on spectrum and pitch. In *Proceedings of the International Conference on Spoken Language Processing*, Beijing, China, vol.3, pages 302-305, October 2000.
- [29] M. Mingming and W. Wijesoma. Automatic On-Line Signature Verification Based on Multiple Models. In *Proceedings of IEEE Conference on Computational Intelligence in Financial Engineering*, pp 30-33, 2000.
- [30] N. Mohankrishnan, W. Lee and M. Paulik. Improved Segmentation through Dynamic Time Warping for Signature Verification using a Neural Network Classifier. In *Proceedings of IEEE Signal Processing Society International Conference on Image Processing*, pp 929–933, 1998.
- [31] F. Monrose, M. K. Reiter, Q. Li and S. Wetzel. Cryptographic key generation from voice (extended abstract). In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pages 12–25, May 2001.
- [32] F. Monrose, M. K. 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*, August, pages 283–296, 2002.
- [33] I. Nakanishi, H. Sakamoto, Y. Itoh, and Y. Fukui. Optimal user weighting fusion in DWT domain on-line signature verification. In *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*, LNCS vol. 3546, pp. 758-766, 2005.
- [34] W. Nelson and Eyal Kishon. Use of Dynamic Features for Signature Verification. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 1504–1510, October, 1991.

- [35] B. L. Pellom and J. H. L. Hansen. An experimental study of speaker verification sensitivity to computer voice altered imposters. In *Proceedings of the 1999 International Conference on Acoustics, Speech, and Signal Processing*, March 1999.
- [36] R. Plamondon (ed.). *Progress in Automatic Signature Verification*. World Scientific, 1994.
- [37] R. Plamondon, D. P. Lopresti, L. R. B. Schomaker, and R. Srihari. Online handwriting recognition. In *Wiley Encyclopedia of Electrical and Electronics Engineering*, John Wiley & Sons, Inc., 1999, pp. 123–146.
- [38] R. Plamondon and G. Lorette. Automatic signature verification and writer identification—the start of the art. In *Pattern Recognition*, vol. 22, no.2, pp. 107-131, 1989.
- [39] K. Price. Bibliography: On-Line Signatures. January 2006. <http://iris.usc.edu/Vision-Notes/bibliography/char1011.html>.
- [40] R. Plamondon and S.N. Srihari. On-line and off-line handwriting recognition: a comprehensive survey. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no.1, pp 63-84, 2000.
- [41] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B.V.K. Vijaya Kumar. Biometric encryption™ using image processing. In *Optical Security and Counterfeit Deterrence Techniques II* (Proc. SPIE 3314), pages 178–188, 1998.
- [42] U. Uludag and A. K. Jain. Fingerprint Minutiae Attack System. The Biometric Consortium Conference, Sept, 2004.
- [43] U. Uludag, S. Pankanti, S. Prabhakar, A.K. Jain. Biometric Cryptosystems: Issues and Challenges. In *Proceedings of IEEE Special Issue on Multimedia Security of Digital Rights Management*, vol. 92, no.6, pp.948–960, 2004.
- [44] C. Vielhauer and R. Steinmetz. Handwriting: Feature Correlation Analysis for Biometric Hashes. In *Journal of Applied Signal Processing*, vol. 4, pages 542-558, 2004.
- [45] C. Vielhauer, R. Steinmetz, and A. Mayerhöfer. Biometric hash based on statistical features of online signatures. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR)*. Quebec City, Canada, pp. 123–126, August 2002.
- [46] J. Wang, C. Wu, Y.-Q. Xu, H.-Y. Shum, and L. Ji. Learning-based cursive handwriting synthesis. In *Proceedings of the Eighth International Workshop on Frontiers of Handwriting Recognition*, Ontario, Canada, 2002.
- [47] B. Yanikoğlu and A. Kholmatov. An improved decision criterion for genuine/forgery classification in on-line signature verification. In *Proceedings of the International Conference on Artificial Neural Networks*, June 2003.
- [48] Y. Yamazaki, A. Nakashima, K. Tasaka, and N. Komatsu. A study on vulnerability in on-line writer verification system. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pp. 640-644, August, 2005.
- [49] D.Y. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, G. Rigoll. SVC2004: First International Signature Verification Competition. In *Proceedings of the International Conference on Biometric Authentication (ICBA)*, Hong Kong, 15-17 July 2004.
- [50] F. Zöbisch and C. Vielhauer. A test tool to support brut[ic]-force online and offline signature forgery tests on mobile devices. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*. pp. III 225–228, 2003.