

DynaHand: Observation-Resistant Recognition-Based Web Authentication

Karen Renaud and Elin Skjegstad Olsen

Abstract—Authentication in a web environment is severely constrained by a minimal expectation of infrastructure made up of software, hardware and operator expertise. The traditional mechanism, the password, is outliving its usefulness in the web arena. This paper presents results from a field test of a graphical authentication system called DynaHand, which utilises biometrics i.e. human handwriting recognition in a graphical authentication mechanism. We also present a tool which supports the analysis of errors evidenced during the authentication process, making it possible to classify failed attempts as either due to genuine user error or attempted intrusions. In the case of the former, the tool helps to reduce the occurrence of such genuine errors by identifying and eliminating distractor images from the challenge set that could potentially cause confusion due to their similarity to target images.

Index Terms—Handwriting recognition; biometric; graphical authentication; web

I. INTRODUCTION

In a world increasingly controlled by computer applications, the man (or woman) in the street often has to enter alphanumerical access codes (passwords) in order to gain access to physical and interactive spaces. (S)he has to remember multiple access codes, and because of human memory limitations will either choose weak passwords, use the same password for all systems, or write the passwords down [5]. This defensive behaviour undermines the password mechanism and impacts negatively on the overall security of the system since an intruder can often gain access to many accounts by means of one particular account protected by a weak password. The problem is not the *password* per se, but rather the sheer numbers of passwords that people have to remember. It places an unrealistic demand on fallible human memory.

Researchers have been testing various alternatives to passwords to alleviate these unrealistic demands, and many of these are based on the well-known fact that it is easier for people to recognise than to recall. Hence alternative systems will often ask users to recognise their code and point it out, rather than generate their code without any cues, as is required for a password-based authentication process. Many recognition-based authentication will make use of images, since people tend to remember pictures more successfully than words.

The web authentication arena is severely constrained by the fact that the mechanism needs to work on any computer and for any user, regardless of disabilities and exper-

tise. Passwords operate within these constraints but the rising numbers of passwords that people have to remember impose a far greater load on computer users than is reasonable. Whereas physiological biometrics offer a viable alternative to the more traditional password and PIN, the constraints of the web environment make this option untenable, at least for the next few years. Hence we have to find other options that operate within the constrained environment but which also address the memorability issues. Graphical authentication mechanisms offer some advantages over traditional password mechanisms especially in a web environment. Alternative systems are a relatively new innovation and, as such, have many open issues. This paper will present one particular kind of image-based authentication system that addresses some of the problems with these mechanisms.

Section II discusses issues which make graphical authentication more or less accessible. Section III identifies the particular problems that occur in making graphical authentication mechanisms as secure as password-based systems. The DynaHand graphical authentication system is discussed in Section IV. Section V presents the similarity analysis tool that can be used to improve the security of these image-based systems. Section VI discusses the evaluation data. Section VII once again considers question of the security of DynaHand in the web environment, based on the evaluation of the mechanism. Section VIII concludes.

II. ACCESSIBLE WEB AUTHENTICATION

Passwords work on the basic principle that if the user possesses the same secret knowledge at authentication that was supplied at enrolment, then the system can assume that the person is who the person claims he or she is. This assumption is irretrievably flawed, since an intruder can easily obtain the knowledge and masquerade as the legitimate user because *factual knowledge is an extremely weak authenticator*. Furthermore, it is well nigh impossible for users to remember *all* the secret facts they are required to remember.

There is no question that users, as a coping mechanism, write down their passwords [11] and finding a more secure alternative is obviously desirable. What is needed is an authentication system which does not rely on a person's ability to remember random codes, but which relies on something the user is unable to forget, such as a biometric. The standard biometric is the finger-print, or the less popular and more expensive iris-scan — neither of which has been widely used so far on the Web since they require

Karen Renaud is with the Department of Computing Science, University of Glasgow. e-mail: karen@dcs.gla.ac.uk

Elin Olsen has just completed a Masters degree at the Department of Computing Science, University of Glasgow. e-mail: elin_osol@hotmail.com

non-standard equipment.

A serious alternative to finger-print readers and iris scanners is a graphical authentication mechanism that relies on the user's own recognition process rather than a biometric reader. This kind of authentication can be performed using a standard browser, which eliminates the need for additional equipment. These systems have a superior solution to memorability problems compared to entering alphanumeric passwords as people have an innate ability to remember pictures better than words [4]. Renaud and De Angeli [5] classified such systems into various categories. Two of these, location-based and recognition-based systems, also called *locimetric* and *cognometric* systems, respectively, have potential in a web environment. The former relies on the user selecting, in sequence, a number of positions within the on-screen image. The latter relies on the user recognising and identifying target images from a set of displayed images. The displayed images could be photos of faces [1], representational objects such as those used by VIP [4] or abstract pictures such as the ones used by Dèjá Vu [6].

Although graphical recognition-based mechanisms support access codes that are more memorable than those supported by recall-based mechanisms, they offer no other cues to the forgetful user. The provision of cues is always a tricky issue, since cues need to be helpful to the legitimate user but not to the intruder. One therefore has to try to provide cues that will make sense only to the legitimate user. Weinshall [20] proposes an ingenious scheme whereby a cue is provided by a slight change in the image being displayed which the user has been trained to observe. The theory is that an imposter's change-blindness [17] will prevent him or her from seeing the slight changes and thus the cue will only be observed by the legitimate user. Unfortunately the changes to the images have to be fairly slight in order to go unobserved by an imposter and so this scheme will not work very well for users with failing sight or attentional difficulties.

Another way to supply cues within an image is to use non-representational images based on a biometric such as the user's handwriting. Back in 1895, Preyer [13] found that a person would produce the same handwriting using either of their hands or feet and proved that handwriting was not determined by the appendage used to produce it but rather by some other process in the brain. He called handwriting "brainwriting" since it was created automatically and impulsively, without any conscious thought or awareness of the formation of the letters or numerals. In more recent publications, Longcamp *et al.* [10] argues that we recognise our own handwriting not just visually but also because it is related to the learnt process of writing the letters and numerals, something referred to as "kinesthetic facilitation" [18], which is combined with the visual control process [21]. A direct consequence of this is that people recognise their own handwriting, even many years after they have produced it.

Handwriting recognition is a skill that is completely effortless and, like many other implicitly learnt skills, it does

not degrade with age or time. Heckman *et al.* [8] carried out experiments with stroke and dementia patients and concluded that handwriting recognition was a special skill which was independent of verbal and lexical tasks. Srihari *et al.* [19] found that in 98% of 1000 cases, the handwriting was clearly individual and this makes it particularly suitable for use by a recognition-based authentication scheme. Furthermore, most importantly in terms of our proposed use of this skill during authentication, is that people cannot formulate the way they do this — the knowledge is tacit, not factual, and this reduces the potential for transfer of the knowledge to an intruder.

Handwritten signatures have been used to authenticate humans for many years, and some shops are still accepting this form of authentication when customers pay for purchases with a bank card. A signature can, however, easily be recognised by people other than the signature owner, and something less universally-recognisable is needed in order for images of the user's handwriting to be used as a cue in online authentication. Srihari *et al.* [19] determined that words had more individuality (more difference) than alphanumeric characters, and that characters had more individuality than numerals. Individuality, in this sense, refers to traits which make it possible to identify the author of a handwriting sample. It is harder for another person to deduce the authorship of a numeral sample than of a letter or word sample. Due to the inherent human handwriting recognition ability, and the need to make the system as secure as possible, the use of numerals in recognition-based authentication is the only viable option. The cue the image provides is of limited use to intruders while still providing a meaningful and helpful cue to the legitimate user.

Renaud [15] describes a system called HandWing, which utilises handwriting recognition in a graphical authentication mechanism which is specifically tailored to the needs of sites requiring a weak authentication mechanism. This system requires the user to recognise a 5-digit code, a postal-code and their own hand-drawn doodle. Together they make up a 3 stage authentication process, where at least 9 distractors are displayed together with the target image at each stage. The site is used on a continuous basis by a group of older users, and the results show that users authenticate themselves successfully with very few failures. The first stage in the HandWing system uses fixed PINs, which allows users to recognise the PIN making in the image and to use the handwriting as a cue.

Whilst the system is very successful in minimising false rejects it does not conclusively prove the efficacy of pure numeral recognition in an authentication setting since it is entirely possible that the users are merely remembering their PINs and not using the handwriting cue. In order to test the viability of pure handwriting recognition, an authentication mechanism called DynaHand was developed and this is described in Section IV. The next section considers the issue of graphical authentication security.

Security specialists have widely-accepted mechanisms at their disposal for quantifying, from a purely technical perspective, the strength of traditional authentication mechanisms [12]. It is harder to quantify the strength of image-based authentication mechanisms.

The quality of the images used, and the manner in which they are used, will either make or break the authentication mechanism from a usability perspective [14] but the metrics associated with image choice from a security perspective remains an open question. Investigation into the security of alternative authentication mechanisms is thus urgently required, since it can lead to the development of secure practice in the use of these mechanisms. The open issues relate to [16]:

1. *Guessability*

- (a) *Number of images to be displayed at authentication* — how easy is it for a “blind guess” to be successful. For example, in order to get the same guessability odds as a 4 digit PIN we would have to display 40 images and ask the user to identify 4 pictures. This is more time-consuming and visually-challenging than clicking on a familiar PIN pad.
- (b) *Choice of authentication code images* — whether the user should be permitted to choose the images in the access code or whether the system should assign them randomly. From the world of passwords we know that people remember passwords better if they choose their own passwords. The flip side of this is that users tend to choose weak passwords and this is duplicated in the graphical world [3]. This aspect of guessability measures the possible success of a “semantic guessing attack” based on prior knowledge of the user’s characteristics, preferences and opinions.
- (c) *Choice of distractor images.* De Angeli *et al.* [4] found that users were easily confused if distractor images were chosen that were semantically similar to the access code images. They solved the problem for the VIP system by choosing semantically-dissimilar images as distractors to minimise confusion.

The higher the guessability, the less resistance is offered to attempts by a person targeting a particular user’s account.

2. *Observability* — the extent to which observation of the access code entry process can impact the strength of the mechanism. Web users often work in unsecured public areas and observability needs to be addressed for graphical mechanisms to be viable. Two issues need to be considered here:

- (a) *The positioning of target images amongst distractors* is an open question. Either the position should be varied at each attempt or kept constant. De Angeli *et al* [5] found that users were assisted in remembering the picture if it was always displayed in the same position. On the other hand, this reliable positioning of the image could assist the theft-minded

observer.

- (b) *The choice of distractor images* — either these are varied at each authentication attempt or chosen when the user enrolls with the system and used consistently for each authentication attempt. If they are varied this will assist observation attempts since only the targets remain constant and repeated observation will reveal valuable clues as to the identity of the access code images.

The lower the observability the fewer the opportunities offered to incidental identity thieves.

3. *Recordability* — how easy it is to record the access code images such that a non-observer can use them to gain access to the system. The lower the recordability the easier it becomes to resist users’ insecure behaviour which is often due to a limited understanding of the security issues.
4. *Analysability* — this measures the ease with which an intruder can analyse the system and thereby break into it more easily. For example, one of the most valuable things to an intruder is a helpful error message. One needs to walk a fine line between providing assistance to the legitimate user and providing clues to the potential intruder.

The final factor mentioned in [16] is *resistibility*. Whereas the previously mentioned four factors contribute to the weakness of the mechanism, the resistibility thereof serves to bolster it, and to defend against intrusion attempts. Some resistibility measures are:

1. *Continuous auditing* — We need to be able to classify failed attempts so that we can improve usability while maintaining security. Since these systems are relative newcomers to the security arena it is helpful to determine whether failures are due to genuine errors (memory lapses or difficulty using the interface) or whether they are intrusion attempts. This can only be done by means of continuous auditing. The use of logs is obviously not as good as direct observation of users because we have to infer behaviour and intentions from logged information but unfortunately when testing a web authentication mechanism, logs are often as close as we can get to direct observation of user behaviour.
2. *Timing interactions* — If the user is taking too long it could be that an impostor is trying to analyse the system to determine which image could be the correct one. On the other hand, if the inputs are coming in too fast it could be a brute force attack. These attacks are easily foiled by applying an n-strikes-and-you’re-out policy. This number is normally set to 3 attempts although Brostoff and Sasse [2] argue that the user be allowed at least 10 attempts before being locked out.
3. *User involvement* — The best person to monitor an online account is the legitimate user. If the system provides information about last-login-time or last-transaction when the user logs in, he or she is more likely to spot that the account has been broken into. Another good technique is to email or SMS a user

whenever the account is accessed. Such a notification may well come in time for the user to notify the bank of the intrusion.

The next section will discuss specific problems related to implementing alternative authentication mechanisms in a web environment.

A. Graphical Authentication over the Web

Authenticating a person over the web is completely different from authenticating an ATM user, for example. Most problems are related to the fact that the authentication application runs in an insecure environment. It runs on a machine that has unknown other software on it, and it runs within another application i.e. a browser. This complicates matters a great deal for the authentication system. Some browser behaviour, which supports browsing activity, can seriously interfere with the integrity of the graphical authentication process, which is often composed of more than one screen. Some examples are:

1. *Page Refresh* — if the system uses varying distractor images a simple refresh assists the potential intruder. Refresh can be done in one of three ways (right-click, using function key 5, main browser menu) and only one of these can be disabled by the authentication system (the right-click) and this only if JavaScript is enabled.
2. *Browsing direction* — the “back” action allows the user to redo previous authentication stages, which can confuse and possibly compromise the the system. This needs to be prevented by ensuring that the page is not cached and that it expires immediately upon exit.
3. *View Source* — determining the “names” of the images being used should not provide any valuable information. If one is not careful here the target image names might help an intruder to identify a particular common location amongst displayed images in successive pages during the authentication process.
4. *Print* — Key-stroke loggers are not a problem for graphical passwords but graphical access codes can easily be recorded by means of the ubiquitous `PrintScreen` key, which cannot be disabled.
5. *Insecure Environment* — the browser communicates from an insecure machine over an insecure network, so the observability options are endless. One can encrypt information sent over the web to the server, but a web administrator can hardly secure each client machine to prevent electronic observation and this is an open problem for web-based systems.
6. *Web agents* — it is entirely possible for an intruder to try to gain access to the system on a number of different occasions and to store all the handwritten numerals and then to use these to determine similarity by means of similarity detection software.
7. *Range of Attackers* — a software system that runs within a company intranet only has to be concerned about the activities of their employees, all of whom face possible unemployment if they are caught trying to break into a system. The web is a different arena.

There are many attackers, most of whom are faceless and nameless, and protecting against an unlimited number of malicious attackers is an uphill battle. The security specialist is often only one step ahead, if he or she is lucky.

The next section presents the DynaHand authentication mechanism.

IV. THE DYNHAND AUTHENTICATION SYSTEM

The DynaHand authentication mechanism uses randomised 5-digit numeral strings utilising the user’s own handwriting, displayed in black numerals on a white background. The 5-digit sequence is randomised for each stage in the login sequence and for each login session so that the user relies only on numeral recognition and not on memory of the PIN.

The user initially identifies himself by means of an email address, and is subsequently presented with a screen containing 9 images (shown in Figure 1). Users are only authenticated after successfully recognising a sequence of numbers displayed in their own handwritten numerals at *three* successive stages, and at each stage they are required to select *their* handwritten numeral string from the display.

In order to capture individual variations in handwriting each participant provided 12 samples of each numeral. The samples were gathered on A4 sheets, which were then digitised, and the numerals were automatically segmented out and stored in an appropriate file structure. At each login stage, handwritten numerals from *one* sample for *each* displayed user was used to display the 5-digit numbers, hence making the handwriting from each user look slightly different at each authentication attempt. This variation, it was hoped, would confuse would-be intruders, but would still enable the legitimate user to identify the target image containing the PIN in his own handwriting.

The main aim of this work has been to find a biometric that can be deployed over the web. In this case we have made use of a graphical authentication mechanism and rely on the user’s ability to recognise his or her own handwritten numerals. The specific problems mentioned in section III have been addressed as follows:

- *Guessability*
 - *Blind guessing*: The current DynaHand system has a guessability of 1 in 729 (1 out of 9 three times) but this is easily increased by using more distractors or increasing the number of stages. If we used 4 stages and 10 images at each stage the guessability would be equal to that of a 4 digit PIN.
 - *Semantic guessing*: the person being authenticated is usually able to recognise her own numerals fairly easily. An observer may see which image the person clicks on, but this information will probably not be useful later since the displayed number will be different.
 - *Caching*: Caching of previously used images is prevented by means of a meta-tag, as these images could be utilised in a handwriting analysis by an intruder to assist the previous type of guessing.

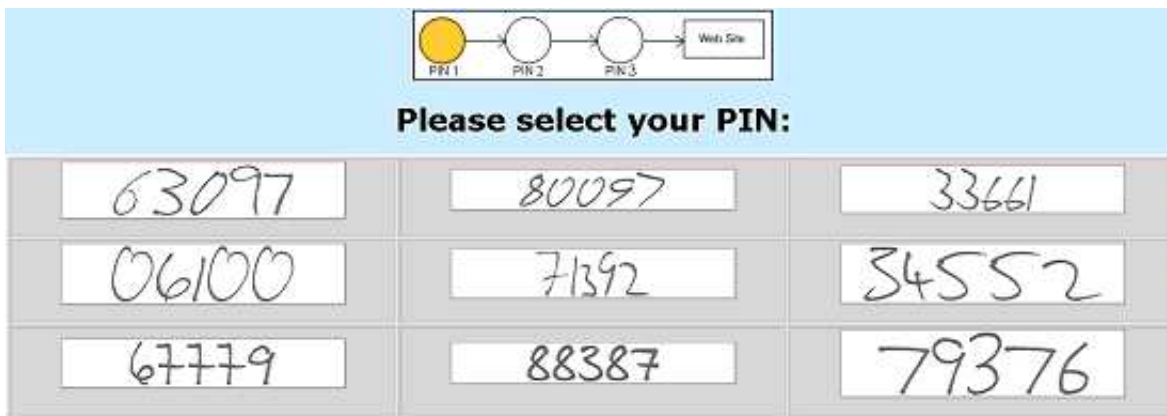


Fig. 1. One of the Stages of the DynaHand Authentication System

- *Observability*
 - *Positioning of target images amongst distractors:* DynaHand chooses the position randomly at each login step and at each login session since a predictable position could assist a casual observer.
 - *Choice of distractor images:* distractors are dynamically generated numbers using other handwriting samples, which makes them vary with each attempt. This could be a problem if distractor handwriting is too similar to the target image handwriting but this can be alleviated by using the similarity detection tool presented in the next section.
- *Recordability* — handwriting recognition is a tacit skill that is not easily articulated or recorded. People can write down their numerals if they wish to share their access key but it is likely that it will take the imposter some time to correctly match the displayed PINs to the written numerals and the system is likely to detect this if attempts are timed. The natural variations in handwriting also helps to mitigate this threat to a certain extent. Hence it is more difficult for someone to convey her access key to anyone else. Furthermore, it is less likely that the person will anticipate forgetting the key since the wherewithal to identify the key is safely stored in his or her own mind. Thus the need to record passwords is eliminated and the side-effect of this is that it strengthens the mechanism. However, it *is* possible for the user to print the authentication screens and to mark their code to give to someone else. It is not as effectual as giving them a password, but it will be helpful to an intruder.
- *Analysability* — the system does not provide any information in the source page that could be used by the intruder to obtain information about the underlying database or the system users. The system cannot disable caching, but we do ensure that the distractors chosen for the session are not varied, so that refreshing does not provide the intruder with any helpful information. We also ensure that a refresh counts as one strike against the user.
- *Resistibility* — the system currently permits 3 attempts before locking the user out. The purpose of

DynaHand was to determine the viability of a scheme based on handwritten digits and the website they gained access to did not hold any sensitive information. Hence the resistibility was not an important feature. In a production system, one would expect the authentication mechanism to include other resistibility features such as continuous logging and auditing. In a multi-page system such as DynaHand one can also practice a certain amount of “deception”. For example, if, on the first authentication page, the person being authenticated chooses incorrectly, the system can use other users’ digits for subsequent pages, and not include the genuine user’s handwritten digits. This will confuse the intruder and raise warning flags. User involvement and notification would also be essential in protecting the system.

The similarities between handwriting samples were computed, and this was used to determine the causes of errors. Errors in the system can be classified as:

1. *Genuine mistakes by a legitimate user* — This can be categorised as one of the following:
 - (a) The user selects a handwriting which is similar to his own;
 - (b) The user selects another handwriting because the numbers being displayed in her PIN are not distinctive enough and she is confused.
 - (c) The user made a genuine mistake — one whose cause cannot be accurately determined. This includes positional errors such as an extra click on the previous page being registered on the current page.
2. *Mistakes made by an impostor.* With most systems this is very difficult to identify and we therefore developed a similarity detection tool to assist this process.

Whereas DynaHand addresses many problems related to observability, guessability and recordability, it still needs to address all the above-mentioned auditing problems. The system cannot classify images based on semantic similarity to determine causes of errors and this makes it difficult to address the problems.

DynaHand particularly needs to choose distractor images that are sufficiently dissimilar to the target image, and

it needs to be able to detect problems any particular user is experiencing in using the system. All login attempts were logged, and the erroneous attempts were analysed using a similarity detection tool described in the next section.

V. SIMILARITY DETECTION TOOL

One way of determining similarity for systems using representational images is by comparing the incorrect images chosen with the target images based on either semantic similarity or visual appearance. So, for example, one of the target images could be a daffodil, but the user chooses a tulip instead. It is harder to make these comparisons for systems using images that cannot be semantically classified.

In order to investigate the potential strength of the DynaHand handwriting recognition method, computations have to be performed to ensure that numerals are easily distinguishable for the person who wrote them, but at the same time distinguishable with difficulty for anyone else. There is thus the need for a tool that can provide us with two kinds of information:

1. *How the distractors should be chosen* — the variations in the handwriting styles must be large enough for the user to distinguish between them. However, as Srihari [19] pointed out, the simple structure on some of the numerals makes it impossible to have a large variation on the writing style. In such cases it is important to have proper software to analyse the similarities in handwriting before they are used in authentication. In other words, the distractors being displayed should go through a validation process to ensure that the legitimate user will not be confused by them. Handwriting samples which are deemed to be within a certain similarity-span of the real handwriting should not be used.
2. *Analyse errors* — Failed authentication attempts should either signal an intrusion attempt or a legitimate user experiencing problems. We need to be able to “score” the target handwriting versus the chosen handwriting to determine their similarity and to determine whether the error was genuine or not.

A. Similarity Rating Tool

In response to the above two demands a tool was created to calculate similarity or dissimilarity between handwritten numerals. This tool runs offline on all the images in the database to ensure that the tool does not impact on response times. The analysis tool used statistical moments [7] and a novel approach with *Vector Quantisation* (VQ). It was encoded using Matlab, where digitised images of the participant’s handwritten numerals were used as input. The original images had single black numerals on white backgrounds, and these were colour-inverted to maximise the contrast of light and dark, as illustrated in Figure 2.

Statistical moments rely on extracting feature vectors and vector distances from the images. The feature descriptors describe the slopes, curves and stroke-characteristics

of the image [7] pp470-472. These numeric representations are in turn used to compute vector distances between images - hence providing a measure of similarity. Statistical moments capture information about the density of the pixels in the image, the orientation and gradient of the strokes, as well as the global height and width of the numeral strokes. In the analysis tool, statistical moments created an 8-element feature vector per image. The second technique applied was novel, as it was believed that vector quantisation had not previously been used for feature extraction. VQ is a technique used for image and voice compression. It works by determining the most descriptive patterns (called the codebook) in an image, and replacing all patterns with the most equivalent of the descriptive patterns. To perform vector quantisation, patches have to be extracted from the image, and an algorithm applied to the extracted patches to find those patches with the most descriptive patterns (i.e. the patches which can be used to encode the image and still obtain a very similar representation of the image.)

Patches of 5*5 and 7*7 pixels were utilised, where the smaller patch-size resulted in slightly more detailed encodings of the original images. Patches were extracted on a pixel-by-pixel basis, hence only moving the “patch widow” one pixel at the time. The 5*5 patches were transformed into 1*25 vectors and were then run through the LBG quantisation algorithm [9]. The LBG is an iterative algorithm which relies on the k-mean nearest neighbour algorithm for finding clusters in data. It starts with a codebook $C(0)$ obtained by taking the average of all the input image patches. The starting point is thus not random, as some versions of the k-mean algorithm are — and this was confirmed when running the same images several times obtained exactly the same distances and codebooks. If comparing each vector in the input with a point in space, the $C(0)$ is the centre point of all points. This initial codebook is then split into 2, which means that the set of points is split and two new centre-points are located. The aim is for the centre-point in a set to be located as far away from all other centres as possible, while simultaneously “owning” the same amount of points as the neighbouring centres. The two regions are split into 4 regions, which in turn are split into 8 regions, until there are as many regions as there are desired codebook elements — 32 in this case.

As the patches were extracted pixel-by-pixel, the images were also encoded pixel-by-pixel. This naturally lead to some pixel positions being presented with stimuli multiple times, and normalisation of the image was hence deemed necessary. To cope with the recursive addition of pixel values to each pixel position, an additive count was kept for each position, and each position was divided by this number in the end.

Upon reconstructing (or encoding) each image with elements from the codebook, a count was kept for each codebook element to log how many times it was used for the current image. Histograms were created and were normalised by the total number of elements contained in it, and it would then serve as a feature vector for the image.



Fig. 2. *Inverted Numeral*

Euclidean distances were used to compute distances between the feature vectors, and in effect the distances between images, much in the same way as was done by Srihari in [19]. Euclidean distance metric is non-intuitive, meaning that the results has to be considered in context. One distance can hence not be computed and used alone - all relevant distances will have to be computed in order to know the span of the results. A distance was computed from each image to every other image, hence providing 120 distances for each participant for each single number. Illustrations and further discussion of distances are given in the next section and in Figure 3.

B. Seeding the Tool

As each author supplied 12 samples of each numeral, distances between samples were computed both within each author as well as between the authors. In some cases the variability of a numeral for one author was larger than compared to some samples from another author. Displaying such similar samples on the screen simultaneously will increase the chances of a person selecting the wrong handwriting when logging in. The similarity computation tool can therefore be used to determine such similarities before displaying images on the screen, thus removing at least one of the causes of errors for genuine users.

The main purpose of the similarity computation tool was to determine the cause of errors. Similarity distances were computed for all authors and all samples in the system. Distances were then computed between the numeral samples actually appearing on the screen during the erroneous login. The largest distance between the offending numerals would be selected as least similar, hence gaining a 0 per cent similarity. The remaining distances between the offending numerals would get a percentage-wise distance in relation to this large distance. Computing distances based on the numerals actually displayed during the login attempt gives a better base for determining the cause for errors. Had a different set of samples been displayed the user might not have performed an erroneous selection, or a different kind of error might have resulted. As will be shown in the next section, the similarity computation tool did indeed match with, and verify, image-sample similarities detected by human manual inspection. The next section also evaluates the time taken to log into the system in relation to the login success-rate.

VI. EVALUATION

A website was created which offered users a new joke upon every successful login attempt and 10 users registered to use the site. There were 139 login attempts performed during the 31 day period during which the site was active. (An *attempt* consisted of a user's single interaction with the authentication system, where the outcome was either success or failure; a *session* is a sequence of attempts occurring straight after one another) After analysis of all login attempts, a single *session* was categorised as belonging to one of the the following (number of occurrences during trial in brackets):

- a single successful attempt not preceded by an erroneous attempt (100)
- a failed attempt immediately followed by a successful attempt (9)
- several consecutive failed attempts immediately followed by a successful attempt (3)
- several consecutive failed attempts *not* followed by a successful attempt (1)
- a failed attempt not followed by a new attempt (2)

The overall success rate of the DynaHand system was 87% for sessions with no errors, and 97.4% for successful sessions containing 0 or more errors. 3 sessions failed to succeed, indicating a 2.6% error rate in the system. However, 2 of the 3 failed sessions did not attempt a second login, which means there is no way of judging whether these sessions were complete failures or whether the user would have succeeded if a second attempt had been made. The third of the 3 failed sessions contained 4 failed attempts, which suggests that the user really did not recognise his handwriting in the system. The known failure-rate is then 1 out of 115 sessions, which constitutes 0.87% of all attempted sessions.

The error rate was 0.6% in the HandWing system [15], which is slightly lower than in DynaHand. Also, the success rate including sessions with 0 or more failed attempts obtained a 98.9% rating in the HandWing system, which is 1.5% better than with DynaHand. However, the success rate alone cannot be an indicator of the authentication mechanism, as the security and time to login also matters.

The average login time for DynaHand was 28.1 seconds, which is overall 9.36 seconds per stage. The average login time per stage was 15.9 sec, 8.1 sec and 6.2 sec, respectively. 27 erroneous attempts were performed in total, where 9 (6.5%) occurred at the first stage, 14 (10.1%) on

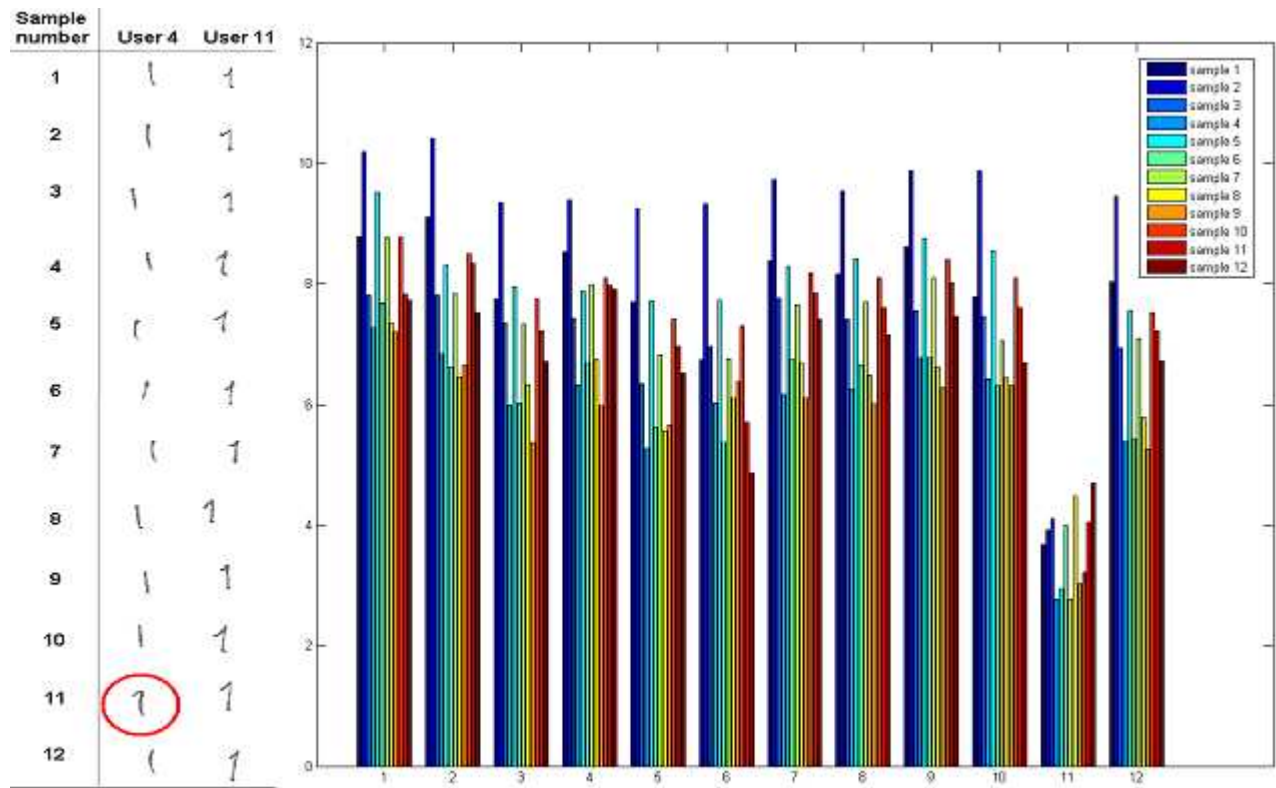


Fig. 3. Perceptive Similarities between Two Participants

the second and 4 (4.3%) at the third stage. The error margins and the login-times indicated that if a user managed to get to the 3rd stage they knew how to recognise their own handwriting.

Five of the 10 participants did not have any failed attempts at logging in. Of the people who did make mistakes, there was no persistent pattern in the handwriting they chose instead of their own. There was thus no evidence that any user consistently mistook one particular handwriting for his or her own. The similarity detection tool also failed to identify any particular tendency or similarity in the erroneous choices. This could be due to the fact that there were only 10 participants, so only a few different kinds of handwriting were used as distractors.

The fact that errors were inconsistent, however, supports the notion that handwriting is indeed unique, but it weakens the belief that people can *always* recognise their own handwriting. On the other hand, a 97.4% success rate indicates that participants did have the ability to recognise their own handwriting the majority of the time. While testing the DynaHand system it was discovered that one of the participants had great difficulty recognising his own handwriting. The authentication system should be flexible enough to offer such users an alternative mechanism.

Furthermore, the analysis of the data from the authentication site indicates that the time taken to log into the system decreases with practice.

To confirm the correctness of the distances resulting from the analytical similarity tool described in section V, manual comparisons were performed on some of the numer-

als. All samples of the same numeral were compared to every other, and order of similarity was determined, ranging from most similar to least similar. Figure 3 indicates distances measured between all samples from 2 participants. Human inspection showed that numeral 11 from participant 1 was more similar to all of participant 2's samples compared to the remaining samples from participant 1. The distances, as computed by the tool, are illustrated to the right in Figure 3, clearly showing the same findings as recorded in the manual inspection. This finding strengthened the belief in the correctness of distances computed by the analytical tool. It also illustrated how erroneous login-attempts made with the DynaHand authentication system could be identified as belonging to one of the classes described in Section IV.

VII. VIABILITY & FEASIBILITY

The initial purpose of DynaHand was to ascertain the viability of the use of the innate human handwriting recognition skills in an authentication setting. We tested it in the Web environment because it is challenging to find alternative authentication mechanisms for this environment due to the constrained nature of the available hardware and software and the uncertain skills of its users. In this sense DynaHand appeared to be a viable alternative because it required no special software or hardware and is very simple to use. In terms of the handwriting recognition side of things, our experimental subjects, with one notable exception, were able to recognise and identify their own handwritten numerals.

However, authentication is all about security, and one has to gauge the security of the mechanism in order to determine whether it indeed holds promise in a Web environment.

There are some problems with DynaHand in terms of security:

1. Caching is a huge problem because it assists the intruder in analysing the system and once the intruder has a full set of the user's handwritten digits it is much easier for him or her to compare these to the displayed digits and to gain access to the system. Whereas the web page setting gives the impression that caching can be disabled, it does not really do so, due to the use of accompanying technologies such as web proxies, for example. Furthermore, some browsers do not respect the NO-CACHE metatag.
2. The similarity detection software can be obtained and used by an attacker to determine similarity. If the attacker has a sample of the user's handwriting and she is trying to determine which of the on-screen images is likely to belong to the same user, this software may well solve the puzzle. Timing interactions *may* detect such efforts, but in a world of increasingly fast processors this is unlikely to be an effective strategy for long, if at all. If the attack is incidental and not targeted, the attacker will not really benefit from having the software since she will have no template to compare on-screen images to.
3. One cannot prevent the user from printing the screen, nor can one prevent attackers from capturing screen images without the user's knowledge.
4. Handwriting is recognisable not only to the scribe, but also to those close enough to the scribe to know his or her handwriting. While people tend to write less in the 21st century, co-habitants are likely to be able to recognise the user's handwriting based on shopping lists, addressed envelopes etc. DynaHand will therefore resist attacks by strangers more successfully than those from friends and family.

Thus DynaHand, whilst addressing observability and guessability problems, still falls foul of recordability and analysability. This is mainly due to the nature of the environment we chose to test DynaHand in — the web is a malicious place and any web authentication mechanism is subject to attacks from countless people and automated processes. Such is the nature of the World Wild Web.

It should be borne in mind, however, that authentication mechanisms should be chosen with the risk associated with the web site content in mind. If the content is fairly innocuous and of little value to an intruder, it could well be that a fairly weak mechanism such as DynaHand could be tenable.

VIII. CONCLUSION

The DynaHand system falls within the cognometric visuo-biometric class of alternative authentication systems. The system dynamically generates 5-digit handwritten random PINs, relying on the user's ability to recognise

his or her own handwriting. A tool was developed to determine similarity between handwriting samples. This analysis was used to assist the analyst in pinpointing the cause of errors when users selected the wrong handwriting in the DynaHand authentication system. With a 97.4% success-rate, the DynaHand system showed that it has excellent potential as an alternative to other recognition-based graphical authentication systems for protecting low-risk systems. What is more, it is one of the first web-based authentication mechanisms that provides cues to the legitimate user that are useless to many potential intruders. However, as discussed in Section VII, DynaHand *is* vulnerable to a determined attack or an attack from a close friend or family member and it should not be used to protect sensitive web content.

REFERENCES

- [1] S Brostoff and A Sasse. Are passfaces more usable than passwords? a field trial investigation. In S. McDonald, editor, *People and Computers XIV - Usability or Else! Proceedings of HCI 2000*, pages 405–424. Springer, 2000.
- [2] S Brostoff and A Sasse. Ten strikes and you're out: Increasing the number of login attempts can improve password usability. In *Workshop on Human-Computer Interaction and Security - Systems*, Fort Lauderdale, Florida, apr 2003. ACM. <http://www.andrewpatrick.ca/CHI2003/HCISEC/>.
- [3] D Davis, F Monrose, and M K Reiter. On user choice in graphical password schemes. In *Proceedings of the 13th USENIX Security Symposium*, San Diego, August 9-13 2004. <http://www.cs.jhu.edu/~fabian>. Accessed Sept 2006.
- [4] A De Angeli, M Coutts, L Coventry, and G I Johnson. VIP: A visual approach to user authentication. In *Proceedings of the Working Conference on Advanced Visual Interfaces AVI. 2002*, pages 316–323. ACM Press, 2002.
- [5] A De Angeli, L Coventry, G Johnson, and K Renaud. Is a picture really worth a thousand words? Reflecting on the usability of graphical authentication systems. *International Journal of Human-Computer Studies: special issue: HCI research on Privacy and Security*, 63(1-2):128–152, July 2005.
- [6] R Dhamija and A Perrig. Déjà vu: A user study using images for authentication. In *Proceedings of USENIX Security Symposium*, pages 45–58, Denver, Colorado, August 2000.
- [7] R Gonzalez, R Woods, and S Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall, 2004.
- [8] J G Heckman, C J Lang, and B Neundorfer. Recognition of familiar handwriting in stroke and dementia. *Neurology*, 57(11):2128–31, dec 2001.
- [9] Y Linde, A Buzo, and R M Gray. Vector quantisation. Internet, <http://www.data-compression.com/vq.shtml>. <http://www.data-compression.com/vq.shtml>. Accessed: Sept 2006.
- [10] M Longcamp, J L Anton, M Roth, and J L Velay. Visual presentation of single letters activates a premotor area involved in writing. *Neuroimage*, 19(4):1492–500, aug 2003.
- [11] OUT-LAW News. Password management still relies on post-it notes. Out-Law.Com, jun 2005. <http://www.out-law.com/page-5790>.
- [12] L O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2019–2040, dec 2003.
- [13] W Preyer. *Zur Psychologie des Schreibens (On the Physiology of Handwriting)*. 1895.
- [14] K Renaud and A De Angeli. My password is here! An investigation into visuo-spatial authentication mechanisms. *Interacting with Computers*, 16(6):1017–1041, 2004.
- [15] K V Renaud. A visuo-biometric authentication mechanism for older users. In *Proceedings British HCI 2005*, pages 167–182, Edinburgh, Sept 5-9 2005.
- [16] K V Renaud. A process for supporting risk-aware web authentication mechanism choice. *Reliability Engineering and System Safety*, 2007. In press.
- [17] R A Rensink. Change detection. *Annual Review of Psychology*, 53:245–277, 2002.

- [18] K Seki, M Yajima, and M Sugishita. The efficacy of kinesi-
thetic reading treatment for pure alexia. *Neuropsychologica*,
33(5):595–609, 1995.
- [19] S N Srihari, S-H Cha, H Arora, and S Lee. Individuality of
handwriting: A validation study. In *Proceedings of the Sixth
International Conference on Document Analysis and Recogni-
tion (ICDAR 01)*, page 106. IEEE Computer Society, 2001.
- [20] D Weinshall. Secure authentication schemes suitable for an as-
sociative memory. Technical Report TR 2004-30, Hebrew Uni-
versity, Leibniz Center for Research in Computer Science, 2004.
- [21] A Zimmer. Do we see what makes our script characteristic —
or do we only feel it? modes of sensory control in handwriting.
Psychological Research, 44(2):165–74, aug 1982.