Déjà Vu: A User Study
Using Images for Authentication*

Rachna Dhamija
rachna@sims.berkeley.edu

Adrian Perrig
perrig@cs.berkeley.edu

SIMS / CS, University of California Berkeley

Abstract

Current secure systems suffer because they neglect the importance of human factors in security. We address a fundamental weakness of knowledge-based authentication schemes, which is the human limitation to remember secure passwords. Our approach to improve the security of these systems relies on *recognition-based*, rather than *recall-based* authentication. We examine the requirements of a recognition-based authentication system and propose Déjà Vu, which authenticates a user through her ability to recognize previously seen images. Déjà Vu is more reliable and easier to use than traditional recall-based schemes, which require the user to precisely recall passwords or PINs. Furthermore, it has the advantage that it prevents users from choosing weak passwords and makes it difficult to write down or share passwords with others.

We develop a prototype of Déjà Vu and conduct a user study that compares it to traditional password and PIN authentication. Our user study shows that 90% of all participants succeeded in the authentication tests using Déjà Vu while only about 70% succeeded using passwords and PINS. Our findings indicate that Déjà Vu has potential applications, especially where text input is hard (e.g., PDAs or ATMs), or in situations where passwords are infrequently used (e.g., web site passwords).
Keywords: Human factors in security, hash visualization, user authentication through image recognition, recognition-based authentication.

*This publication was supported in part by Contract Number 102590-98-C-3513 from the United States Postal Service. The contents of this publication are solely the responsibility of the author and do not necessarily reflect the official views of the United States Postal Service.

1 Introduction

User authentication is a central component of currently deployed security infrastructures. We distinguish three main techniques for user authentication: *Knowledge-based systems*, *token-based systems*, and *systems based on biometrics*.

In today's security systems, knowledge-based schemes are predominantly used for user authentication. Although biometrics can be useful for user identification, one problem with these systems is the difficult tradeoff between impostor pass rate and false alarm rate [DP89]. In addition, many biometric systems require specialized devices, and some can be unpleasant to use.

Most token-based authentication systems also use knowledge-based authentication to prevent impersonation through theft or loss of the token. An example is ATM authentication, which requires a combination of a token (a bank card) and secret knowledge (a PIN).

For these reasons, knowledge-based techniques are currently the most frequently used method for user authentication. In this paper we focus on authentication based on passwords or PINs.

Despite their wide usage, passwords and PINs have a number of shortcomings. Simple or meaningful passwords are easier to remember, but are vulnerable to attack. Passwords that are complex and arbitrary are more secure, but are difficult to remember. Since users can only remember a limited number of passwords, they tend to write them down or will use similar or even identical passwords for different purposes.

One approach to improve user authentication systems is to replace the precise recall of a password or PIN with the recognition of a previously seen image, a skill at which humans are remarkably proficient. In general, it is much easier to recognize something than to recall the same information from memory without help [Nie93]. Classic cognitive science experiments show that humans have a vast, almost limitless memory for pictures in particular [Hab70, SCH70]. In fact, experiments show that we can remember and recognize hundreds to thousands of pictures in fractions of a second of perception [Int80, PC69]. By replacing precise recall of the password with image recognition, we can minimize the users cognitive load, help the user to make fewer mistakes and provide a more pleasant experience.

The basic concepts of recognition-based authentication are described by Perrig and Song [PS99]. In this paper, however, we explore the user authentication aspects more thoroughly, design the Déjà Vu system, and make the following contributions. First, we perform user studies of a prototype system to validate and improve our image-based user authentication system. Second, we analyze the security of Déjà Vu, discuss possible real-world attacks and illustrate countermeasures.

In the next section we enumerate the shortcomings of password-based authentication. In section 3, we discuss our approach of recognition-based authentication and introduce our solution, Déjà Vu. In section 4, we describe a user study that compares Déjà Vu to traditional authentication methods, and we summarize our findings. Finally, we discuss related work in section 5 and present our conclusions and future work in section 6.

2 Shortcomings of Password-Based Authentication

In this section, we enumerate the problems of password-based authentication, which we address with our work in section 3.

Password and PIN-based user authentication have numerous deficiencies. Unfortunately, many security systems are designed such that security relies entirely on a secret password. Cheswick and Bellovin point out that weak passwords are the most common cause for system break-ins [CB94].

The main weakness of knowledge-based authentication is that it relies on **precise recall** of the secret information. If the user makes a small error in entering the secret, the authentication fails. Unfortunately, precise recall is not a strong point of human cognition. People are much better at imprecise recall, particularly in **recognition** of previously experienced stimuli [Int80, PC69].

The human limitation of precise recall is in direct conflict with the requirements of strong passwords. Many researchers show that people pick easy to guess passwords. For example, an early study by Morris and Thompson on password security found that over 15% of users picked passwords shorter or equal to three characters [MT79]. Furthermore, they found that 85% of all passwords could be trivially broken through a simple exhaustive search to find short passwords and by using a dictionary to find longer ones. They describe an effort to counteract poor passwords, which consists of issuing random pronounceable passwords to users. Unfortunately, the random number generator only had 2^{15} distinct seeds, and hence the resulting space of "random" passwords could be searched quickly. Klein conducted a wide-reaching study of password security in 1989 and notes that 25% of all passwords can be broken with a small dictionary [Kle90].

Other notable efforts to design password crackers were conducted by Feldmeier and Karn [FK89] and Muffett [Muf92]. Because of these password cracker programs, users need to create unpredictable passwords, which are more difficult to memorize. As a result, users often write their passwords down and "hide" them close to their work space. Strict password policies, such as forcing users to change passwords periodically, only increase the number of users who write them down to aid memorability.

As companies try to increase the security of their IT infrastructure, the number of password protected areas is growing. Simultaneously, the number of Internet sites which require a username and password combination is also increasing. To cope with this, users employ similar or identical passwords for different purposes, which reduces the security of the password to that of the weakest link.

Another problem with passwords is that they are easy to write down and to share with others. Some users have no qualms about revealing their passwords to others; they view this as a feature and not as a risk, as we find in the user study discussed in section 4.

The majority of solutions to the problems of weak passwords fall into three main categories. The first types of solutions are proactive security measures that aim to identify weak passwords before they are broken by constantly running a password cracking programs [MT79, FK89]. The second type of solution is also technical in nature, which utilizes techniques to increase the computational overhead of cracking passwords [Man96]. The third class of solutions involves user training and education to raise security awareness and establishing security guidelines and rules for users to follow [AS99, Bel93].

Note that all three classes of solutions do not remedy the main cause of password insecurity, which is the human limitation of memory for secure passwords. In fact, most previously proposed schemes for knowledge-based user authentication rely on perfect memorization. One exception is the work of Ellsion et al. , which describes a knowledge based authentication mechanism that can tolerate user memory errors [EHMS99]. We discuss these schemes in detail in section 5.

3 Déjà Vu

In this section, we present a solution to address the shortcomings of passwords discussed in the previous section. In particular, we aim to satisfy the following requirements:

- The system should not rely on precise recall. Instead, it should be based on recognition, to make the authentication task more reliable and easier for the user.
- The system should prevent users from choosing weak passwords.
- The system should make it difficult to write passwords down and to share them with others.

3.1 System Architecture

We propose Déjà Vu as a system for user authentication. Déjà Vu is based on the observation that people have an excellent memory for images [Hab70, SCH70, Int80, PC69].

Using Déjà Vu, the user creates an image *portfolio*, by selecting a subset of p images out of a set of sample images. To authenticate the user, the system presents a *challenge set*, consisting of n images. This challenge contains m images out of the portfolio. We call the remaining $n - m$ images *decoy images*. To authenticate, the user must correctly identify the images which are part of her portfolio.

Déjà Vu has three phases: portfolio creation, training, and authentication.

Portfolio Creation Phase

To set up a Déjà Vu image portfolio, the user selects a specific number of images from a larger set of images presented by a server. Figure 2 shows the image selection phase in our prototype.

The type of images used has a strong influence on the security of the system. For example, if the system is based on photographs, it would be easy for users to pick predictable portfolios, to describe their portfolio images and to write down this information and share it with others. For this reason, we use Andrej Bauer's *Random Art* to generate random abstract images [Bau98]. Given an initial seed, *Random Art* generates a random mathematical formula which defines the color value for each pixel on the image plane. The image generation process is deterministic and the image depends only on the initial seed. It is therefore not necessary to store the images pixel-by-pixel, since the image can be computed quickly from the seed. All images are hand-selected to ensure consistent quality.[1]

Figure 1 illustrates sample *Random Art* images and appendix A discusses *Random Art* in more detail. Other methods exist for automatically synthesizing images [Sim91]. We did not explore these and leave this as an area for future study.

Training Phase

After the portfolio selection phase, we use a short *training phase* to improve the memorability of the portfolio images. During training, the user points out the pictures in her portfolio from a challenge set containing decoy images. The selection and the training phase need to occur in a secure environment, such that no other person can see the image portfolio.

[1]From our experience, about 70% of all generated *Random Art* images are aesthetically interesting and are therefore suited for usage in Déjà Vu. The remaining 30% are either too simplistic or fall into a class of images which are frequently generated and visually similar. Since we desire visually distinguishable images in the portfolio and the decoy set, we currently filter out weak images through hand selection.

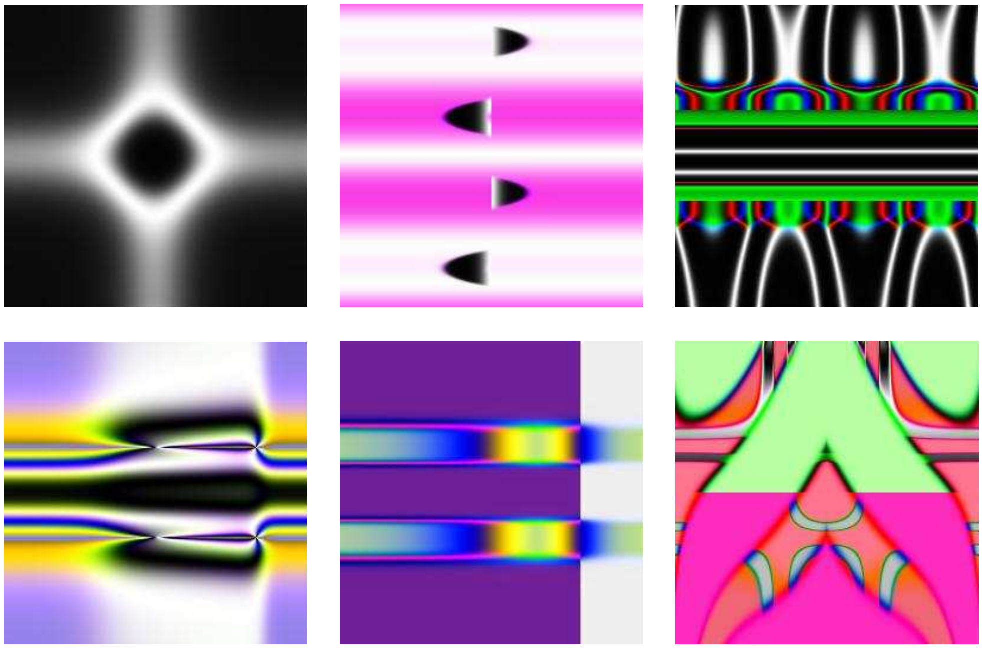


Figure 1: Examples of *Random Art* images

Authentication Phase

A trusted server stores all portfolio images for each user. Since each image is derived directly from the seed, the server only needs to store the seed and not the entire image. In our prototype implementation, the seed is 8 bytes long, hence the storage overhead for each portfolio is small. For each authentication challenge, the server creates a challenge set, which consists of portfolio and decoy images. If the user correctly identifies all portfolio images, she is authenticated.

In general, a weakness of this system is that the server needs to store the seeds of the portfolio images of each user in cleartext. Tricks similar to the hashed passwords in the `/etc/passwd` file do not work in this case, because the server needs to present the portfolio to the user, hidden within the decoy images. For this reason, we assume the server to be secure and trusted, similar to Kerberos [SNS88]. To reduce the trust required from each server, the portfolio can be to split among multiple servers, and each server can contribute a part of the challenge set for each authentication.

3.2 Attacks and Countermeasures

We identify a number of possible attacks which serve to impersonate the user. In the following scenarios, Mallory is an attacker who wants to impersonate Alice.

Brute-force attack. Mallory attempts to impersonate Alice by picking random images in the challenge set, hoping that they are part of Alice's portfolio. The probability that Mallory succeeds is $1/\binom{n}{m}$, which depends on the choice of n, the number of images in the challenge set, and m, the number of portfolio images shown. For example, for $n = 20$ and $m = 5$, we get $1/\binom{20}{5} = 1/15504$, which is equivalent to a four-digit PIN. To prevent brute-force attacks, the system may deny access after a small number of trials.

Educated Guess Attack. If Mallory knows Alice's taste in images he might be able predict which images are in Alice's portfolio.

Our first countermeasure is to use *Random Art*, which makes it hard for Mallory to predict Alice's portfolio images, even if he knows her preferences. Our user study shows that if photographs are used instead of Random Art, it is easier to predict some portfolio images chosen by Alice, given some knowledge about her.

Since users tend to pick the most aesthetically appealing pictures for their portfolios, it will be clear which images in the challenge set are the portfolio images if they are not all equally appealing. We therefore hand select images to ensure that no weak images are used. (We call images *weak*, if no user would select them for their portfolio). Hand selecting images is not a drawback, since a Déjà Vu system can function with a fixed set of images, on the order of 10,000 images.

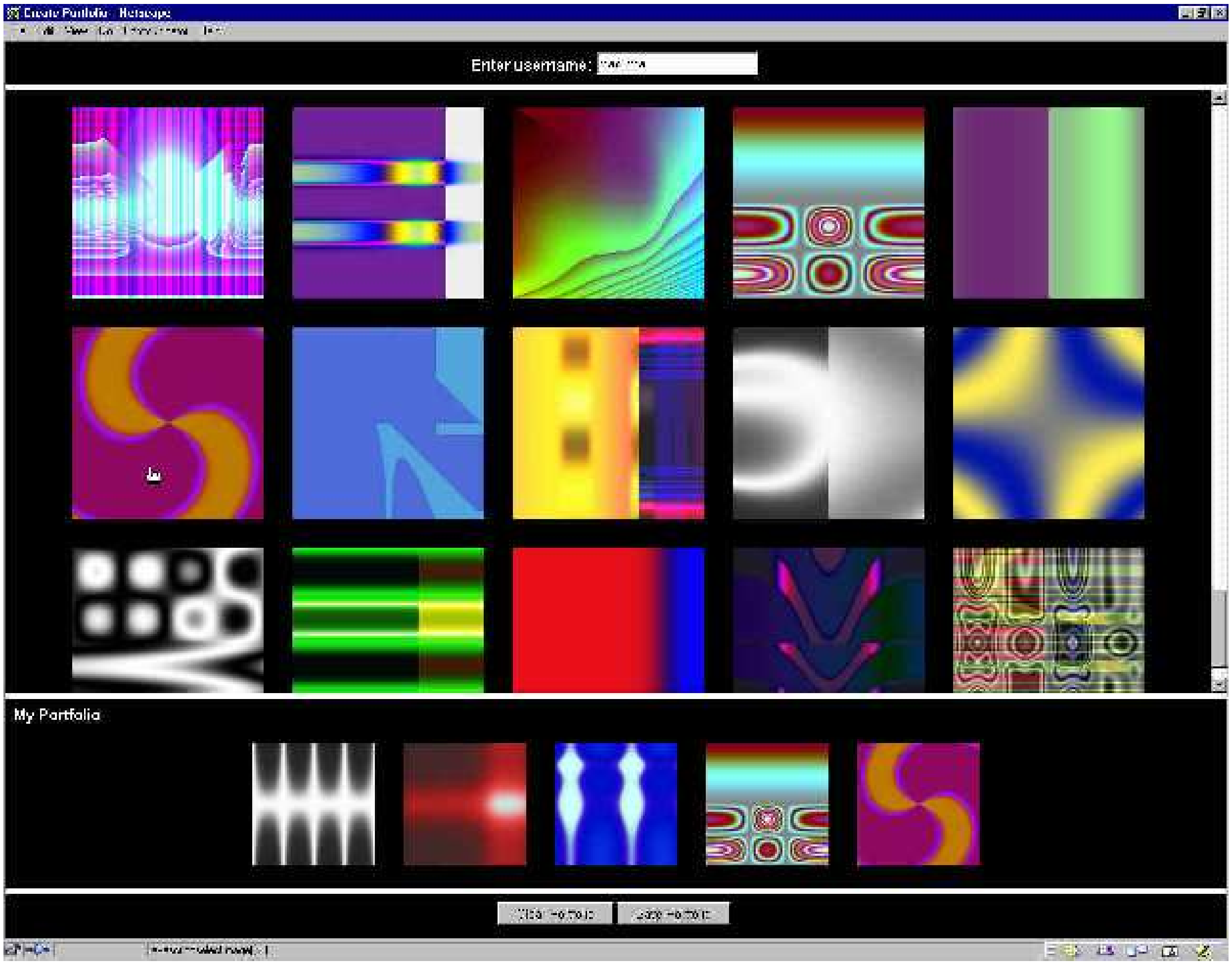


Figure 2: Portfolio selection window

Observer Attacks. Ross Anderson shows that observation of PIN codes on ATMs has been used to impersonate users [And94]. Similarly, if Mallory observes Alice during multiple authentications, he can know Alice's portfolio perfectly. We propose the following countermeasures.

- If the size of Alice's portfolio p is larger than the number of portfolio images in a challenge set m, the probability that an observer sees the same portfolio images after one observation is $1/\binom{p}{m}$. Although the security is still weakened after an observer learns images in a portfolio, an observer still can not impersonate Alice easily.

  Assuming that the images are displayed in a way that only Alice can see them clearly, the observer gains no knowledge of the portfolio by observing which images she selects, since the position of the portfolio images within the challenge set is randomized.

- The method for the image selection is hidden, such that an observer cannot see whether a given image is in the portfolio or not. If the observer cannot see which keys are pressed or can not determine which images are selected, he gets no useful information.

- The portfolio images can be slightly changed in each authentication. The goal is that a legitimate user can still recognize her portfolio images, while leaking less information about the portfolio to an observer. Further study is needed to explore image distortion methods and to determine how modifications in images are perceived by users.

Intersection Attack. If all the portfolio images are part of the challenge set, and all decoy images are changed in each challenge, Mallory can use the intersection of two challenge sets to reveal the portfolio. This is a serious problem, but we can design a system which can resist this attack through the following countermeasures.

- The same challenge set (portfolio images and decoy images) is always presented to the user. If it remains the same, an intersection attack does not reveal any useful information. The drawback, however, is that since the decoy images remain the same across many login sessions, Alice might start to remember decoy images and flag them as portfolio images in future authentication sessions. Future study is needed to see if this is the case.
- A small number of decoy images remain in the challenge set over multiple authentications. Again, the problem with this approach is that users may learn a decoy image if it is repeated enough times and then mistake it for a portfolio image.
- The authentication can be split up into multiple stages. Each stage presents a challenge set with a random number of portfolio images. If a user makes a mistake in any stage, all subsequent stages will only display decoy images without any portfolio images. This prevents an adversary from performing repeated impersonation attacks to discover the entire portfolio.
- We find in the user study that the failure rate is much lower for Déjà Vu than for password or PIN-based systems. This increased accuracy allows us to tighten the bound on unsuccessful logins before the account is blocked. This, however, opens the door to denial-of-service attacks which may render this method impractical.

Another possibility is to combine the countermeasures such that Mallory does not receive any useful information from multiple unsuccessful logins. First, the system uses the multi-stage authentication, which reveals only decoy images after the user makes an error in any stage. In addition, the system discards portfolio and decoy images that are shown in any unsuccessful login attempt. A shortcoming is that too few images may remain in the portfolio, and the system would need to perform a portfolio replenishment phase after a successful login. Since this takes time and may annoy the user, this method might be impractical. To prevent a denial-of-service attack from depleting the portfolio, the system can disable logins after a small number of unsuccessful login attempts. In case a user successfully authenticates after an unsuccessful attempt, the system can then replace the previously discarded portfolio images and perform a training phase with the images the user forgot.

3.3 Sample Applications

We describe two applications for which Déjà Vu is well suited and would improve security.

Customer Authentication at ATM

Banks face a multitude of problems concerning customer authentication at ATM's. First, many people have problems memorizing their PIN and pick either trivial PINs or write them on the ATM card. Anderson enumerates the many security problems with ATM's [And94].

The main problem for using Déjà Vu for ATM's is the portfolio creation. This is not a problem when customers pick up their card at the bank, since the portfolio selection and training can be done in a secure environment at the bank. If the client receives the ATM card in the mail, the portfolio creation is a more difficult problem. Sending all the images of the portfolio in the mail is not satisfactory, because we want to prevent people from possessing a paper copy of their secret information. Instead, we could use a one-time PIN to bootstrap the system, which the user can authenticate with initially at the ATM, which will then perform the portfolio creation and training.

The seeds of the portfolio images would be stored on a secure server. The authentication process would work as we describe previously. To achieve the same order of security as a four-digit PIN, we can use five images per portfolio and fifteen images in the decoy set. The probability of guessing the correct portfolio is $1/\binom{20}{5} = 1/15504$, which is lower than the $1/10000$ for four-digit PINs.

Web Authentication

The main problem with user authentication on the Web is that many sites are used infrequently and people forget their passwords over time. Another problem is that the number of sites which require a username and password combination to access it are increasing dramatically. The result is that users choose trivial passwords or they pick the same password that they already use for higher-security applications. Even so, users often forget their passwords; that's why many sites have forgotten-password recovery systems in place.

Déjà Vu is well suited for this problem because the "forgotten-password" recovery rate is very high for *Random Art* images, as we show in the user study. The creation of image portfolios is also easy to accomplish over the web.

4 User Study

We conducted a user study to compare a prototype image authentication system to traditional recall-based authentication systems (passwords and PINs). We compare two types of image portfolios, one using *Random Art* images and another which uses photographs. The user study consists of three phases: interviews, low-fidelity testing and formal prototype testing. In all phases participants were selected to be representative of the general population of computer users. An equal number of novice and expert users were selected, all of who were familiar with password authentication.

4.1 Task Analysis

In order to analyze the task of password authentication, we interviewed thirty people about their password behavior. While the sample size is small, our findings mirror the results of other larger surveys on the subject [AS99].

- We find that while participants have 10 - 50 instances where passwords are required, our users have only 1-7 unique passwords, which they use for multiple situations. Many of these unique passwords are variations on each other to aid memorability.
- Users have a variety of ways for coming up with passwords that they can remember. In most cases, people choose something that is personally meaningful to them (e.g., their own names, family members names, phone numbers, favorite movies). When asked to change passwords, most use a variation on a previous password. The average password length is 6 characters and the majority of passwords are composed of alphabetic characters appended by one or two numerical characters.
- The vast majority of participants write their passwords down (independently of whether they are novices or experts, or have been trained in password security). Some have a policy of writing all passwords down, while others just write down passwords initially until they remember them or only write down infrequently used passwords. Some users store their passwords in PDAs.
- System restrictions do impact password behavior. In general, users expend the minimum effort that is required to manage their passwords. For example, some will only make passwords alphanumeric or insert special characters if required, and most users did not ever change their passwords unless required to do so. However, restrictions do not prevent users from finding workarounds or engaging in other insecure behavior. One user likes having only one password to remember, so when she is required to change any password, she will change all of her other passwords to be the same.
- The level of security education or training also does not appear to have any impact on behavior. Although most users have received some sort of password security training, they ignored it stating that it was too cumbersome or simply not practical to follow.
- Some users who spoke foreign languages reported that they used their own names or words common in their native language as passwords, because "if it is not in English, it is hard for hackers to break". Apparently our users were not aware of the existence of multi-lingual password cracking dictionaries.
- An interesting finding is that people viewed the ability to share passwords with others as a feature. Almost all participants shared their bank PIN with family or friends and several users shared account passwords with others because this was a convenient way to collaborate, share information or transfer files.
- All participants expressed strong feelings of dislike and frustration with their experiences remembering, using and losing passwords. Yet surprisingly, most people preferred them to alternatives. For example many disliked hardware tokens because of experiences losing or misplacing them. A couple of participants who had experience with biometrics (fingerprint readers) felt that these systems were unreliable and performed poorly compared to passwords. Others disliked biometrics because of perceived privacy threats.

4.2 Informal Low-fi Prototype testing

Unlike high fidelity prototypes which are very detailed and may look very much like the final interface, low fidelity or "low-fi" prototypes are a rough rendition of the interface that presents only the main features. Low-fi prototypes are especially useful in early stage interface design to quickly iterate, test and experiment with new designs.

We tested the low-fi prototype to get early feedback on interfaces for portfolio selection and authentication (we did not compare it to text-based authentication at this stage). The low-fi testing also helped us to determine which variations in the *Random Art* algorithm produces the most memorable and distinguishable images and served as a way to preselect the images that would be used for future testing.

4.3 Formal User Testing

We developed a web-based prototype of Déjà Vu that allows users to create image portfolios and to authenticate themselves to the system later by selecting their portfolios from a challenge set. We designed a user study to compare Déjà Vu to standard web authentication using password/PIN dialogues.

We selected twenty participants (11 males and 9 females) to be representative of the general population of computer users. An equal number of novice and expert users were selected, all of who were familiar with password authentication.

The testing consisted of two sessions. During the first session, participants had to create a four digit PIN and a password with a minimum of six characters, both which they believed to be secure and that they had never used before. Other than character length, we imposed no limitations on the type of password or PIN created.

Participants also created two types of image portfolios, one consisting of five *Random Art* images and another consisting of five photographs. We presented each user with the same set of one hundred images to choose from, although the image order was randomized, to see if there was any similarity in the images chosen by users.

From user to user, we varied the order in which passwords, PINs, *Random Art* portfolios and photo portfolios were created to ensure that there was no bias due to task sequence.

Participants next had to authenticate using all four techniques, in the same order that they had created them. This ensured that several minutes and tasks elapsed between each PIN, password and portfolio creation and the login using that technique. To authenticate using image portfolios, users had to select their five portfolio images, which were randomly interspersed with twenty decoy images that were never seen before. (Selecting 5 images form a challenge set of 25 images results in 53,130 possible combinations, which is equivalent to a 4-5 digit PIN.) We gave participants an unlimited amount of time and attempts to login.

The second session occurred one week later and participants once again had to login using all four techniques (i.e., with the PIN, password and portfolios created in the first session). Again, we allowed an unlimited amount of time and number of attempts.

4.4 Task Completion Time and Error Rate

It took longer for users to create image portfolios than to create passwords and PINS. Photo portfolios took longer to create than *Random Art* portfolios, because people spent more time browsing and looking at each image.

Users also required more time to login with image portfolios compared to passwords and PINs. It took slightly longer for users to login using *Random Art* compared to photos, suggesting that people can recognize photographic images more quickly than abstract images.

After one week, however, there was a greater degradation in performance with PINs and passwords compared to portfolios. Table 1 shows the average creation and login times. The reason for the longer than expected login times for passwords and PINs is that several users required multiple attempts. (Note that login times include multiple attempts, but do not include those who could not login at all).

A number of minor and major errors were made with PINs, passwords and portfolios. During the first session all users were able to recover from their errors and to login successfully with portfolios, but this was not always the case with PINs and passwords, no matter how long or how many login attempts were made.

Even after one week, the number of unrecoverable errors made with images was far lower than that of passwords

	PIN	Password	Art	Photo
Create	15	25	45	60
Login	15	18	32	27
Login (after one week)	27	24	36	31

Table 1: Average seconds to create/login

and PINs. If we imposed more secure password and PIN restrictions (e.g., restrictions on character length and type, limited number of attempts), we suspect that the number of failed logins with passwords and PINs would increase. In contrast, all users were able to remember at least four out of five of their portfolio images on the first attempt.

Further study is needed to discover how frequency of use and long term memory effects will influence performance and error rates in portfolio authentication.

4.5 Qualitative Results

Déjà Vu is easier than it looks: Although some users remarked that they would never be able to remember the portfolios they created, all were surprised that they could recognize their images and at how quickly the selection took place. It is interesting to note that after the first week, more users forgot their usernames than their portfolios.

Text vs. images: The majority of users reported that photo portfolios were easier to remember than PINs and passwords, especially after 1 week, and that they would use such a system if they were confident that it was secure and if image selection times were improved.

Random Art vs. photos: Users varied in whether they thought photo or *Random Art* portfolios were easier to use.

Users tend to select photographic images based on a theme or something that has personal meaning to them. (e.g., hobbies, places they have visited). There was much more variation in the *Random Art* images selected by users compared to the photographs. For example, although participants were presented with a choice of 100 images, 9 out of the 20 participants included a photograph of the Golden Gate bridge in their portfolios. In contrast, there were few *Random Art* images that were chosen by more than one user.

After the user testing was complete, users described the portfolios they had chosen. The descriptions of a photograph chosen by more than one user were virtually identical from user to user. However, no two descriptions of a *Random Art* image were alike. Participants found it hard to describe or to recall the *Random Art* images in concrete terms and instead related them to objects or actions (e.g., "it looks like a woman dancing"). For this reason, we conjecture that it would be hard for a third party to identify another's portfolio images based on descriptions or recalled drawings alone. Further study is needed to see if this is the case.

4.6 Interface Issues

Faster image portfolio creation and login will help to make such a system usable. One improvement would be to reduce image size to minimize the need for scrolling, which occupied a significant portion of the task completion time.

Users wished to have more feedback in many instances, but it will be important to give users feedback without compromising security. For example, during portfolio creation and authentication, some users were confused about how many images they had picked thus far if a portfolio window was not available. If portfolios are created in a secure environment, it would be possible to show thumbnails of the selected images. In the case of an insecure environment, simply providing the number of images picked thus far would be an improvement.

5 Related Work

We review previous work which makes an attempt to solve the problem of password-based user authentication.

Blonder patented a "graphical password", which requires a user to touch predetermined areas of an image (tap regions) in a predetermined sequence for authentication [Blo96]. The main drawback to this system is that

	PIN	Password	Art	Photo
Failed Logins	5% (1)	5% (1)	0	0
Failed Logins (after one week)	35% (7)	30% (6)	10% (2)	5% (1)

Table 2: % Failed logins (# failed logins/20 participants)

it is location and sequence dependent, so the user is required to recall the regions to tap and the correct order in which to tap them.

Jermyn, et al. propose a graphical password selection and input scheme, where the password consists of a simple picture drawn on a grid. [JMM+99]. A benefit of their solution is that it removes the need for temporal recall, by decoupling the position of inputs from the temporal order in which those inputs occur. Early cognition experiments do indeed support the claim that pictures are recalled better than words. Their solution, however, still suffers from the fact that it requires users to precisely recall how to draw their images, rather than relying on recognition.

Passlogix Inc. distributes v-go, an application which remembers user names and passwords and automatically logs the user on to password-protected Web sites and applications [Pas00]. They allow users to create passwords by clicking on objects in a graphical window, such as by entering the time on a clock, drawing cards from a card deck, selecting ingredients to mix a cocktail or to cook a meal, dialing a phone number, hiding objects in a room, trading stocks, and entering a password on a keyboard. The weaknesses of their system are manyfold.

First, the space of different passwords is very small. For example, there are only limited places available to select to cook a meal. In the case of hiding objects in a room, the requirement to hide the objects already strongly reduces the state space. It would be better if the user could place objects in arbitrary locations. There are only a few places in the given room where the objects can really be hidden, for example under the mattress or the cabinet are locations which users are likely to select.

Furthermore, the system allows users to pick poor passwords. For example, choosing all aces in a deck of cards is certainly not secure. It is likely that many users will choose commonly known combinations, for example by choosing to mix the same drinks.

Finally, the system requires users to precisely recall the authentication task, instead of relying on recognition. Another weakness is that an attacker will only need to break the v-go password to get access to all the users' other passwords.

IDArts distributes Passfaces, an authentication system based on recognizing previously seen images of faces [Art99]. This idea is similar to ours, and there is strong evidence to support their claim that humans have an innate ability to remember faces. They claim that authentication rates can be significantly improved by "training" the user during passface creation, which we did not do in our study.

A drawback of their system is that users pick faces which they are attracted to, which greatly facilitates impersonation attacks. Interestingly, in our study many users told us that they did not select photographs of people because they did not feel that they could relate personally to the image. We did notice that when pictures of people were chosen, the people closely resembled the users (e.g., one user selected an image that resembled his grandparents, one Indian woman selected an image of an Indian woman and a Chinese woman selected an image of a Chinese man). Since we use randomly generated images, knowing the preferences of a person only has limited usefulness.

Ellison et al. propose a scheme in which a user can protect a secret key using "the personal entropy in his whole life", that is by encrypting the passphrase using the answers to several personal questions [EHMS99]. The scheme is designed so that a user can forget the answers to a subset of the questions and still recover the secret key, while an attacker must learn the answer to a large subset of the questions to learn the key.

Naor and Shamir propose a Visual Cryptography scheme, which splits secret information into two transparencies, such that each part contains no useful information, but the combination reveals the secret [NS95]. Naor and Pinkas extend this idea as a means for a user to authenticate text and images [NP97]. In this case, the recipient is equipped with a transparency. When the recipient places the transparency over a message or image that was sent to him, the combination of both images reveals the message. Visual cryptography could be used to devise a user authentication scheme that is token based.

Ian Goldberg's "visual key fingerprint"[Gol96] and Raph Levien's [Lev96] PGP Snowflake were developed as a way to graphically identify and recognize PGP key fingerprints.

Adams and Sasse propose that educating users in security is a solution for the problem of choosing weak passwords [AS99]. They claim that if users receive specific security training and understand security models, they will select secure passwords and refrain from engaging in insecure behavior. In our user study, however, we discover that the level of security training did not prevent users from choosing trivial passwords or from storing them insecurely. We conjecture that this is the case because people prefer convenience over security. Therefore, security should be an inherent component of the system by default.

6 Conclusions and Future Work

Previous research recognized the weaknesses of knowledge-based authentication schemes (in particular password-based computer logins). So far, however, most of the proposed solutions have been based on technical fixes or on educating users. Neither of these address the fundamental problem of knowledge-based authentication systems, which is that the authentication task is based on precise recall of the secret knowledge.

Since people are much better at recognizing previously seen images than at precisely recalling pass phrases from memory, we employ a recognition-based approach for authentication. We examine the requirements of a recognition-based system and propose Déjà Vu, in which we replace the precise recall of pass phrases with the recognition of previously seen images. This system has the advantage that the authentication task is more reliable, easier and fun to use. In addition, the system prevents users from choosing weak passwords and makes it difficult for users to write passwords down and to communicate them to others.

We conducted a user study which compares Déjà Vu to traditional password and PIN authentication. Results indicate that image authentication systems have potential applications, especially where text input is hard (e.g., PDAs or ATMs), for infrequently used passwords or in situations where passwords must be frequently changed. Since the error recovery rate was significantly higher for images, compared to passwords and PINS, such a system may be useful in environments where high availability of a password is paramount and where the difficulty to communicate passwords to others is desired. Further study is required to determine how user performance and error rate will vary with frequency of use, over longer time periods and with large or multiple portfolios.

Many improvements can be made to strengthen the system against attack and to improve its usability. For example, we are exploring ways to mask or distort portfolio images, such that users will be able to recognize their images, while leaking information about the portfolio to observers. We are also exploring authentication schemes that take advantage of other innate human abilities (e.g., spatial navigation).

Hackers recognize that humans are often the weakest link in system security and exploit this using social engineering tactics[Kni94]. Yet designers do not always include human limitations in their evaluation of system security. Systems should not only be evaluated theoretically, but by how secure they are in common practice.

Acknowledgments

We would like to thank Doug Tygar, James Landay, and John Canny for their encouragement and advice. We would also like to thank Dawn Song and Ben Gross for their valuable feedback. Furthermore, we would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[And94] Ross J. Anderson. Why Cryptosystems Fail. *Communications of the ACM*, 37(11):32–40, November 1994.

[Art99] ID Arts. `http://www.id-arts.com/technology/papers/`, 1999.

[AS99] Anne Adams and Martina Angela Sasse. Users are not the enemy: Why users compromise computer security mechanisms and how to take remedial measures. *Communications of the ACM*, 42(12):40–46, December 1999.

[Bau98] Andrej Bauer. Gallery of random art. WWW at `http://andrej.com/art/`, 1998.

[Bel93] W. Belgers. Unix password security, 1993.

[Blo96] G. Blonder. United states patent, 1996. United States Patent 5559961.

[CB94] B. Cheswick and S. Bellovin. Firewalls and internet security: Repelling the wily hacker, 1994.

[Dha00] Rachna Dhamija. Hash visualization in user authentication. In *Proceedings of the Computer Human Interaction 2000 Conference*, April 2000.

[DP89] D. W. Davies and W. L. Price. *Security for Computer Networks*. John Wiley and Sons, 1989.

[EHMS99] Carl Ellison, Chris Hall, Randy Milbert, and Bruce Schneier. Protecting secret keys with personal entropy. to appear in Future Generation Computer Systems, 1999.

[FK89] D. C. Feldmeier and P. R. Karn. UNIX password security—ten years later (invited), 1989. Lecture Notes in Computer Science Volume 435.

[Gol96] Ian Goldberg. Visual key fingerprint code. Available at `http://www.cs.berkeley.edu/iang/visprint.c`, 1996.

[Hab70] Ralph Norman Haber. How we remember what we see. *Scientific American*, 222(5):104–112, May 1970.

[Int80] Helene Intraub. Presentation rate and the representation of briefly glimpsed pictures in memory. *Journal of Experimental Psychology: Human Learning and Memory*, 6(1):1–12, 1980.

[JMM+99] Ian Jermyn, Alain Mayer, Fabian Monrose, Michael K. Reiter, and Aviel D. Rubin. The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Symposium*, August 1999.

[Kle90] Daniel Klein. A survey of, and improvements to, password security. In *Proceedings of the USENIX Second Security Workshop, Portland, Oregon*, 1990.

[Kni94] The Knightmare. *Secrets of a Super Hacker*. Loompanics Unlimited, Port Townsend, Washington, 1994.

[Lev96] Raph Levien. Pgp snowflake. Personal communication, 1996.

[Man96] Udi Manber. A simple scheme to make passwords based on one-way functions much harder to crack. *Computers and Security*, 15(2):171–176, 1996.

[MT79] R. Morris and K. Thompson. Password security: A case history. *Communications of the ACM*, 22(11), Nov 1979.

[Muf92] D. Muffett. Crack: A sensible password checker for unix, 1992. A document distributed with the Crack 4.1 software package.

[Nie93] Jakob Nielsen. *Usability Engineering*. Academic Press, 1993.

[NP97] M. Naor and B. Pinkas. Visual authentication and identification. In Burt Kaliski, editor, *Advances in Cryptology - Crypto '97*, pages 322–336, Berlin, 1997. Springer-Verlag. Lecture Notes in Computer Science Volume 1294.

[NS95] M. Naor and A. Shamir. Visual cryptography. In Alfredo De Santis, editor, *Advances in Cryptology - EuroCrypt '94*, pages 1–12, Berlin, 1995. Springer-Verlag. Lecture Notes in Computer Science Volume 950.

[Pas00] Passlogix. v-go. WWW at `http://www.passlogix.com/`, 2000.

[PC69] A. Paivio and K. Csapo. Concrete image and verbal memory codes. *Journal of Experimental Psychology*, 80(2):279–285, 1969.

[PS99] Adrian Perrig and Dawn Song. Hash visualization: A new technique to improve real-world security. In *Proceedings of the 1999 International Workshop on Cryptographic Techniques and E-Commerce (CrypTEC '99)*, 1999.

[SCH70] L. Standing, J. Conezio, and R.N. Haber. Perception and memory for pictures: Single-trial learning of 2500 visual stimuli. *Psychonomic Science*, 19(2):73–74, 1970.

[Sim91] Karl Sims. Artificial evolution for computer graphics. In Thomas W. Sederberg, editor, *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics (SIGGRAPH '91)*, pages 319–328, Las Vegas, Nevada, USA, July 1991. ACM Press.

[SNS88] J. Steiner, C. Neuman, and J. Schiller. Kerberos: An authentication service for open network systems. In *USENIX Conference Proceedings*, pages 191–200, 1988.

[WT99] Alma Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th USENIX Security Symposium*, August 1999.

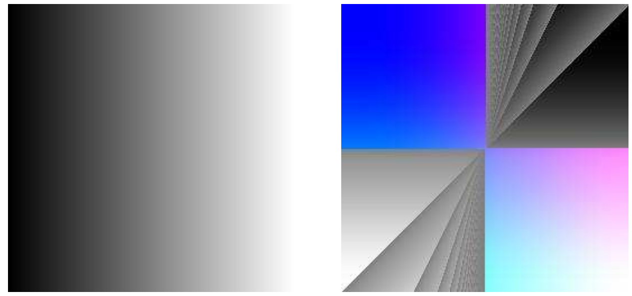
A *Random Art*

One proposed hash visualization algorithm is Random Art, a technique that converts meaningless strings into abstract structured images. *Random Art* was developed by Andrej Bauer, and is based on an idea of genetic art by Michael Witbrock and John Mount. Originally *Random Art* was conceived for automatic generation of artistic images. A brief overview and demonstration of *Random Art* can be found at Andrej's *Random Art* web site [Bau98].

The basic idea is to use a binary string s as a seed for a random number generator. The randomness is used to construct a random expression which describes a function generating the image—mapping each image pixel to a color value. The pixel coordinates range continuously from -1 to 1, in both x and y dimensions. The image resolution defines the sampling rate of the continuous image. For example, to generate a 100×100 image, we sample the function at 10000 locations.

Random Art is an algorithm such that given a bit-string as input, it will generate a function $\mathcal{F} : [-1,1]^2 \to [-1,1]^3$, which defines an image. The bit-string input is used as a seed for the pseudo-random number generator, and the function is constructed by choosing rules from a grammar depending on the value of the pseudo-random number generator. The function $\mathcal{F}$ maps each pixel (x, y) to a RGB value (r,g,b) which is a triple of intensities for the red, green and blue values, respectively. For example, the expression $\mathcal{F}(x, y) = (x, x, x)$ produces a horizontal gray grade, as shown in figure 3(a). A more complicated example is the following expression, which is shown in figure 3(b).

$$\begin{aligned}&\mathsf{if}\ xy > 0\ \mathsf{then}\ (x, y, 1)\\ &\qquad\mathsf{else}\ (\mathsf{fmod}(x, y), \mathsf{fmod}(x, y), \mathsf{fmod}(x, y))\end{aligned} \tag{A.1}$$



(a) Image for expression (x, x, x)

(b) Image for expression (A.1)

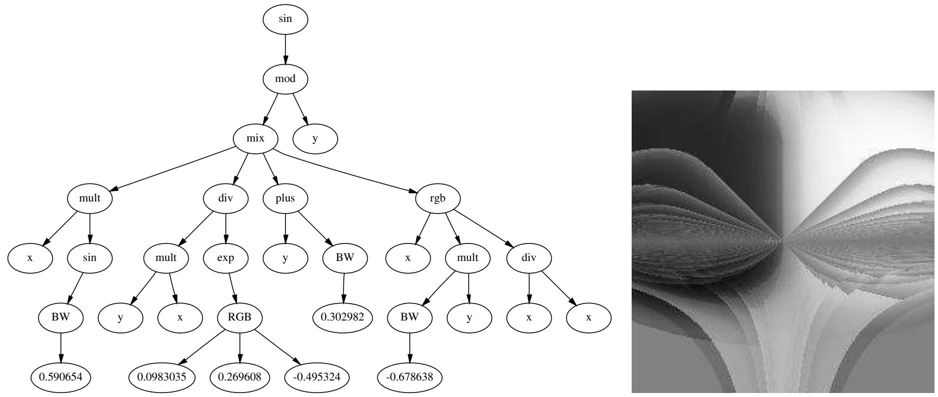
Figure 3: Examples of images and corresponding expressions.

The function $\mathcal{F}$ can also be seen as an expression tree, which is generated using a *grammar* G and a *depth parameter* d, which specifies the minimum depth of the expression tree that is generated. The grammar G defines the structure of the expression trees. It is a version of a context-free grammar, in which alternatives are labeled with probabilities. In addition, it is assumed that if the first alternative in the rule is followed repeatedly, a terminal clause is reached. This condition is needed when the algorithm needs to terminate the generation of a branch. For illustration, consider the following simple grammar:

$$\begin{aligned}E &::= (C, C, C)^{(1)}\\ A &::= \langle\text{random number} \in [-1,1]\rangle^{(\frac{1}{3})} \mid x^{(\frac{1}{3})} \mid y^{(\frac{1}{3})}\\ C &::= A^{(\frac{1}{4})} \mid \mathrm{add}(C, C)^{(\frac{3}{8})} \mid \mathrm{mult}(C, C)^{(\frac{3}{8})}\end{aligned}$$

The numbers in subscripts are the probabilities with which alternatives are chosen by the algorithm. There are three rules in this simple grammar. The rule E specifies that an expression is a triple of compound expression C. The rule C says that every compound expression C is an atomic expression A with probability $\frac{1}{4}$, or either the function add or mult applied to two compound expressions, with probabilities $\frac{3}{8}$ for each function. An atomic expression A is either a constant, which is generated as a pseudorandom floating point number, or one of the coordinates x or y. All functions appearing in the *Random Art* algorithm are scaled so that they map the interval $[-1, 1]$ to the interval $[-1, 1]$. This condition ensures that all randomly generated expression trees are valid. For example, the scaling for the add function is achieved by defining $\mathrm{add}(x, y) = (x + y)/2$.

The grammar used in the *Random Art* implementation



(a) *Random Art* expression tree (b) Generated image

Figure 4: *Random Art* expression tree and the corresponding image

is too large to be shown in this paper. Other functions included are: sin, cos, exp, square root, division, mix. The function $\mathrm{mix}(a, b, c, d)$ is a function which blends expressions c and d depending on the parameters a and b. We show an example of an expression tree of depth 5 in figure 4, along with the corresponding image. For the other images in this paper, we used a depth of 12.