Panic Passwords: Authenticating under Duress

Jeremy Clark*
School of Computer Science
University of Waterloo
j5clark@cs.uwaterloo.ca

Urs Hengartner*
School of Computer Science
University of Waterloo
uhengart@cs.uwaterloo.ca

ABSTRACT

Panic passwords allow a user to signal duress during authentication. We show that the well-known model of giving a user two passwords, a 'regular' and a 'panic' password, is susceptible to iteration and forced-randomization attacks, and is secure only within a very narrow threat model. We expand this threat model significantly, making explicit assumptions and tracking four parameters. We also introduce several new panic password systems to address new categories of scenarios.

1. INTRODUCTORY REMARKS

As important services and sensitive data congregate online, attackers have an increasing incentive to obtain the passwords that protect these services and data. Panic passwords are a mechanism to allow a user to use a special type of password to signal to the server that her password is being entered as the product of a coercive action. While panic passwords are currently used in home security systems to trip a silent alarm, they could also find application online. For example, a shortcoming of Internet-based voting is the move from a private voting booth to an open environment where voters could be coerced or bribed to demonstrably vote a certain way. A panic password could allow a voter to cast a coerced ballot as if it were her real vote, while in reality the ballot is spoiled by the server.

This paper is motivated by a deficit of academic literature exploring the topic of panic passwords, the underlying threat models panic passwords address, and possible schemes to mitigate the threat while being usable. In particular, it is perturbing that the best-known example of a panic password scheme, one where the authenticator knows two passwords: a 'regular' one and a 'panic' one, is very easily defeated by coercing the victim to authenticate twice using different passwords each time.

Contributions: This paper makes the following new contributions to the academic literature on panic passwords:

- a thorough threat model for categorizing scenarios of coercion and undue influence,
- the application of an iteration and forced-randomization principle to panic passwords,
- the precise criteria for using the known scheme `2P`,
- the introduction of three new panic password schemes: `2P-lock`, `5-dictionary`, and `5-click`.

To make our contributions concrete, we will consider several representative scenarios where panic passwords could be employed. Our ultimate interest, however, is in Internet-based voting, which has been widely used in Estonia and in a limited capacity in Australia, Switzerland, and Canada. We will examine the potential of panic passwords to mitigate the *improper influence problem*: that removing the privacy of the voting booth opens the voting process up to the possibility of voter coercion and vote buying schemes.

2. RELATED WORK

Panic passwords are alternatively referred to as *distress passwords* or *duress codes* in the academic, commercial, and military literature. Due to their applicability in military and intelligence scenarios, it is not possible to determine the exact extent to which panic password schemes have been studied, as any such work would be classified. However a survey of the non-classified literature reveals very little.

Panic password schemes show up in patented technology, usually as one feature of a larger system. The exact method is often not included. A selection of patents include: ATMs which display a list of words from which the user may choose a predefined one for normal authentication and any other word to signal duress and to limit the amount of cash available for the transaction [12]; a home security system that incorporates a basic panic password scheme [9]; the use of panic passwords to authenticate over a network where panic mode relays the user to a different database than the normal one [8]; the use of a panic password to keep data protected on a mobile device [10]. Chaum mentions duress codes in passing for preventing coerced transactions using a credential [5]. In a policy recommendation, the *Foundation for Information Policy Research* suggest that files hidden on a hard-disk (using a steganographic file system [2]) could be erased upon entering a panic password [1].

3. THREAT MODEL

Each of the scenarios considered herein involves three participants: Alice, Bob, and Oscar. Alice is typically a human

*The authors acknowledge the support of this research through an NSERC (Natural Sciences and Engineering Research Council of Canada) Canada Graduate Scholarship and an NSERC Discovery Grant, respectively.

and wishes to communicate, typically for the purpose of authentication prior to gaining access to a resource, to Bob, who may be human, a server, or a software application. Oscar is an opponent in the system and he attempts to coerce Alice's communications with Bob in order to benefit his nefarious purposes. We will also consider less coercive scenarios where Alice wishes to sell her access rights to Oscar. We assume that Bob is a trusted entity and, in particular, is not in collusion with Oscar.

Consider the password space P, where each $p \in P$ is in one of two sets, V for valid or I for invalid. As in any password scheme, a regular password p_0 is selected from P and is included in V. Entering an invalid password results in an error (observable to Alice and Oscar) and has no further consequence. To extend the notion of passwords to panic passwords, we define the following concepts.

DEFINITION 1. ***Panic Password**: a covert communication from Alice to Bob over an observed channel indicating an abnormal state. We define a panic password as $p^* \in P$ and include it in V. A scheme may include more than one panic password. All elements of P other than p_0 and the panic password(s) are invalid.*

DEFINITION 2. ***Unobserved Reaction**: in reaction to receiving a p^* from Alice, Bob may engage in an unobserved reaction $q^* \in Q$, where Q is the set of all such reactions and $q_0 \in Q$ is the regular (non-panic) reaction. This reaction is unobservable to Alice or Oscar. A scheme will include a single regular reaction and any number of panic reactions (i.e., none, one, or multiple reactions).*

DEFINITION 3. ***Observable Response**: in reaction to receiving p^* from Alice, Bob may respond to Alice with a panic response $r^* \in R$, where R is the set of all responses and $r_0 \in R$ is the regular response. This response is observable to Alice and Oscar in principle, however it is not necessarily distinguishable to Oscar which response is which. A scheme will include a single regular response and any number of panic responses.*

Consider again Internet-based voting. To authenticate to the voting service (Bob), assume Alice has a set of two passwords: a normal password and a single panic password, $\{p_0, p^*\}$. Entering p_0 will cause Alice's vote to be cast correctly, while entering p^* could trigger an unobserved reaction from Bob, q^*: for example, Bob could mark the ballot as spoiled or could alert the authorities. Entering p^* could also trigger an observable response r^*: Bob may falsely inform Alice that her vote has already been cast and she may not modify it, or he may report that the server is down and unable to accept votes at this time. In other areas where panic passwords are used, an unobserved reaction could be a silent alarm, while an observable response may be modified access control permissions or an upper limit on funds available for a financial transaction.

A typical transaction will take the form,

$$A \rightarrow B \quad : \quad p \in P \tag{1}$$

$$B \quad : \quad q \in Q \tag{2}$$

$$A \leftarrow B \quad : \quad r \in R \tag{3}$$

3.1 Security Assumptions

Our threat model is based on four pragmatic assumptions.

Parameter 1	Parameter 2
– persistent	– $\{p^*, q^*, r_0\}$
– non-persistent	– $\{p^*, q_0, r^*\}$
	– $\{p^*, q^*, r^*\}$

Parameter 3	Parameter 4
– $G_O[\forall q \in T : \neg q^*]$	– screening
– $G_O[\exists q \in T : q_0]$	– signaling
– $G_O[\forall r \in T : \neg r^*]$	
– $G_O[\exists r \in T : r_0]$	

Table 1: Summary of parameters to the threat model.

Assumption 1. **Kerckhoffs' principle**: The details of the authentication system that is in place is public information and known to all participants, including Oscar. Alice and Bob retain a shared secret (*e.g.,* a password or set of passwords) that is both private and the foundation upon which the security of the system rests.

Assumption 2. **Observational principle**: Alice communicates with Bob over a semi-private channel that can be observed by Oscar prior to the password being secured (*i.e.,* encrypted or hashed). Namely, Oscar can observe the password(s) used by Alice and could even enter in Alice's password himself, after coercing her into revealing it.

Assumption 3. **Iteration principle**: Unless explicitly prevented by the underlying system, Oscar is not bound to a single instance of coercion against Alice. He may force Alice to authenticate multiple times. Combined with the observational principle (Assumption 2), Oscar can force Alice to use a different password each time.

Assumption 4. **Forced-randomization principle**: Oscar can choose to eliminate any strategy Alice may employ through the order in which she reveals the passwords she knows. For example, Oscar may force Alice into writing down a set of passwords so that he can randomly choose the order in which to iterate through them. The assumption is that this option is available to Oscar, not that he will necessarily employ it.

3.2 Threat Parameters

The underlying threat model is dependent on the scenario where a panic password system is employed. We identify four parameters along which the threat model could vary. They are summarized in Table 1.

Parameter 1. **Oscar's Persistence**: We can define the period of time under which Alice is expected to be coerced by Oscar as beginning at t_0. In some scenarios, Oscar may be *non-persistent* and limited in time to t_1, as in the example of Oscar coercing Alice into withdrawing funds from an ATM. In other, typically less-threatening, scenarios, Oscar may be *persistent* in his coercion for an arbitrarily long period of time, as in the example of a employer influencing the vote of her employee. A panic password scheme that disables access to an account could deter a non-persistent attacker if the account were disabled for a period of time greater than t_1 but this measure would not effectively deter a persistent attacker.

Parameter 2. **Bob's Reaction and Response**: We can summarize Equations 1 to 3 by defining a transaction as a tuple of parameters, such as $\{p_0, q_0, r_0\}$ to mean Alice submits non-panic communication and Bob reacts with the regular reaction and response. If Alice presents a panic password, Bob has three options. He could perform an unobserved reaction but not change his observable response, $\{p^*, q^*, r_0\}$; he could modify his response but not commit any unobserved reaction, $\{p^*, q_0, r^*\}$; or he could do both, $\{p^*, q^*, r^*\}$.

Parameter 3. **Oscar's Goal**: Oscar's goal, denoted G_O, in coercing Alice may take different forms. His proximate goal is for Alice to enter p_0 but since he reserves the option of iteration, we must distinguish between a few scenarios. These distinctions are dependent on his ultimate goal with respect to Parameter 2.

If Bob performs a panic reaction to a panic password, Oscar could have one of two goals: to prevent this unobserved reaction from ever occurring in the set of transactions T, $G_O[\forall q \in T : \neg q^*]$; or to achieve at least one regular reaction, $G_O[\exists q \in T : q_0]$. If Oscar is limited to a single transaction, $|T| = 1$, these goals are equivalent. However under the Iteration Principle, Oscar may force Alice to authenticate more than once and these goals are no longer equivalent when $|T| > 1$. If q^* were a silent alarm, Oscar would have the first goal of preventing any occurrence of q^*. However if q^* was the reaction of spoiling a ballot in an Internet-based election, Oscar may have the second goal: he could force Alice to re-authenticate many times with different passwords and vote each time, hoping that eventually she enters p_0 and this transaction will be counted as a valid vote.

If Bob responds differently to a panic password than to a regular password in parameter 2, Oscar also has one of two goals: to prevent any observable panic response $G_O[\forall r \in T : \neg r^*]$ or to achieve at least one regular response $G_O[\exists r \in T : r_0]$. As an example of the first, consider the case where Alice has data hidden in an encrypted partition on her harddrive. The regular response could reencrypt this data and then provide access to it, while a panic response could overwrite the most sensitive data with random data and provide access to only the less sensitive data. Oscar's goal would be to prevent a panic response, as he cannot recover from such a state. Alternatively, a panic response could be less permissive access control permissions than is regular, and Oscar's goal will be to apply iteration until he gets full access.

Parameter 4. **Screening vs. Signaling**: A proper panic password scheme should cause Alice's use of a panic password to be indistinguishable by Oscar from use of her valid password. If Oscar coerces Alice into using her regular password, he succeeds by being able to distinguish (or convince Alice he can distinguish) the use of a panic password. Alice succeeds by not having to enter her regular password. Any mechanism that allows Oscar to separate panic and regular passwords is called a *screen*. Alternatively, Alice may want to prove to Oscar she is not using a panic password. Here Alice's goal is to demonstrate to Oscar that she is entering a regular password, and any mechanism she can use to accomplish this is called a *signal*. In Internet-based voting, voter-coercion is based on screening, whereas vote-selling is based on signaling. Although there is no duress in vote-selling schemes, the availability of panic passwords to prevent voter-coercion offers the additional feature that Oscar cannot determine if he is actually buying a legitimate vote from Alice, who could cheat him with her panic password. Thus, signal prevention is often a free additional property of panic password schemes.

4. ILLUSTRATIVE THREATS AND THEIR PREVENTION

We now consider several illustrative scenarios where panic passwords can be applied. These scenarios were chosen to represent a class of specific threat models to which a particular panic password scheme applies.

4.1 Unrecoverable Reactions

Consider a scenario with panic communication $\{p^*, q^*, r_0\}$ such that $G_O[\forall q \in T : \neg q^*]$. Our approach to this problem is invariant with respect to the persistence of Oscar or the prevention of signals as opposed to screens. The approach also applies in an equivalent manner to the scenarios: $\{p^*, q_0, r^*\}$ such that $G_O[\forall r \in T : \neg r^*]$ and $\{p^*, q^*, r^*\}$ such that $G_O[\forall q, r \in T : \neg(r^* \vee q^*)]$.

Working Example: Consider the scenario where Alice is an employee at an office building with a security alarm that must be deactivated with a numerical password (entrance code) upon entry. Alice wants to communicate a forced entry which alerts the alarm company or law enforcement (q^*) but still grants Oscar access to the building (r_0). Oscar's goal is to prevent the silent alarm as he cannot escape on time to avoid detention.[1]

4.1.1 Solution

Security alarms on the market already handle this through a system we will call `2P` for two passwords.

Scheme 1. `2P`. Alice knows two passwords $\{p_0, p^*\}$. For normal authentication purposes, Alice uses p_0. To communicate panic, she uses p^*.

In the working example, Alice could use a four digit PIN, $d_0d_1d_2d_3$, as p_0 and invert the order of the last two digits to construct p^*: $d_0d_1d_3d_2$. Under our first security assumption, Kerckhoffs' principle, Oscar is assumed to know the structure of the system while not knowing p_0 or p^*. From the observational principle, Oscar can observe Alice enter a password, p_1, but cannot determine if it is p_0 or p^*. If access is granted, Oscar can learn that it was one of the two passwords. Since Oscar's goal is to prevent the silent alarm, q^*, forcing Alice to enter a second password after entering the first (assumption 3) will only assure he does not achieve his objective. Under these three security assumptions, the `2P` scheme succeeds at mitigating the threat.

The fourth assumption, however, presents a problem. Since Oscar knows the scheme being used, he can ask Alice to reveal the two passwords she knows. He cannot distinguish them but instead of him allowing Alice to enter the password of her choosing, which presumably would be p^*, he can randomly choose one. This allows him to achieve his

[1]To demonstrate why Oscar's goal is $G_O[\forall q \in T : \neg q^*]$ and not $G_O[\exists q \in T : q_0]$, consider a situation where Oscar forces Alice to authenticate twice, once yielding q^* and once q_0. This outcome is unacceptable to Oscar if he has the former goal but is acceptable for the latter goal. If q^* is a silent alarm, this outcome is unacceptable and thus Oscar must not have the latter goal in this scenario.

goal half of the time on average. However this probability could be reduced by having a larger set of panic passwords, as will be examined below, and the significant probability of being caught that 2P provides is likely a large enough deterrent for scenarios in this class.

4.2 Non-persistent Attacks

Consider a scenario where Oscar has a non-persistent influence and his goal is either $G_O[\exists q \in T : q_0]$ or $G_O[\exists r \in T : r_0]$. This scenario differs from the previous one since Oscar is also not attempting to prevent a q^* or r^*, only to gain a clean q_0 or r_0 as the situation dictates (thus, the outcome in footnote 1 would be acceptable).

Working Example: Consider the scenario where Alice is forced to withdraw money from an ATM by Oscar. In addition to wanting to signal a silent alarm (which we assume Oscar is not trying to prevent, as he believes he can escape on time), the ATM issues marked bills in place of real money. Oscar has a method for distinguishing marked bills but cannot do it on the spot. Therefore, his goal is only to escape with unmarked money even if he is not initially sure which bills are which.

4.2.1 Solution

The 2P system is inadequate in this scenario as Oscar can conduct an attack using the iteration principle (assumption 3). He will first force Alice to type in her PIN and withdraw money, and then knowing the 2P system is in place (assumption 1), he will force her to authenticate a second time while observing that she enters a different PIN (assumption 2). Alice will have no choice but to enter p_0 on one of the two occasions and Oscar will achieve his goal.

To address this threat, we can exploit the fact that Oscar is non-persistent and must end his coercion at some point, t_1, to escape. Specifically if Alice's account is temporarily locked until some time $t_2 > t_1$, Oscar will not achieve his goal. We must, however, exercise caution in how locking of an account is triggered. A naïve approach may be to lock the account after receiving p^*, however Oscar could use this information to screen p_0 from p^*. He could demand that Alice enters a password that does not produce a locked account by threatening retaliation against her if her password does, thereby accomplishing his goal of getting unmarked money. For this reason, the event that locks the account must be invariant to the type of the passwords being entered.

Scheme 2. 2P-lock. Alice knows two passwords $\{p_0, p^*\}$. For normal authentication purposes, Alice uses p_0. To communicate panic, she uses p^* which causes r^* or q^* as the scenario dictates (but does not not lock the account). If Alice authenticates multiple times within a window of time, t_1, using the same password each time, her account will not be locked. However upon using two different passwords within t_1, the account will be made inaccessible for a period of t_2 such that $t_2 > t_1$.

The 2P-lock scheme is designed specifically to thwart an iteration attack. Once Alice has entered one password, forcing her to enter the other password is futile as it will only lock the account. Furthermore, the locking will occur regardless of whether Alice initially entered p_0 or p^*—making it impossible for Oscar to determine if he achieved his goal by the behaviour of the scheme. Oscar can conduct a forced-randomization attack, giving him probability 1/2 of receiving unmarked money, however the scheme could include more than two panic passwords which would lower this probability. Also, Oscar could coerce Alice when he knows she has had recent (less than t_1 ago) communications with Bob. Since Oscar knows Alice has recently entered p_0, he knows that any password that locks the account is a panic password, giving him the same screening and coercion abilities that he had under 2P. However if Oscar could observe when Alice communicates with Bob, he could more simply hijack her account after she has authenticated and before she logs out.

4.3 Persistent Attacks

Consider a scenario with panic communication $\{p^*, q^*, r_0\}$ and a persistent adversary with goal $G_O[\exists q \in T : q_0]$. In addition, preventing both signals and screens is important.

Working Example: Consider the previously defined improper influence problem in Internet-based voting. Alice could be coerced by Oscar into voting a certain way. Alternatively, Alice may want to sell her vote to Oscar by casting her ballot in his presence. Bob's observable response will be the report of a successful casting of Alice's vote, but he will take the unobserved reaction of disregarding any votes cast under a panic password.

4.3.1 Solution

In this case, we assume Oscar is persistent, meaning he can observe Alice for extended periods of time. In the working example, he could be her employer, an intrusive partner, or a union representative. For this reason, locking the account until t_2 is not guaranteed to be effective. It is also not prudent as it would easily allow an adversary to prevent Alice from voting by coercing her into repeatedly locking herself out of her account—an effective denial of service attack. Therefore 2P-lock is not suitable. The nature of the system does not preclude the iteration attack either, rendering 2P ineffective as well.

The problem with a persistent attacker is that he could eventually exhaust Alice's memory of panic passwords, forcing her to enter p_0 eventually. Therefore a suitable scheme for this model would equip Alice with an arbitrarily large number of panic passwords, so that she could produce a virtually endless list of them.

Scheme 3. P-Compliment Alice knows one password, p_0, which she uses for normal authentication purposes. To communicate panic, she uses any other password: $\{\forall p \neq p_0 \in P : p^*\}$. There are no invalid passwords in this scheme.

This system allows Alice to enter any password other than her real password to communicate a panic state. The drawback of P-Compliment is that if Alice mistakenly enters the wrong password, by definition of the problem, she cannot distinguish the panic state from the normal one. From a usability perspective this is undesirable. In the working example, given that there is no consequence to entering a panic state, Alice could vote more than one time using p_0 to ensure that she entered the password correctly at least once. However this is a specific example and we would like a solution for the entire class.

If we generalize P-Compliment, Alice knows two things: one real password p_0 and a rule, '*anything except p_0 is a panic password.*' The fundamental problem is that the rule is too inclusive. We can generalize to four desirable properties,

1. an arbitrarily large number of panic passwords,
2. an arbitrarily large set of invalid passwords,
3. a small distance between panic and invalid passwords,
4. a rule that is cognitively easy to apply.

The reason for the first property is twofold: we wish to prevent an iteration attack without locking the account, and as a password scheme, it must have all the security properties of any password system. As such, it must not be susceptible to exhaustive search attacks. The second property is to promote the probability that a mistyped p_0 is an invalid password instead of a panic password. This property is necessary but not sufficient—we also need to assure that panic and invalid password spaces are well mixed. For example, a good metric for passwords is the Damerau-Levenshtein distance [3], which is the minimum number of a specific set of operations to convert one string into another, where the operations are insertion, deletion, or substitution of a single character and transposition of two characters. The metric is designed for application to spell-checking. Finally, from a usability perspective, users need to apply the rule confidently, especially since they are operating under duress.

As a general approach, we start with a password space P and we apply the separating function $f()$ to each element in it such that any $p \in P$ is mapped to V or I (*i.e.,* valid or invalid). $|V|$ and $|I|$ should both be sufficiently large. p_0 is selected from V and a panic password is $\{p_v^* \in V \| p_v^* \neq p_0\}$. All elements of I are invalid.

One implementation of this general scheme is used by some US governmental agencies [4]. A password is combined with a PIN to create p_0. Panic passwords are any combination such that the password portion is correct and the PIN is incorrect. Compositions of a wrong password are considered invalid. This scheme is problematic if a mistake is made on the PIN portion. Also, under coercion, Alice is forced to reveal the password part of p_0 in order to use a p^* which reduces the security of p_0 against an exhaustive search significantly. Since the threat model these passwords are used under is unknown by the authors, we are not suggesting this scheme is used improperly—only that it has drawbacks for scenarios of this type.

Scheme 4. `5-Dictionary` Let P be any five strings and let V be any five strings in a standard dictionary. Alice composes a password by selecting a set of five dictionary words. Any other set of five dictionary words is a panic password. Any other set of either dictionary or non-dictionary strings is considered invalid.

This systems meets the five criteria listed above. The standard Unix dictionary is just over 25,000 words, which provides up to 73 bits of security for a 5 word combination. The invalid password space is arbitrarily large, as any word can be any other combination of alphanumerical characters. Although empirical data of common typos, along with numerical analysis would need to confirm the average Damerau-Levenshtein distance between typos and words, our intuition is that there is a considerable probability that many typos would be caught—certainly higher than in `P-Compliment`. A similar approach could be taken with a click-based graphical password scheme.

Scheme 5. **5-Click** Alice is presented with a sequence of five images, and a rule for what regions of the image are in V based on the content of the picture. She knows one password $\{p_0\}$ which is a sequence of one click per image which she uses for normal authentication purposes. To communicate panic, she clicks on another valid region for at least one of her clicks.

For example, in the PassPoints scheme [13], users authenticate by clicking on certain pixels (within a tolerance) in an image. A separating function could be used to predefine certain regions of an image for V and give them semantic meaning to help users remember them. For example, V could be faces on the cover image of *Sgt. Pepper* or cars in an image of a parking lot. Using a sequence of different images will increase entropy, and is similar to the Cued Click-Points (CCP) scheme [7], where users authenticate by clicking one point in each of a sequence of five images. However in CCP, the i^{th} image displayed to the user is dependent on where the user clicked in the (i-1)^{th} image. This last property has significant usability improvements, as it provides feedback to the user in case of an error, but in this model it could also be used by an adversary to screen—the user would know the sequence of images for p_0 and but not p^* (unless she explicitly committed a panic sequence to memory). Instead, using the same sequence of images would be a suitable compromise between increased entropy and usability.

4.4 Screening Observable Responses

For a final scenario, consider a scenario with $\{p^*, q_0, r^*\}$ in a screening model where persistent Oscar's goal is $G_{\text{O}}[\exists r \in T : r_0]$. In particular, our focus is on an inherent problem between r^* and screening. Our approach to this scenario applies equivalently to $\{p^*, q^*, r^*\}$ and $G_{\text{O}}[\exists q \in T : q_0]$.

Working Example: Consider the scenario where Alice has network access to sensitive information, such as credit card account numbers, which are password protected. Under coercion, she wants the server to return false information that looks real or be routed through to a "honeypot" server [11].

4.4.1 Solution

Consider using `2P`. Given that $r^* \neq r_0$, Oscar can apply the iteration principle to r instead of p. He can request that Alice authenticates to the server using a different password such that it produces a different r. Once he observes two different r values, he knows one must be r_0. The solution to this problem parallels the dictionary solution—the set of r must be arbitrarily large. For this reason, it is not always possible to solve this problem. The password-side of scheme can be handled by `5-Dictionary` but the response-side would have to contain randomization of data.

It may appear possible to compromise by having a finite but secrete number of panic responses, such that Alice could claim that all possible responses have been iterated through when at least the real one still remains unseen by Oscar. There are two problems with this approach. First, it is a slight breach of assumption 1 depending on how you define what is part of the system and what is the shared secret. Second, if the panic state draws randomly from a finite set of possibilities, these states will eventually repeat while the legitimate data will not repeat.[2] For this reason, Oscar can screen out sets of data as illegitimate once they appear more

[2]We assume a consistent username is used by Alice.

than once and use this property to force Alice to authenticate with p_0.

We leave a generalized solution to scenarios of this nature for future work. Presenting randomized responses that appear legitimate depends on the what "appear legitimate" and are thus situational. It appears that solutions would have to be tailored for the specific circumstances.

5. DISCUSSION

5.1 Issuing New Panic Passwords

In the case of panic passwords, the ability to be issued a password or reset a password on the basis of supplemental authentication information provides a backdoor that an adversary can successfully exploit. Instead of forcing Alice to authenticate, Oscar would simply force her to reveal the information necessary to reset the password or be issued a new one. Without completely solving the problem, passwords could be issued but not reset, effectively limiting Oscar's window of opportunity—this may feasible in less critical scenarios. For more serious scenarios, there appears to be no alternative to having the user create and reset passwords in a controlled, coercion-free environment. This could be by visiting a physical bank to set an ATM PIN or online banking account, or a one-time in-person registration to enable online voting, which is likely an improvement over voting in person in every election on a specific day. In some countries, government services are provided under federated access and so the registration could be used in scenarios beyond voting.

5.2 Reconsidering Trustworthy Bob

When introducing the three participants in the threat model—Alice, Bob, and Oscar—we assumed Bob, the entity being authenticated to by Alice, was trustworthy. In certain scenarios, this is not an adequate assumption. For example, we considered the example of Internet-based voting where Bob's hidden reaction was to dispose of votes cast under a panic password and count only the votes cast under a normal password. Much voting technology research is aimed at eliminating the trust of any entity during the voting process, including the server. End-to-end (E2E) voter verifiability means that voters can verify that their cast ballots are included in the final tally unmodified. To take one example, the Scantegrity II voting system [6] adds E2E integrity to optical-scan election systems.[3] Like most E2E systems, Scantegrity II issues a receipt to a voter that allows her to ensure that her vote was cast-as-intended without revealing how she voted. It is not immediately apparent how receipt-based voting could be made compatible with panic password schemes. If Alice does not trust Bob, she wants proof (*i.e.,* through a receipt and audit) that her real ballot was cast and her panic ballots were discarded. However such proof could also be used by Oscar to screen the ballots he forced her to cast. We leave this problem open to future work.

6. CONCLUDING REMARKS

Despite a large volume of research on the use of passwords, examined from many diverse angles—security, usability, agreement schemes, storage, dual factor, biometrics, to name a few—academic study of panic passwords is virtually non-existent. While they appear in some commercial and military applications, our hope is that this paper provides a basis for further attention from the research community. We believe that better schemes are possible and more involved threat models could be devised. In particular, attention from a usability perspective would be extremely valuable, including user studies and case studies.

[3]In optical-scan systems, voters mark a paper ballot by penciling in an oval and this ballot is then scanned for electronic tallying.

7. REFERENCES

[1] FIPR response to the Home Office: "Consultation on the draft code of practice for the investigation of protected electronic information – Part III of the Regulation of Investigatory Powers Act 2000". http://www.fipr.org/060901encryption.pdf, September 2006.

[2] R. J. Anderson, R. M. Needham, and A. Shamir. The steganographic file system. In *Proceedings of the Second International Workshop on Information Hiding*, pages 73–82, 1998.

[3] G. V. Bard. Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric. In *Fifth Australasian Information Security Workshop (AISW 2007)*, pages 117–124, 2007.

[4] R. T. Carback. Private conversations, April 2008.

[5] D. Chaum. Achieving electronic privacy. *Scientific American*, pages 96–101, August 1992.

[6] D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. A. Ryan, E. Shen, and A. T. Sherman. Scantegrity II: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *EVT'08: Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop*, 2008.

[7] S. Chiasson, P. C. van Oorschot, and R. Biddle. Graphical password authentication using cued click points. In *ESORICS*, volume 4734 of *Lecture Notes in Computer Science*, pages 359–374. Springer, 2007.

[8] I. Leemon C. Baird, M. E. Harmon, R. R. Young, and J. James E. Armstrong. Apparatus and method for authenticating access to a network resource. United States Patent, 6732278, 2004.

[9] R. J. Massa, T. R. Ellis, and R. G. LePage. Intelligent surveillance alarm system and method. United States Patent, 4589081, 1986.

[10] A. Munje and T. Plestid. Password methods and systems for use on a mobile device. United States Patent (Pending), 11181522, 2007.

[11] N. Provos. A virtual honeypot framework. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 1–14, 2004.

[12] R. K. Russikoff. Computerized password verification system and method for atm transactions. United States Patent, 6871288, 2005.

[13] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon. Passpoints: design and longitudinal evaluation of a graphical password system. *Int. J. Hum.-Comput. Stud.*, 63(1-2):102–127, 2005.