Make Mine a Quadruple: Strengthening the Security of Graphical One-Time PIN authentication

Ravi Jhawar, Philip Inglesant, Nicolas Courtois, M. Angela Sasse
Dept. of Computer Science
University College London
Gower Street, London WC1E 6BT, UK
ravi.jhawar.09@ucl.ac.uk, p.inglesant@ed.ac.uk, {n.courtois, a.sasse}@cs.ucl.ac.uk

***Abstract*—Secure and reliable authentication is an essential prerequisite for many online systems, yet achieving this in a way which is acceptable to customers remains a challenge. GrIDsure, a one-time PIN scheme using random grids and personal patterns, has been proposed as a way to overcome some of these challenges. We present an analytical study which demonstrates that GrIDsure in its current form is vulnerable to interception. To strengthen the scheme, we propose a way to fortify GrIDsure against Man-in-the-Middle attacks through (i) an additional secret transmitted out-of-band and (ii) multiple patterns. Since the need to recall multiple patterns increases user workload, we evaluated user performance with multiple captures with 26 participants making 15 authentication attempts each over a 3-week period. In contrast with other research into the use of multiple graphical passwords, we find no significant difference in the usability of GrIDsure with single and with multiple patterns.**

***Index Terms*—Graphical Passwords, one-time PINs, Usable Security, Man-in-the-Middle, Entropy, GrIDsure**

I. Introduction

Secure and usable authentication remains a challenge for most information systems. Despite other forms of authentication, such as biometrics, being developed, knowledge-based authentication through passwords and PINs are very widely used. However, Users have well-documented problems recalling text-based passwords or Personal Identification Numbers (PINs) [1], [14]; as a result and tend to choose predictable values [16], [29] or resort to other potentially unsafe practices.

PINs, in particular, have a low set of possible values, and users tend to select from an even smaller set of choices - to increase memorability, they either choose significant dates or use simple sequences of numbers such as 1248 [24]; for example, [21] reported a study which finds that 18% of users chose their birthdays as PINs. Given that PINs can be captured through attacks such as key-logging and shoulder-surfing, they are at risk of being compromised [12], [16]. Security could be improved through the use of one-time PINs; however, ever since their first use as far back as World War II they have been known to be difficult to use [18] and are not practical in many situations.

Graphical password schemes [8], [10], [15], [19], [28] have been proposed as more memorable alternatives to textual passwords, using the recognised ability of human memory to recall (or in some schemes, to recognise) pictures or shapes rather than text.

In this paper, we analyze the security of GrIDsure [13], a patented authentication scheme which combines a graphical, shape-based password with a one-time PIN, without requiring special hardware. GrIDsure has been applied to Microsoft IAG and UAG, Windows or Active Directory login - optionally as part of 2-factor authentication - document authorisation and signing, and has been implemented as an authentication option on a smartcard.

The statistical security of GrIDsure has been investigated by Weber [27], who concluded that GrIDsure is at least as secure as a static PIN, and, for threats such as shoulder-surfing provides far greater security. In contrast, Bond [6] argued that this scheme is no more secure than a standard PIN because users are likely to choose from a limited subset of predictable patterns.

Brostoff et al. [7] found that users did indeed chose predictable patterns unless they were instructed to pick less predictable ones, and concluded that Bond's analysis affected security in some usage scenarios, but not others. They recommended its use as a second factor authentication where the capture of both one-time PIN and grid is unlikely such as at Point-of-Sale. The authors argued that user performance with GrIDsure warranted further examination, and whether the security issues could be addressed through modifications.

In this paper, we exactly do that: with a detailed analysis of the security of GrIDsure from a new empirical work, we suggest an enhanced system to overcome the security issues identified in previous studies. The aim of our effort was to see if it is possible to increase the security of the system without reducing its usability. We therefore conducted a usability study on a prototype version of the new system and obtained encouraging results.

The remainder of this paper proceeds as follows. After describing the GrIDsure scheme and summarising previous studies on its security and usability, we demonstrate that GrIDsure as it stands is not resistant to intercepted communications. Exploiting the commonly used computer security concepts, we identify several enhancements which provide effective resistance for GrIDsure against Man-in-the-Middle attacks. We report the results of the usability of these enhancements based on our evaluation, and find that there is no significant difference in recall reliability between the original GrIDsure and our enhanced design.

II. Introduction to GrIDsure

Graphical passwords are more easy to use than passwords and PINs because they offer cued (rather than unaided) recall and this makes them particularly well suited for infrequent authentication [3], [4], [23]. GrIDsure uses graphical scheme to generate a one-time PIN which users read off and enter into another application or device. It is essentially an example of a graphical password scheme and effectively a combination of both, a graphical and PIN authentication scheme.

With GrIDsure, the user has to remember a pattern rather than a passcode or a complex password. It works in two basic steps:

1) Registering Personal Identification Pattern (PIP)
   The user has to choose a pattern - a shape and sequence of squares on the grid, and register the pattern with username or account. The pattern can be of any length - e.g. 4 to replace a 4-digit PIN - and any shape that the user finds easy to remember. Note that the *order* of the chosen squares is significant.
   For enrollment, a grid with non-repeating characters spread in a random fashion can be used, where the user enters the characters which correspond to his or her chosen pattern.
2) Using the Personal Identification Pattern
   A grid with random numbers in each cell is displayed to the participant when he uses the system. The user then has to enter the numbers that correspond to his registered pattern as his one-time PIN.
   An example of a pattern on a random grid is shown in Fig. 1 (of course, in real use the cells are not shaded orange).

In principle, GrIDsure can be implemented on a grid of any reasonable size or shape; for the purposes of this study, however, we consider only the common use of a grid of 5X5 cells from which users choose ordered patterns of 4 cells and on which re-use of cells is allowed.

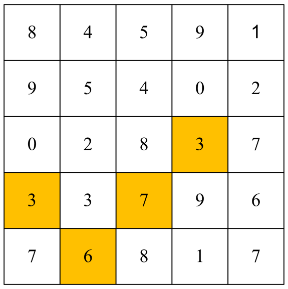
On authentication, if the digits 0-9 are used on a 5X5 grid, there will be some repeated numbers, and this is an important feature of GrIDsure security.

The system affords some protection from observation and replay attacks because *(i)* although a user's pattern is constant, the grid is randomised with each use, so the resulting PIN will be different each time; and *(ii)* there is always more than one possible pattern, on the randomised grid, that could have produced an observed PIN.

III. Related work

A. Graphical Passwords

From a usability point of view, the key advantage of graphical passwords over traditional knowledge-based authentication schemes is that they can offer cued, rather than unaided recall. Human memory performance with cued recall is significantly better than for unaided recall, particularly with infrequent usage [23]. Another advantage of graphical passwords is that psychology research has consistently found that pictures are recalled more readily than concrete words, and concrete words more readily than abstract [17].



8	4	5	9	1
9	5	4	0	2
0	2	8	3	7
3	3	7	9	6
7	6	8	1	7

Fig. 1: Entering a PIN (the cells are shaded for illustration only): 3, 6, 7, 3

Graphical passwords schemes, such as *PassPoints*, offer *cued recall*, typically involving users in recalling a specific target on an image, as presented in [3], [28]. Of the recognition-based schemes, perhaps the best-known graphical authentication scheme is PassFaces^{TM} [8], [20]. The most similar to GrIDsure is the recall-based Background Draw-a-Secret (BDAS) [10] scheme.

Graphical password schemes can produce high levels of maximum theoretical entropy; a DAS pattern in which the total length of the strokes is 11, for example, has a raw entropy of around 53 bits [15]. However, the number of "memorable" DAS patterns is considerably lower than this, as [15] show, and if "memorable" is assumed to mean "symmetric" then the size is lower still [26].

Unfortunately, if users are permitted to choose their own passwords, graphical passwords in general can end up being weaker than textual passwords because users choose predictable credentials to improve memorability. For instance, with PassFaces^{TM}, users prefer certain types of faces - what Monrose and Reiter termed as "beauty bias" [19]. On the other hand, there is evidence that, in certain configurations graphical passwords can be less vulnerable to shoulder-surfing than strong textual passwords [25].

B. Existing Research on GrIDsure

Weber [27] performed an analysis of the *statistical* security of GrIDsure.

Assuming a 4-cell pattern, the probability of randomly guessing the correct PIN by simply typing a PIN is 0.0001, as for any other 4-digit PIN. However, an attacker can gain a higher probability of success by entering the PIN that corresponds to a randomly chosen pattern. The probability of guessing the correct PIN in this way is 0.000342102; this is higher than that from simply guessing the PIN because not all PINs occur in the grid with the same probability [27].

This is a key point in the consideration of GrIDsure security, because the probability of guessing a PIN generated from a secret pattern - not of guessing the secret pattern itself, but a PIN which matches it - is greater than the probability of guessing a 4-digit static PIN. The additional security claimed for GrIDsure therefore rests on its resistance to capture of the transaction, together with the assertion that, unlike a static PIN, successful authorisation using a guessing method does not compromise the secret pattern.

The guessing probability can be minimised by choosing a grid calculated so that each digit appears as near as possible, an equal number of times, rather than strictly randomly chosen across the set of possible digits. For example, using digits 0-9, 5 digits appear exactly 3 times each and other 5 appear twice. This is called a "balanced grid". On a balanced 5X5 grid, the probability of guessing a correct PIN by entering a random pattern is 0.000116986 [27].

While a balanced grid makes random-guessing more difficult, it increases the risk from intercepted communications; having captured a PIN and grid, it is generally easier for an attacker to reverse-engineer the pattern with a balanced than with a random grid. We expand on this point later from our empirical work.

GrIDsure has been found to be easy to learn and the recall of patterns is acceptably reliable; however, as with other password schemes, the effective pattern space is far smaller than the maximum possible [7]. However, to understand the actual pattern space, simple assumptions are not sufficient [6]; as well as the shape, the order of cells and placement on the grid are important factors distinguishing between patterns. Although there are common patterns, these do not all occur with similar frequency [6], [7]. Brostoff et al. have developed a taxonomy of patterns, and our current work builds on this.

IV. Empirical Study of GrIDsure Security

We now re-consider the security and usability of GrIDsure from our empirical work. We are able to make an early estimate of the entropy of the GrIDsure's pattern space and give a far more thorough analysis of the risks from multiple captures than the rough figure of "2 in most cases" as suggested by Bond [6].

A. *The Actual Entropy of GrIDsure*

The maximum entropy of the possible pattern space for a 5X5 grid, from which users choose patterns of 4-cells is $\log_2(25^4) = \log_2(390625) = 18.5754$. This is considerably less than the 52 bits of entropy of a random 8-character password from a 95-character set, but comparable with a Draw-a-Secret password of length 4 strokes (which would be a very simple DAS password) [15].

However, the entropy of patterns actually chosen is lower than the theoretical entropy of the grid. From the patterns chosen by participants in our study described in section 6, we calculate a lower bound to the entropy, based on the calculation for a balanced estimator of the Shannon entropy:

$$\hat{H}_S^{bal} = \frac{1}{N+2}\sum_{i=1}^{M}\left[(n_i+1)\sum_{j=n_i+2}^{N+2}\frac{1}{j}\right]$$

from [5], where M is the number of patterns and N is the sample size. In our sample of 140 there were 102 distinct patterns, of which 78 were chosen once, 15 twice, 7 chosen 3 times, and 1 each 5 and 6 times. This gives a low entropy of 6.56, which suggests that GrIDsure may be rather easier to guess than it might first appear.

B. *Resistance of GrIDsure to Interception*

A capture of both PIN and Grid is possible in a Man-in-the-Middle (MiM) or shoulder-surfing attack. For shoulder-surfing, it is unlikely that an observer would be able to memorise the Grid at the same time as observing a PIN, but video recording would make this vulnerable. MiM could also be effectively carried out in the form of malware on the user's computer or a fake "Phishing" website. In this paper, we use MiM to refer to any situation where an attacker can capture both the grid and the user's PIN, and is therefore relevant to most systems even where there is reasonable security on the transmission channels.

In the case of traditional PINs or passwords, a single capture can be used by the attacker. In this limited sense, GrIDsure is an advance on traditional PINs. When GrIDsure is used as the authentication mechanism, the MiM can see the grid that the server sends to the user, keep a copy of it or change it and forward it to the user. In the same way, he can look at the user's response containing the one-time PIN which the user has read from the grid, and forward it to the server; he then has a copy of both the grid and the PIN.

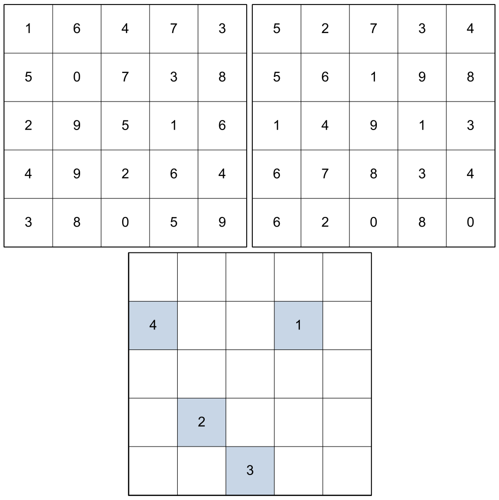
Weber [27] shows that with a 4-cell pattern on a 5X5 balanced grid, an attacker can find on average 45.6976 patterns for each entered PIN. This seems like a reasonable improvement over a static PIN, particularly if, as is usual, an account or card is blocked after a number of consecutive authentication failures. But what if an attacker is able to make multiple captures? In the case of a MiM, this is realistic; if a communication channel has been intercepted, the intercept is likely to remain in place. In the following section, we show that if the MiM can successfully capture the grid and user response on multiple occasions, reverse engineering will rapidly reduce the possibilities to 1 pattern.

1) Multiple Captures: An illustration: As an illustration, consider the case in which an attacker successfully captures the first grid displayed in Fig. 2 and the corresponding user response (one-time PIN) captured is {3, 9, 0, 5}.

In this grid, 5 digits 1, 2, 7, 8 and 0 repeat twice and the other five digits are repeated three times; the grid is a balanced grid. From the user's response {3, 9, 0, 5}, digits 3, 9 and 5 occur three times in the grid and 0 occurs twice; thus the number of possible patterns that could correspond to the first grid with {3, 9, 0, 5} PIN is $3^3 * 2^1 = 27 * 2 = 54$.

To aid in matching the patterns, the adversary considers a grid numbered 1 to 25, left to right, top to bottom, as a reference grid to compare all distinct patterns. Using the reference grid, it is now possible to construct a list of all 54 candidate patterns. Examples of these patterns, numbered as in the reference grid, are {9, 12, 7, 6} and {21, 25, 23, 24} . . . and so on up to 54.

Now, suppose the adversary captures the second grid and the corresponding PIN. In this case, suppose the user enters his PIN as {9, 7, 0, 5}. It so happens that the number of possible



1	6	4	7	3
5	0	7	3	8
2	9	5	1	6
4	9	2	6	4
3	8	0	5	9

5	2	7	3	4
5	6	1	9	8
1	4	9	1	3
6	7	8	3	4
6	2	0	8	0

4			1	
	2			
		3		

Fig. 2: First and Second captured grids in this example; below: Chosen pattern

patterns in this case is only $2^4 = 16$. As for the previous capture, the attacker lists all possible patterns for the second grid that gives the PIN {9, 7, 0, 5} using the reference grid; for example {13, 17, 23, 1}, {9, 3, 25, 6} . . . and so on up to 16.

We now have two sets of possible patterns - one from each grid making use of the user's input. Comparing both sets of patterns, retaining only those patterns from the first set that have at least one matching entry in the second set, only a small number of possible patterns remains. In fact, in this case there is only one pattern matching both the sets, so the pattern has been reverse-engineered with only two captures; it is {9, 17, 23, 6} (see Fig. 2).

2) Emulating MiM Captures Programmatically: The previous section provided an illustration, but to understand the real risks, we have investigated the process of multiple captures empirically or mathematically. We chose to use a simple Monte Carlo technique to build grids programmatically and "capture" the PIN and grid in order to find the number of captures needed to reverse-engineer a pattern with certainty.

Having "captured" a PIN and grid, the program then simulates further captures of the same pattern (the grid and the corresponding PIN are obviously different), each time generating the set of patterns corresponding to the captured PIN and grid. In each iteration, the program also matches the patterns from the previously generated set, as described in the previous section.

The average number of patterns that match a captured PIN i.e. those which match all captured patterns in a trial are given in Table 1. Note that the average number of matches in a capture rises after 4 captures, but there are few trials that actually reach this number - most patterns have been found with fewer captures. We do not show the figures for later captures, for reasons of space. As expected, the average number of matching patterns in a capture from a random grid is far higher than from a balanced grid; this makes the attacker's job harder in terms of reverse-engineering the pattern but it is easier to make a successful random guess.

TABLE I: Numbers of matches after multiple captures

Iteration	Average matches	
	balanced grid	random grid
1	45.7019	146.6415
2	1.5605	2.5489
3	1.1003	1.1936
4	1.0874	1.1231
5	1.0910	1.1306

TABLE II: Captures to reverse-engineer a pattern

Found after captures	balanced grid	random grid
1	0	129
2	675282	422230
3	299699	496400
4	23304	72871
5	1599	7527
6	108	770
7	6	69
8	3	4

TABLE III: Expected and observed probabilities of matching patterns per PIN

Number of matches	Calculate Probability	Observed in 1000000 trials
16	0.0256	25661
24	0.1536	153611
36	0.3456	344925
54	0.3456	345995
81	0.1296	129808

From Table 2 also observe that, although there are a few reverse-engineered patterns using a random grid on the first capture while none with a balanced grid, the number of patterns that can be reverse-engineered with a random grid (42.2%) is far fewer when compared to a balanced grid (67.5%) with only 2 captures. The number of captures required with a random grid are higher from third and more captures.

By running our program over 1000000 simulated attacks, we obtained the following results for balanced and random grids:
Average number of captures to reverse-engineer a pattern:
Balanced grid: 2.3516; Random grid: 2.6680
Maximum: Balanced grid: 8; Random grid: 8

We already know, the expected number of PINs with patterns which match a PIN entered by a user, from Weber's [27] work; from our program, we are also able to make an estimate of the probable number of matches on multiple captures. After running our program for 1000000 trials, we generated a mean of the number of matching patterns. At the first capture, for a balanced grid this was 45.7019, similar to the expected value derived theoretically by Weber [27]. The different numbers of possible matches occur with different frequencies. The probabilities for each possible number of matches, which we have calculated using Weber's method of "templates" of pattern types, and observed in our simulation,

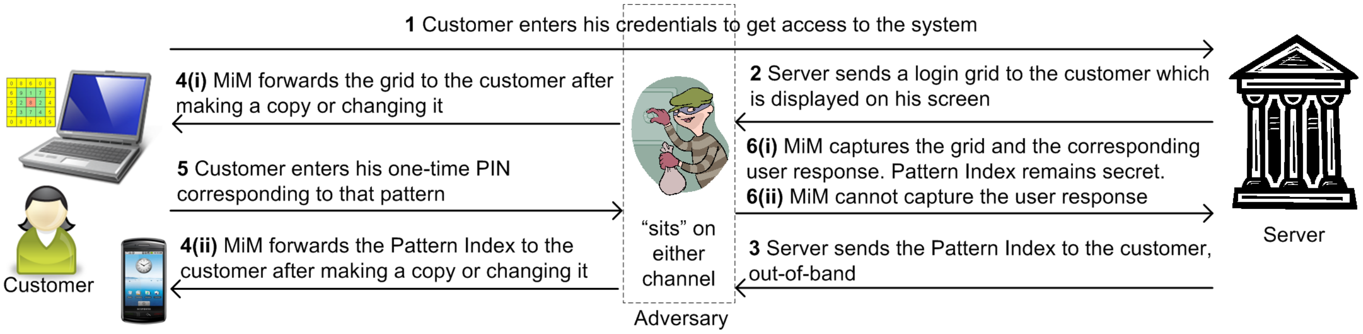

1 Customer enters his credentials to get access to the system
2 Server sends a login grid to the customer which is displayed on his screen
4(i) MiM forwards the grid to the customer after making a copy or changing it
6(i) MiM captures the grid and the corresponding user response. Pattern Index remains secret.
6(ii) MiM cannot capture the user response
5 Customer enters his one-time PIN corresponding to that pattern
"sits" on either channel
4(ii) MiM forwards the Pattern Index to the customer after making a copy or changing it
3 Server sends the Pattern Index to the customer, out-of-band
Customer
Adversary
Server

Fig. 3: Sending pattern index OOB when there is a risk of MiM attacks

are shown in Table 3. The closeness of the observed figures to their expected values indicates that our program is operating correctly. Note that we generate the "user's" pattern randomly for each trial, since not all patterns are equally easy to reverse-engineer.

So far, we have only modeled an attacker who assumes that all matching patterns are equally probable. If an attacker guesses patterns which are known to occur with higher probability, part of the subject of our further study, then the number of captures needed to reverse-engineer a pattern, which is already small, would be reduced further.

V. Fortifying GrIDsure against MiM

We have shown that GrIDSure is not resistant to MiM-type attacks since *(i)* patterns can be cracked with only a small number of captures and *(i)* the actual entropy is much lower than the theoretical entropy. In this section we present our modified system that greatly increase the resistance of GrIDsure to MiM and similar attacks.

A. Enhancement 1

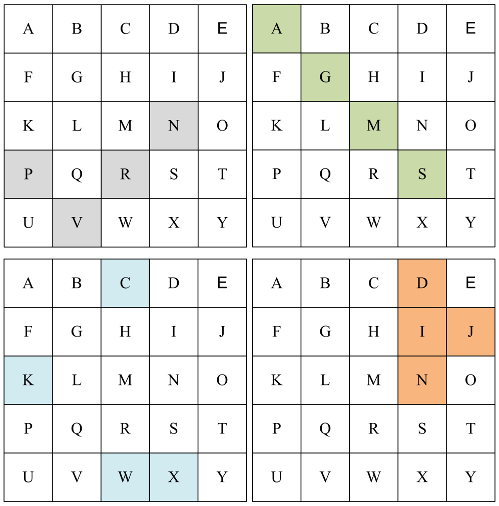
In our proposed enhancement, Users choose and register multiple (different) patterns with his account/username, as shown in Fig. 4; for clarity, we show the grid in alphabetic order, although in actual implementation it is preferable for the enrollment grid to be ordered randomly, to prevent users from using guessable words as a form of pattern. Although, in principle, a system can implement our solution using any number of patterns, use of 4 patterns seem to be a reasonable balance considering that average users are now registered to more than 20 different accounts and have difficulties in managing their credentials [11].

Each time the user tries to login using our proposed system (GS4), he is informed which one among his registered patterns to use (the "pattern index") for successful authentication, using an Out-Of-Band (OOB) technique like sending an SMS to the user's mobile phone, as shown in Fig. 3.

With the use of GS4, unable to intercept the OOB channel, the attacker has no idea against which pattern a capture is to be matched. Simply comparing multiple captures will no longer reduce rapidly to a single matching pattern with a small number of captures.

Suppose an attacker has captured two PIN and grid transactions by intercepting the channel where the user enters his login details (steps 4(i) and 6(i) in Fig. 3). Of course, if the pattern index is the same for both captures (although the attacker cannot know this), then there must be at least one pattern that matches on both grids. However, this might happen even if the two patterns are different. An attacker finding that two or more captures match one unknown pattern could guess that all of the captures correspond to one pattern, but he cannot be sure.

On the other hand, with a large enough set of captures, the attacker can certainly *reject* some of the potential matches; comparing captures resulting from patterns which correspond to different pattern index will rapidly reduce to zero matching patterns. However, to find these sets of "non-matching" captures, every capture has to be compared with every other capture, and this still does not provide any patterns which are known with certainty. Eventually, by eliminating these non-matches, the attacker can build up a set of probable patterns, but note that the attacker still cannot know the corresponding pattern indexes.



A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Fig. 4: Registering patterns in the proposed system. Pattern 1: PVRN Pattern 2: AGMS Pattern 3: KCWX Pattern 4: IDNJ

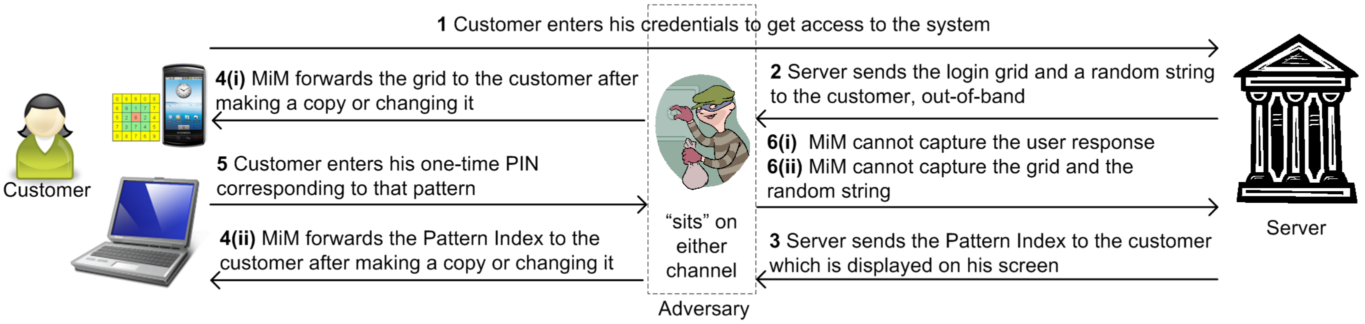

1 Customer enters his credentials to get access to the system
4(i) MiM forwards the grid to the customer after making a copy or changing it
2 Server sends the login grid and a random string to the customer, out-of-band
6(i) MiM cannot capture the user response
6(ii) MiM cannot capture the grid and the random string
5 Customer enters his one-time PIN corresponding to that pattern
Customer
"sits" on either channel
Server
4(ii) MiM forwards the Pattern Index to the customer after making a copy or changing it
3 Server sends the Pattern Index to the customer which is displayed on his screen
Adversary

Fig. 5: Sending the login grid OOB when there is a risk of MiM attacks

If the pattern index generation does not follow a random distribution and is instead fabricated to ensure non-repetition of consecutive index sequences; assuming the system registers four patterns with each user account, the best that can be said is that the attacker can learn all the four patterns after 8 captures but still cannot know the pattern indexes. If all the patterns are known by the attacker, the probability of successfully entering a PIN in response to a challenge from the server is 0.25 on each attempt. In this case, clearly the number of attempts the server allows to login to the system becomes a critical parameter of consideration. If a user is allowed 3 attempts, the attacker has ≅ 0.75 probability of getting in to the system; this reduces to 0.25 if only one attempt is allowed, and if 4 attempts are permitted then the entire set of 4 patterns would be effectively compromised.

B. Enhancement 2

In a further enhancement involving Out-of-Band communications, another parameter, such as a random one time string, is also sent to the user's mobile phone along with expected pattern number. The user reproduces this string during login. Assuming that the attacker cannot control both communication channels, the attacker will never be able to login to the system.

It is worth a note that OOB cannot be used standalone to authenticate users. If it is not used in conjunction with a knowledge-based authentication mechanism like GS4, the security of the system reduces only to the physical security of the device. For example, authentication in a conventional username/password scheme where the user reproduces the password sent as an SMS to his mobile phone is only based on "something you have". If the mobile phone is compromised, the attacker can easily gain access to the system. In contrast, our proposed system offers security of a higher magnitude i.e. of both "something you have and something you know", providing a two-factor authentication.

A variation of these approaches is shown in the Fig. 5, where, instead of sending the expected pattern number to the mobile phone, the server can send the login grid to the mobile phone and display the expected pattern number on the screen where the user is expected to type his details.

In this case, it becomes almost impossible for the MiM to be able to capture the grid and the user response both. If the attacker "sits" on the communication channel between the server and the login terminal, he will learn only the one time PIN - i.e. the user's response and the expected pattern number (steps 4(ii) and 6(ii) in Fig. 5). If the attacker "sits" in the communication channel between the server and the mobile phone, he can see the grid which the server sends to the user, which in any case changes each time, but cannot capture the user's response (steps 4(i) and 6(i) in Fig. 5). This defeats the MiM attack, as it is highly unlikely that the attacker can control both the web and phone channel. The only drawback with this method is that the user must have a mobile phone that is capable of displaying the grid.

Since the server initiates the transmission, this channel is not at risk of being controlled by an attacker (even though it could itself be intercepted or overseen). However, the OOB channel could be any convenient form of electronic communication, which makes our proposal extremely flexible for use at, for example, an ATM.

VI. Usability of multiple patterns - an evaluation

Whilst the use of multiple patterns would fortify GrIDsure against MiM and shoulder-surfing attacks, this is not a practical solution unless it is possible for users to recall multiple patterns. Studies of other graphical authentication schemes have shown that adding a second graphical password significantly reduces the number of correct recalls - e.g. for Passfaces^{TM} [26]. There is also evidence that multiple graphical passwords produce interference problems similar to those known in textual passwords [11]. Therefore, to investigate the usability of our proposals, we conducted a field trial evaluation, using the Authentication Performance Evaluation Tool (APET) online web-based tool described in [2].

A. Methodology

We used a Chi-squared test to decide the number the participants to be recruited for the study so as to obtain the best results. The results of the test suggested the value of N = 26. We recruited 30 people to allow for a 15% dropout rate.

30 participants from varying age groups and education levels were recruited over a three-week period. None of the participants had any previous experience of using GrIDsure.

Participants were divided into two groups. Each participant was assigned to make 15 login attempts over a three-week period during the experiment; this was done entirely using

TABLE IV: Reliability of recall of GrIDsure patterns

		Successful Login (%)		Complete Failure(%)
		>1 attempt	first attempt	
Group A	GS1	2.77	95.83	1.38
	GS4	6.94	90.20	2.77
Group B	GS4	6.77	88.98	4.23
	GS1	1.50	98.50	0
Overall	GS1	2.22	97.03	0.74
	GS4	6.87	89.69	3.44

email and web. Participants in Group A used GrIDsure with one pattern (GS1) for first five logins and then used GrIDsure with four patterns (GS4) for the subsequent ten logins, whereas Group B used GS4 for the first ten logins and GS1 for the subsequent five. In this way, we avoided bias between the groups since each group used both designs, and we avoided bias within groups since the groups used the designs in different orders.

At the start of the trial, participants were sent an email requesting that they register their pattern(s) and then login using their first or only registered pattern (depending on the group) in the same session. The first three emails also included the instructions on the usage of the scheme, as initial training. All subsequent emails provided only those details necessary to login (no instructions). Participants were sent 4-5 emails a week over the three week experimentation period.

A well-known issue in the design of studies in usable authentication is that, in the real world, authentication is not users' primary task; they are using the authentication mechanism only to gain access to some service. For this reason, we devised a study in which the authentication was considered as the secondary, rather than the primary, task.

We used the *Barter World* scenario described in [2]. In this game, participants complete services - Gardening, Babysitting, Cleaning, or Teaching - for the community, and in return receive tokens for the appropriate working hours. (Participants did not actually have to perform the services, but they did have to log the hours worked to their personal account, protected by GrIDsure authentication, to claim their payment.)

As in [7], [11], participants were compensated with the exchange of tokens for gift certificates, at the rate of £1.33 for every successful login and £1.20 otherwise, giving a guaranteed minimum for participation up to a maximum of £20.

The community manager sent them an email when a barter task "had been completed". The email included a hyperlink and the pattern number (index) with which to authenticate, and an additional random string. If a user failed to authenticate within three attempts, the authentication server sent another email containing a hyperlink to the registered pattern; this simulates a real-world "password reset". If a participant failed to attempt authentication before midnight, they could no longer authenticate and log the claim.

The APET system [2] records *(i)* the time taken to login, *(ii)* number of login attempts, *(iii)* whether or not the attempt was successful, *(iv)* the IP address, and, if more than one attempt was required, *(v)* the PIN entered and what it should have been.

Following a successful GrIDsure authentication, participants entered a "claim code" consisting of the random string contained in the email. This implements the random out-of-band data suggested in section 5.2.

B. Results

As in Brostoff et al.'s study [7], user performance results are encouraging. All participants in Group A and 14/15 participants in Group B were able to login successfully.

In Group A, during the 75 usages (5 each by 15 participants) in the GS1 phase of evaluation, there were 3 occurrences of participants failing to respond to emails before midnight and hence of the request expiring. During the 150 usages - 10 each - of the GS4 phase, there were 6 occurrences of email expiry. In Group B, 2 participants discontinued the study. Of the 65 usages in the GS1 phase (which was completed after the GS4 phase for Group B) - 5 each by the remaining 13 participants - there were 2 occurrences of email expiry and 12 during the GS4 phase.

Excluding these non-attempted authentications, the results are shown in Table 4.

A failure rate of 3.44% would not be considered good in ordinary password use, but for an initial encounter with an unfamiliar mechanism, the performance is encouraging.

An important result, for the validating the usability of our proposal, is that there is no significant difference in user performance between GS1 and GS4.

We consider only 13 participants in each of groups A and B (13+13=26) as required by the Chi-squared test. This gives 127 of 128 successful logins within 3 attempts using GS1 and 237 of 246 successful using GS4.

Applying a χ^2 test, we find χ^2(df=1, Yates' correction) = 1.68, p=0.194. However, since the expected value of failed logins using GS1 is less than 5, which suggests that χ^2 may be unreliable, we also apply Fisher's exact test which gives a one-sided significance of p=0.091 (>0.05) i.e., not statistically significant, although it's not strong enough to say that GS4 is as easy as GS1. Intuitively, from our results, GS4 is at least a bit harder than GS1, but we can infer that we have not found it to be *significantly* less easy.

This finding is surprising - given that studies have found a clear interference effect for multiple passwords in other graphical authentication schemes [11] - and encouraging. However, GrIDsure is quite different from the scheme used in Everitt et al.'s study, which was based on recognition of faces, similar to PassFaces^{TM} [20]. In addition, factors such as frequency of use and length of use are known to have important impacts on the usability of passwords generally.

VII. Conclusions and Future Work

In this paper, we present empirical results which confirm the assertions of Brostoff et al. [7] and Bond [6], that GrIDsure is not resistant to multiple captures of the grid and PIN. We have also developed enhancements to GrIDsure which fortify it against Man-in-the-Middle and similar attacks.

From our user study we did not find the enhanced GrIDsure, with 4 personal patterns and Out-of-Band secrets, to be significantly less usable than simple GrIDsure.

Given the documented problems with interference between different graphical passwords just as for textual ones [11], it might be more usable to remember 4 patterns across a number of services than different patterns for each service. We agree with [7] that the re-use of patterns across different services is insecure in general. However, if it can be shown that our enhancements add sufficient security to enable the safe use of patterns across multiple services, it is possible that our solution will actually improve the overall usability of GrIDsure.

We have also completed a longer-term study of the usability, interference and memorability of GrIDsure with 4 patterns. Our results here have confirmed the importance of detailed research on actual user behaviour as the basis for the design of emulation experiments and estimates of actual entropy.

Therefore, our future work will provide a more detailed study of the effective entropy of GrIDsure based on the taxonomy of patterns which we are developing from our empirical studies. This taxonomy will also enable us to enhance our system using Monte Carlo technique, by emulating patterns actually chosen rather than random ones. We will also emulate an attacker who guesses similar patterns, or who uses the kinds of "clever" guessing methods suggested by Weber [27] by choosing the more frequently occurring digits from a grid. Finally, a more complex algorithm will enable us to emulate grid and PIN capture in our enhanced GrIDsure with 4 patterns.

References

[1] A. Adams, M. A. Sasse, and P. Lunt. Making Passwords Secure and Usable. *People and Computers XII: HCI 97*, 1997.

[2] A. Beautement and M. A. Sasse. Gathering Realistic Authentication Performance Data Through Field Trials. *Usable Security Experiment Reports (USER) Workshop, Symposium On Usable Privacy and Security*, 2010.

[3] R. Biddle, S. Chiasson, and P. C. van Oorschot. Graphical Passwords: Learning from the First Generation. *Technical Report TR-09-09*, 2009.

[4] R. Biddle, S. Chiasson, and P. C. van Oorschot. Graphical Passwords: Learning from the first twelve years. *ACM Computing Surveys*, 2011.

[5] J. A. Bonachela, H. Hinrichsen, and M. A. M. noz. Entropy estimates of small data sets. *Journal of Physics A: Mathematical and Theoretical*, 41(20):1–9, 2008.

[6] M. Bond. Comments on Gridsure Authentication. 2008.

[7] S. Brostoff, P. G. Inglesant, and M. A. Sasse. Evaluating the usability and security of a graphical one-time PIN system. *Conference on Human Computer Interaction BCS*, 2010.

[8] S. Brostoff and M. A. Sasse. Are Passfaces more usable than passwords? A field trial investigation. *HCI 2000 - People and Computers XIV - Usability or Else! BCS*, 2000.

[9] J. Cohen. Quantitative Methods in Psychology, A Power Primer. *New York University*.

[10] P. Dunphy and J. Yan. Do Background Images Improve Draw a Secret Graphical Passwords? *Conference on Computer and Communications Security*, pages 36–47, 2007.

[11] K. M. Everitt, T. Bragin, J. Fogarty, and T. Kohno. A Comprehensive Study of Frequency, Interference, and Training of Multiple Graphical Passwords. *27th International Conference on Human factors in Computing Systems*, 2009.

[12] D. Florêncio and C. Harley. A Large-Scale Study of Web Password Habits. *Proceedings of WWW 2007 (Banff, Alberta, Canada*, May 2007.

[13] GrIDsureLimited. http://www.gridsure.com.

[14] P. G. Inglesant and M. A. Sasse. The True Cost of Unusable Password Policies: Password Use in the Wild. *28th International Conference on Human Factors in Computing Systems (CHI 2010)*, 2010.

[15] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin. The Design and Analysis of Graphical Passwords. *8th USENIX Security Symposium*, 1999.

[16] D. V. Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. *Second USENIX Workshop on Security*, pages 5 – 14, 1990.

[17] S. A. Madigan. Picture memory. *Imagery, Memory, and Cognition, Yuille, J. C. (ed.)*, 1983.

[18] L. Marks. Between Silk and Cyanide: A Codemaker's Story 1941-1945. *HarperCollins, London, UK*, 2000.

[19] F. Monrose and M. K. Reiter. Graphical Passwords. Chapter 9 in Security and Usability: Designing Secure Systems That People Can Use, Cranor, L. F. and Garfinkle, S. (eds.), O'Reilly, Sebastopol, CA, USA; Cambridge, UK. pages 161 – 179, 2005.

[20] passfaces.com. http://www.passfaces.com.

[21] One in five use birthday as PIN number. http://www.telegraph.co.uk/finance/personalfinance/borrowing/creditcards/8089674/OneinfiveusebirthdayasPINnumber.html.

[22] A. Polyviou. The impact of interference and frequency of use on the performance of three authentication mechanisms. *Masters thesis (unpublished), University College London*, 2010.

[23] M. A. Sasse, S. Brostoff, and D. Weirich. Transforming the 'Weakest Link' - a Human/Computer Interation Approach to Usable and Effective Security. *BT Technology Journal*, 19, July 2001.

[24] E. M. Tamil, A. H. Othman, S. A. Z. Abidin, M. Y. I. Idris, and O. Zakaria. Password Practices: A Study on Attitudes towards Password Usage among Undergraduate Students in Klang Valley, Malaysia. *Journal of Advancement of Science & Arts*, 3:37–42, 2007.

[25] F. Tari, A. A. Ozok, and S. H. Holden. Comparison of Perceived and Real Shoulder surfing Risks between Alphanumeric and Graphical Passwords. *Symposium on Usable Privacy and Security (SOUPS 06), ACM Press*, 2010.

[26] J. Thorpe and P. C. van Oorschot. Graphical Dictionaries and the Memorable Space of Graphical Passwords. *13th USENIX Security Symposium*, 2004.

[27] R. Weber. The Statistical Security of GrIDsure.

[28] S. Wiedenbeck, J. Waters, J. C. Birget, A. Brodskiy, and N. Memon. PassPoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies*, (63):102 – 107, 2005.

[29] M. Zviran and W. J. Haga. Password Security: An Empirical Study. *Journal of Management Information Systems*, 4(15):161–185, 1999.