Quantifying the Security of Preference-based Authentication

Markus Jakobsson*
Palo Alto Research Center
Palo Alto, CA 94304
markus.jakobsson@parc.com

Liu Yang, Susanne Wetzel
Stevens Institute of Technology
Hoboken, NJ 07030
{swetzel,lyang}@cs.stevens.edu

Abstract

We describe a technique aimed at addressing a long-standing problem for password reset: security and cost. In our approach, users are authenticated using their preferences. Experiments and simulations have shown that the proposed approach is secure, fast and easy to use. In particular, the average time for a user to complete the setup is approximately two minutes, and the authentication process takes only half that time. The false negative rate of the system is essentially 0% for our selected parameter choice. For an adversary who knows the frequency distributions of answers to the questions used, the false positive rate of the system is estimated at less than half a percent, while the false positive rate is close to 0% for an adversary without this information. Both of these estimates have a significance level of 5%.

1 Introduction

One of the most commonly neglected security vulnerabilities associated with typical online service providers lies in the password reset process. By being based on a small number of questions whose answers often can be derived using data-mining techniques, or even guessed, many sites are open to attack. A good overview of this problem was recently provided by Rabkin [15]. To exacerbate the problem, many sites pose the very same questions to users wishing to reset their forgotten passwords, creating a common "meta password" between sites: the password reset questions. At the same time, as the number of accounts per user increases, so does the risk for the user to forget her password. Unfortunately, the cost of a customer-service mediated password reset—currently averaging $22 [14]—is far beyond possible for most service providers.

In a recent paper by Jakobsson, Stolterman, Wetzel and Yang [9], a promising alternative was introduced. Therein, a system based on user preferences was proposed in order to reduce the vulnerability to data-mining. The viability of such an approach is supported by findings in psychology [2, 16], showing that personal preferences are often more long-lived than long-term memory. However, in spite of the desirable properties of the approach by Jakobsson et al., their practical implementation remained impractical: To obtain a sufficient security against fraudulent access attempts—which for many commercial application is set below 1% false positive and very close to zero false negative negative—a very large number of preference-based questions was needed. More specifically, to achieve these error rates, a user would have to respond to some 96 questions, which is far too many in the minds of most users.

In this paper, we show that a simple redesign of how questions are selected can bring down the number of questions needed quite drastically. Motivated by the observation that most people do not feel strongly (whether positively or negatively) about all but a small number of topics, we alter the setup interface from classification of preferences (as in [9]) to a *selection of topics for which the user has a reasonably strong opinion.* An example interface is shown in Section 3.

The main focus of this paper is a careful description of the proposed system, a description of the expected adversarial behavior, and a security analysis to back our claim that the desired error rates are attainable with only sixteen questions. The analysis is carried out by a combination of user experiments and simulations. The user experiments, many of which were already performed by Jakobsson et al., establish answer distributions for a large and rather typical user population. The simulations then mimic the behavior of an adversary with access to the general answer distributions (but with no knowledge of the preferences of the targeted individuals). Further, and in order to provide a small error margin of the estimates of false positive rates, a large number of

*Much of this work was performed for RavenWhite Inc., and while the author was with Indiana University.

user profiles are emulated from the initial distributions. These are then also exposed to the simulated adversary. The false negative rates are estimated using user experiments in which users indicate their preferences, and then, attempt to provide the correct answers to the corresponding questions. This second part of the experiment was performed at least 24 hours after the first part, to avoid interference with short-term memory. (We do not have to worry so much about long-term memory, since, after all, the user is not asked to remember anything.)

While only extensive use of the technology can assert the estimated error rates we have identified, it is indisputable that the use of the proposed technique will have one immediate security benefit: Unlike currently used methods, our proposed method significantly reduces the vulnerability to attacks in which fraudsters set up sites that ask users to provide the answers to security questions in order to register—and later turn around and use these very answers to gain access to other accounts for these users. The reason for this lies not only in the much larger pool of questions that users can select from, but also in a randomization technique that makes it impossible to anticipate what questions a user selected—or was even allowed to make selections from.

It is worth mentioning that if a server were to be compromised, and user preference data leaked—or if a user is afraid that his preferences may have been learned by an attacker for some other reason—then it is possible for him to set up a new profile. Simply put, there are enough items to be selected from even if a first profile would be thrown away. As more questions are developed onwards, this protection will be strengthened further. This puts our password reset questions on par with passwords in the sense that a user may change it over time, and still be able to authenticate. This is not quite the case for traditional password reset questions due to the very limited number of available questions. For the same reason, it is possible to deploy our proposed scheme at multiple sites, without having to trust that one of these does not impersonate the user to another.

We believe our approach may have profound benefits on both Internet security and on the costs of managing password reset. However, as with any technology in its infancy, we are certain that there are further enhancements that can be made—whether to lower the error rates or to introduce security features that have not even been identified to date.

Outline. We begin by reviewing related work (Section 2), after which we provide an overview of the system (Section 3). We then detail the adversarial model (Section 4). In Section 5, we quantify the security of our proposed technique, first by describing experimental results (Section 5.1), after which we detail simulation results (Section 5.2) and explain the accuracy of our estimates (Section 5.3).

2 Related Work

Security questions are widely used by online business for fallback authentication. It is believed that banks are well motivated to secure the accounts of their customers. Therefore, banks represent the industrial state-of-the-art in security-question-based authentication. A recent survey conducted by Rabkin [15] supports the common belief that many security questions suffer from weaknesses related with either usability or security, and often both.

An early empirical study on security questions was conducted by Haga and Zviran [7], who asked users to answer a set of personal security questions and then measured the success-rate of answers from users, users' friends, family members, and significant others. Many of the questions studied in [7] are still used by online banks today. Recently, research has shown that many of those questions are vulnerable to guessing or data-mining attacks [11, 6] because of the low entropy or public availability of their answers.

Improving password reset is a problem that is beginning to receive serious attention from researchers. A framework for designing challenge-question systems was described by Just in [12]. This paper provides good insights on the classification of different question and answer types, and discusses how they should meet the requirements for privacy, applicability, memorability, and repeatability. The paper points out that for recovery purposes, it is desirable to rely on information the user already knows, rather than requiring him or her to memorize further information. It is important to note that the preference-based authentication technique has this property.

Security questions are also used by help desks to identify users. A method called query-directed passwords (QDP) was proposed by O'Gorman, Bagga, and Bentley [13]. The authors specified requirements for questions and answers and described how QDP can be combined with other techniques like PINs, address of physical devices, and client storage in order to achieve higher security. Unfortunately, QDP was mainly designed for call centers to identify customers. Thus, QDP will have approximately the

same high running cost [14] as other password reset approaches involving help desk service.

Aside from being used for password reset, personal questions have been used to protect secrets. Ellison, Hall, Milbert, and Schneier proposed a method named *personal entropy* to encrypt secrets or passwords via the answers to a number of questions by users [4]. Their approach was based on Shamir's secret sharing scheme, where a secret is distributed into the answers of n questions, and at least t of them need to be correctly answered in order to reconstruct the secret. Frykholm and Juels proposed an approach called *error-tolerant password recovery (ETPAR)* to derive a strong password from a sequence of answers to personal-knowledge questions [5]. ETPAR achieves fault tolerance by using error-correcting codes in a scheme called *fuzzy commitment* [10]. Our approach has the property of error-tolerance, but we achieve that in a different way and with much greater flexibility in terms of the policy for what constitutes a successful attempt. Also, ETPAR requires significant key-lengths, as offline attacks can be mounted in that system. In contrast to ETPAR, we do not protect the profile information of users against the server; it may be possible to extend our work in that direction, but it is not within the scope of this paper.

Asgharpour and Jakobsson proposed the notion of *Adaptive Challenge Questions* [1], which does not depend on preset answers by users. It authenticates users by asking about their browsing history in a recent period, which the server mines using browser recon techniques [8]. While this may be a helpful approach, it is vulnerable to attackers performing the same type of browser mining, which suggests that it should only be used as an add-on authentication mechanism to increase the accuracy of another, principal method.

Our work is based on the work of Jakobsson, Stolterman, Wetzel, and Yang [9] who proposed a password reset approach nameded *preference-based authentication*. The basis for their approach is that preferences are stable over time [2, 16], and they are less likely to be publicly recorded as fact-based security questions, e.g., high school name, mother's maiden name, etc. [12]. Preference-based authentication provides a promising direction to authenticate users who have forgotten their passwords. However, in order to obtain sufficient security against fraudulent access, a user has to give his opinions to a large number of questions when registering an account, making the system impractical. In this paper, we show a redesign of the interface can drastically reduce the number of required questions for authentication without losing security. However, our contribution goes beyond proposing a better user interface; other important contributions of our paper relate to the techniques we developed in order to assess the resulting security. This involves user experiments, user emulations, and simulations of the attacker, and an optimization of parameters given the obtained estimates.

3 Overview of the System

In [9], Jakobsson et al. propose to authenticate users by their personal preferences instead of using knowledge associated with their personal information. In their approach, a user has to answer 96 questions during the setup phase in order to obtain sufficient security against fraudulent access. Our experiments suggest that very few users are willing to answer more than perhaps 20 questions for authentication, and a system asking too many questions for authentication is not usable in practice. An open question posed in [9] was whether preference-based questions can be used to design a truly practical and secure system. This paper answers that question in the affirmative: We show that a simple redesign of the setup interface can reduce the number of required questions quite dramatically.

Our design is motivated by an insight obtained from conversations with subjects involved in experiments to assess the security of the system: Most of them indicated that they only have reasonably strong opinions (whether like or dislike) about a small portion of the available topics. Thus, instead of classifying each available topic according to a 3-point Likert scale (like, no opinion, dislike), the new interface lets users select topics that they either like or dislike. The majority of topics are not selected, which requires no user action. The authentication interface is designed to only require classification of preferences (like or dislike) for the selected topics, displayed to the user in a random order.

Setup. During the setup phase, a user is asked to select L items he likes (e.g., *Playing baseball, Karaoke, Gardening, etc.*) and D items he dislikes (e.g., *The opera, Jazz music, Reality shows, etc.*) from several categories of topics. For each user, only a subset of items are presented for selection. The subset is chosen in a random way from a candidate item set, and the order of the items in each category is randomized. Our experiments tested a range of different parameter choices; these guided us to select $L = D = 8$. The output from

the setup phase is a collection of preferences—these are stored by the authentication server, along with the user name of the person performing the setup. An example of the setup interface is shown in Figure 1. This is the version of the system available at `www.blue-moon-authentication.com`.

Authentication. During the authentication phase the user presents his username, and the server looks up the previously recorded preferences. These items are then randomly ordered, and turned into questions to which the user has to select one out of two possible opinions: like or dislike. The correctness of the answers is scored using an approach described in [9], viz to assign some positive points to each correctly answered question and some negative points to each incorrectly answered question; the exact weighting of these also depends on the entropy of the distribution of answers to this question among the population considered. The authentication succeeds if the score is above a preset threshold, denoted by T.

Returning to the differences in user interfaces, we see that the user interface we propose represents a simplification over the interface proposed in [9], where the entire classification of each topic is performed for both setup and authentication. In our version, a user only has to classify $L + D = 16$ topics during authentication. (It may be possible to further reduce this number by selecting topics with a higher entropy, and, of course, if a lower degree of assurance is required than what we set out to obtain.)

As mentioned before, our system has the security benefit that it is not possible for a "pirate site" to ask a user the same questions as the user answered at another site in order to learn his answers and later impersonate him. (Of course, this only holds as long as the pirate site does not connect to the other site, posing as the user, claiming to have forgotten his password.) This is a benefit that is derived from the user interfaces we proposed, and was not a security feature offered by the original system. Therefore, our system is not vulnerable to this *question-cloning attack*, in contrast to the system in [9]. The reason is simple: since our proposed protocol requires honest sites to randomize the order of topics that the user can select from, and almost any user can find multiple selections that represent his preferences, it is unlikely for an adversarial site to capture the same selections as another site would. The security of this feature will be further enhanced if a larger number of selectable topics are presented to the user, but that is beyond the scope of the current paper.

Computation of Scores. The method to compute the score follows the methodology in [9]. The score of an authentication attempt measures the correctness of the answers. It is defined as a ratio between two values: S_A/S_S, where S_A denotes the accumulated points earned during the authentication phase and S_S denotes the total points of items selected during the setup phase. The points associated with an item are based on its importance, where an item is considered important if it is hard to guess the opinion given by a user. The importance of an item is measured by its information entropy, which is computed according to the definition in [17]. During the authentication, a user receives the points associated with an item if he correctly recalls the original opinion. If he makes a mistake, he is punished (by receiving negative points). The punishment for a mistake equals the points associated with this item, multiplied by a parameter c that controls the balance between the benefit of providing a correct answer and the punishment for providing an incorrect one. (If a legitimate user would always answer all questions correctly during authentication, this would be set at negative infinity; however, since we must allow users to make a small number of mistakes, this is not the parameter choice we make.)

4 Adversarial Model

We study the security of the scheme by investigating how likely an attacker can successfully impersonate a targeted user. For each targeted user, the attacker is only allowed to have one try. (Obviously, this is a matter of policy, but simplifies the analysis.) An attack is considered to *succeed* if the resulting score is above a preset threshold T. The attacker is assumed to know the user name and have access to the Internet connection through a personal computer. A two-tiered adversarial model is considered, which includes two types of attacks, named *naive* and *strategic* attacks.

Naive Attack. In this type of attack, the adversary is assumed to have the information that users are asked to select L items they like and D items they dislike during the setup phase but know nothing of the relative selection frequencies of the available topics. To impersonate a user, the adversary randomly selects *like* opinion for L items and *dislike* opinion for D items during an authentication attempt. This is a realistic assumption for most real-life adversaries, as supported by the fact that most phishing attacks do not use advanced

Figure 1: An example of the setup interface, where a user is asked to select 8 items he likes and 8 items he dislikes.

javascript techniques to cloak the URLs or use targeting of attacks—it is easier to spam a larger number of people than to attempt to increase yields by better background research.

Strategic Attack. In this type of attack, in addition to knowing L and D, an adversary knows the distributions of the opinions associated with the items used by the system. In particular, for each item used in the authentication, the adversary knows the percentages of users who chose *like* and *dislike* respectively. We call these percentages *like rate* and *dislike rate*, denoted by p and q. The adversary selects a set of opinions which maximize his likelihood of success by using the following strategy. For the presented items, the adversary selects *like* opinion for L items, and *dislike* opinion for D items such that $p_{i_1} \times \ldots \times p_{i_L} \times q_{j_1} \times \ldots \times q_{j_D}$ is maximized, where $(i_1, \cdots, i_L, j_1, \cdots, j_D)$ is a permutation of indices $(1, 2, \cdots, L + D)$ for the $L + D$ items.

The best strategy of either of our adversaries is different from the adversary described in [9] as follows: The adversary in [9] does not know the total number of strong opinions chosen by a user, while an adversary in our method knows the number of opinions selected by a user. Because the number of strong opinions selected by a user is unknown in [9], the best strategy for an adversary in [9] is to answer each question by selecting an opinion that the most users had. In contrast, in our model, L and D are known, and the method for the adversaries to get the highest likelihood of success is to select $L + D$ opinions such that the product of the corresponding like rates and dislike rates is maximized. (While this may appear as a minor difference, it had a large impact on the efficiency of our simulations, which were computationally quite demanding.)

Remark: Our work does *not* consider correlations between preferences, in spite of this being a natural fact of life. While the items from which to select preferences were chosen in a way that would avoid many obvious correlations, it is clear that a more advanced adversary with knowledge of correlations would have an advantage that the adversaries we consider do not have. The treatment of correlations is therefore of large practical importance but is beyond the scope of this paper.

5 Quantifying the Security

The security features of our approach have been evaluated in three ways: using experiments, user emulations, and attacker simulations. The goals of the experiments were to obtain user data to be used to assess error rates. Due to a relative shortage of suitable subjects, we augmented the experimental data with emulated user data derived from distributions obtained from Jakobsson et al. [9]. The simulation model we developed provides a way to evaluate the security of the system and to find suitable parameters to minimize and balance the error rates. This is done by simulating the two types of adversaries we consider for each profile—whether obtained from the experiment or the emulation. In addition, the simulation provides measures for the accuracy of our estimates. (The accuracy part is what made the need for emulated users evident, as a total of 49000 user profiles was needed to get the desired accuracy of our simulations.)

From the description of the experiments and simulations, it is possible not only to understand why our proposed system is secure, but it is also possible to follow how our experiments shaped our system over time. More specifically, while our final

system uses a total of 16 questions, many of the early experiments used only 12 or fewer. When these experiments pointed to the need for additional questions, we changed the parameters and extrapolated from the findings involving only 12 or fewer questions. (We will explain why this extrapolation is reasonable to make after describing the experiments.) Similarly, whereas the proposed system requires users to identify the same number of likes as dislikes, our experiments do not consider only this parameter choice—however, our exposition focuses on this case, since that parameter choice resulted in the best error rates. Consequently, the following subsections will at times use slightly different parameter choices than we ended up with. To avoid introducing confusion due to this, we will occasionally remind the reader of the difference between the experimental observations and the final conclusions. Most prominent among these will be the final error rates that we computed.

5.1 Experimental Evaluation

We conducted an experiment involving 37 human participants. Unlike our final system shown in Figure 1 which asks users to select 8 items they like and 8 items they dislike, users in this experiment were asked to select 5 items they like and 5 items they dislike during the setup phase. For each participant, there was at least a 24 hour time period between the setup and authentication phase. Each user was allowed to perform one authentication attempt. All participants completed the setup and authentication phases. Our tests showed that a user takes approximately two minutes on average to complete the setup, and half of that to complete the authentication phase. This is much shorter than the time reported in [9].

As already explained in Section 3, an authentication attempt succeeds if the resulting score is above a specific threshold T. For a specific T, the false negative rate (denoted by f_n) of the system is defined as the ratio of the number of users' unsuccessful authentication attempts (i.e., attempts resulting in a score lower than T) over the total number of authentication attempts. The false positive rate (denoted by f_p) corresponds to the success rate of an attacker. An attack is considered successful if the respective authentication results in a score above the threshold T. For each user setup profile the adversary is allowed to try an attack only once. In our experiment and simulation the false positive rate is then determined as a ratio between the number of successful attempts and the number of setup profiles being attacked. As described in Section 3, the parameter c is used to adjust the quantity of punishment for incorrect opinions. From the point of view of system design, choosing a high value of c can severely punish incorrect opinions during an authentication attempt, which is beneficial for keeping an adversary from succeeding. This is due to the fact that there is a much higher likelihood for an adversary to provide one or more incorrect opinions for questions than a legitimate user does. However, a high value of c also increases the likelihood that a legitimate user who accidentally gave one or more incorrect answers fails to authenticate. Thus, it is important to find a suitable value for c such that both f_n and f_p are as small as possible yet well-balanced. To reach this goal, we have investigated the effects of c and T on f_n and f_p by considering f_n and f_p as functions of c and T. Based on experimental data we have determined suitable values for c and T by performing a two-dimensional search in the space (c, T), where we let c range from 0 to 30, and T range from 0 to 100% (taking steps of size 1 for c and 1% for T).

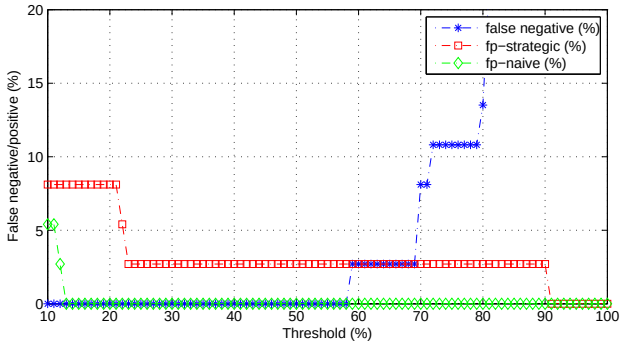

Figure 2: The false positive and false negative rates as a function of the threshold T, when users were asked to select 5 items they like and 5 items they dislike during the setup phase.

Figure 2 shows the variation of false negative and false positive rates with respect to the value of threshold T when users were asked to select 5 items they like and 5 items they dislike. The false positive rates were computed for both types of attacks—naive and strategic—for the 37 user setup profiles. The naive adversary selects opinions in a random way, while the strategic adversary maximizes its likelihood of success based on its knowledge of frequency distribution of opinions associated with items. The suitable value we determined through the search is $c = 6$. For $T = 58\%$, we see that the false negative rate is 0%, the false positive rate for the strategic attack is 2.7%, and the false positive rate for the

naive attack is 0%. This finding led us to consider increasing L and D in order to obtain lower false positive rates. As we will see in Figure 3, the false positive rate for the strategic attack decreased to $1.623 \pm 0.125\%$ when users were required to select 6 items they like and 6 items they dislike. This is for twelve questions only; when extrapolating to sixteen questions, we obtain an approximate false positive rate of less than half a percent against the strategic adversary, and the false positive rate is close to zero for the naive adversary.

5.2 Simulation-based Evaluation

Our simulation method works in two steps. The first step is to emulate how a user selects items he likes and dislikes during the setup phase by using statistical techniques. We denote this process by *EmuSetup*. Executing *EmuSetup* once will generate a *setup profile* for a hypothetical user, where the setup profile contains L items liked, and D items disliked by the hypothetical user. The Mann-Whitney test The setup profiles generated by *EmuSetup* are believed to have the same distribution as the setup profiles of real users in real experiments. This will be explained further in the following subsections of this paper. By repeatedly executing *EmuSetup*, we generated a large number of hypothetical user setup profiles. The second step of the simulation is to apply the two types of attacks—naive and strategic—to the hypothetical setup profiles and determine the success rates of these attacks, which correspond to the false positive rates of the system. The details of designing and carrying out the simulation are described in the remaining part of this section.

5.2.1 Intuitive Approach of Emulation

The *EmuSetup* function emulates how users perform the setup using the interface described in Section 3. The emulation is based on making use of the distribution of preferences determined as part of an experiment described in [9]. In *EmuSetup*, a setup profile is generated by presenting several lists of items to a hypothetical user who then selects items according to the real probability distributions. For example, if the hypothetical user is asked to select an item that he likes from a list containing twelve possible items, then the selection is made according to the like rates (see Section 4) of the items obtained from real users in [9]. A toy example is as follows: Consider the three items *Vegetarian food, Rap music* and *Watching bowling*. Assume that the frequencies with which people responded *like* for these three items were 0.3, 0.2, 0.1, then the overall sum of these frequencies is 0.6. If a hypothetical user has to select one item he likes from the three, then he would select *Vegetarian food* with a probability of $0.3/0.6 = 50\%$, select *Rap music* with a a probability of $0.2/0.6 = 33.3\%$, and select *Watching bowling* with a probability of $0.1/0.6 = 16.7\%$. By using this approach, a hypothetical user selects L items he likes and D items he dislikes. In our simulation, a large number of hypothetical users were emulated as above. (While they fail to take correlations into consideration, that is not a limitation in the context of the adversaries we consider.)

5.2.2 Mathematical Description

Now we give the mathematical description of how an emulated user selects preferences from a list of items. Suppose the list contains m items and the associated like rates are $p_1, p_2, \cdots, p_m$, and the corresponding dislike rates $q_1, q_2, \cdots, q_m$. The likes rates and dislike rates for all items were obtained from an experiment involving 423 participants in [9]. Assume the selections of items are independent. This is reasonable when the size of the candidate set is large. Then a hypothetical user will select to like the ith item in the list with a probability of

$$P_i = Pr\{X = i\} = \frac{p_i}{\sum_{j=1}^{m} p_j} \tag{1}$$

where X denotes the index of an item in this list, and analogously for the dislikes.

The idea of Equation (1) is implemented using the following approach: To decide which item to select, pick a random value between 0 and 1 from a uniform distribution and see which interval it falls into

$$I_i = [S_{i-1}, S_i) \tag{2}$$

for $i = 1, \cdots, m$, where $S_i = \sum_{j=0}^{i} P_j$, and $P_0 = 0$. If the random value falls into I_i, then the ith item is selected. The method for a hypothetical user to select one item he dislikes is similar to the process described above, except that the dislike rates of the items are used to make the decision.

For $L = D = 5$, we performed Mann-Whitney tests on the setup profiles generated by real users in Section 5.1 and the setup profiles generated by *EmuSetup*. The results show that they are not significantly different with a significance level of 0.05. Thus, this provides strong evidence the setup profiles generated by *EmuSetup* have the same distribution as those provided by real users for the same choices of L and D.

5.2.3 Computation of False Positive Rates

The setup profiles generated by *EmuSetup* are used to evaluate the security of our approach by estimating the false positive rates for certain choices of L and D. According to the Central Limit Theorem in statistics, the larger the sample size is, the closer the sample mean is to the theoretical expectation of a random variable. Based on this insight, we generated more than enough setup profiles for hypothetical users in order to obtain high accuracy in our evaluation. The number of setup profiles we generated was 49000. How this number was determined is discussed in Section 5.3. In our emulation, each of the 49000 hypothetical users picks 6 items he likes and 6 items he dislikes as his setup. Then we applied the naive and strategic attacks to the generated setup profiles and computed the success rates of these attacks. The success rates of these attacks correspond to the false positive rates of the system. Figure 3 shows the relationship between the obtained false positive rates and the value of threshold T when $c = 6$ (determined in Section 5.1). For any threshold value between 23% and 58%, the false positive rate for the strategic attack is $1.623 \pm 0.125\%$. For the naive attack, the false positive rate is $0.137 \pm 0.033\%$. The significance level of our estimates is 5%.

By comparing the false positive rates in Figure 2 and Figure 3, we observed two interesting phenomena:

- The larger L and D are, the smaller false positive rates we get, i.e., the more secure the system is.
- When $L = D$, the f_p corresponding to the strategic attack is close to $\frac{1}{2^L}$. For example, in Figure 2 where $L = 5$, the estimated f_p for the strategic attack is 2.7% (close to $\frac{1}{2^5}$); in Figure 3 where $L = 6$, the estimated f_p for the strategic attack is 1.623% (close to $\frac{1}{2^6}$). Based on this observation, we predict that when users are asked to select 8 items they like and 8 items they dislike, the false positive rate of the system is expected to be a value close to 0.391% for the strategic attack.

Remark: The above is a slight simplification of the actual simulations. We simulated 19 *combinations* of setup profiles. We refer to a combination by the L and D used for setup. For example, a (5,5) combination is one with 5 likes and 5 dislikes. As a result, we found that the combination of (6,6) results in the lowest false positive rates among the 19 cases when $L + D \leq 12$. No simulations have yet been performed for $(L, D) = (8, 8)$, but we have only extrapolated our findings from $(L, D) = (6, 6)$ to obtain our claimed error rates for sixteen questions.

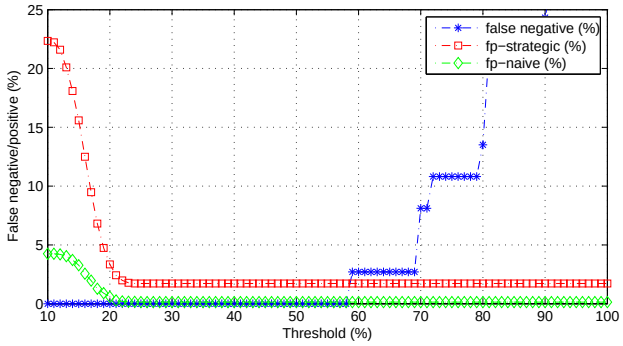

Figure 3: The relationship between false positive rates and the threshold of scores when 49000 setup profiles were simulated, where a hypothetical user is asked to select 6 items he likes and 6 items he dislikes.

5.3 The Accuracy of the Analysis

We now discuss the precision of our estimates on the false positive rate f_p. If the error of the estimate is denoted by ϵ, then f_p can be expressed by $f_p = \hat{f}_p \pm \epsilon$, where $\hat{f}_p$ is the estimated value of f_p. We assume that the false positive rate has a normal distribution. Such an assumption is reasonable when the sample size is large [3]. According to the principle of large-scale confidence intervals for a population proportion in statistics [3], the value of ϵ can be computed by

$$\epsilon = z_{\alpha/2}\sqrt{\hat{f}_p(1-\hat{f}_p)/n}, \tag{3}$$

where n is the number of setup profiles used to compute $\hat{f}_p$ and $z_{\alpha/2}$ is the critical value corresponding to the significance level α for a normal distribution (The critical values for typical distributions can be found in [3]). Solving for n in Equation (3) yields

$$n = \frac{z^2_{\alpha/2}\hat{f}_p(1-\hat{f}_p)}{\epsilon^2} \tag{4}$$

Equation (4) determines the required number of setup profiles to reach a certain precision ϵ of the estimated f_p. Figure 4 visualizes the relationship between the ϵ of the estimated f_p (for the strategic attack) and the required number of setup profiles when $\hat{f}_p = 1.623\%$ (computed in Section 5.2). It shows that to make $\epsilon = 0.125\%$, at least 39256 setup

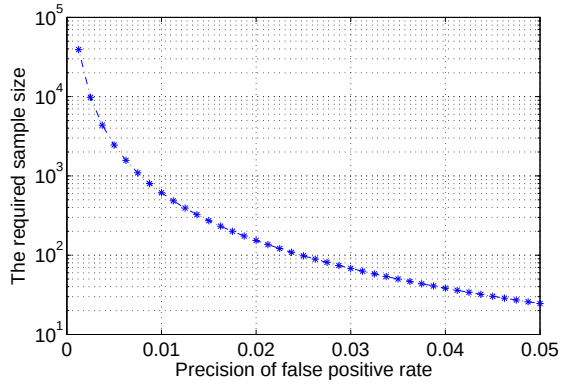

Figure 4: The relationship between the required number of setup profiles and the precision of the estimated false positive rate for the strategic attack when $\hat{f}_p = 1.623\%$ (computed in Section 5.2).

profiles are needed. This indicates that the use of the 49000 setup profiles in Section 5.2 provides enough precision, making the error of the estimate less than 0.125%.

Extrapolating to Sixteen Questions. The above findings support a false positive rate of $0.137 \pm 0.033\%$ against the naive adversary, and of $1.623 \pm 0.125\%$ against the strategic adversary. Assuming that each question used contributes the same amount to the said rates, it is possible to extrapolate the rates to more questions. This gives us false positive rates of 0.391% against the strategic adversary (the f_p will be close to 0 against the naive adversary) when users are asked to select 8 items they like and 8 items they dislike. We note that while it is not reasonable to expect that the extrapolation would be possible to make to an arbitrary number of questions (due largely to the fact that users have a limited number of likes and dislikes), it is highly reasonable to expect that it works when extrapolating from 12 questions to 16.

6 Conclusion and Future Work

We have described a new password reset system, improving on the work by Jakobsson et al. [9]. Our new user interface allows us to reduce the amount of interaction with users, resulting in a practically useful system while maintaining error rates. At the same time, we have described how the new interface introduces a new security feature: Protection against a site that attempts to obtain the answers to a user's security questions by asking him the same questions as another site did. While this does not offer any protection against man-in-the-middle attacks, it forces the attacker to interact with the targeted site, which could potentially lead to detection, at least when done on a large scale. It is an interesting open problem how to extend this protection towards more aggressive types of attacks.

We have evaluated the security of our proposed system against two types of realistic real-world attackers: the naive attacker (who knows nothing about the underlying probability distributions of the users he wishes to attack) and the strategic attacker (who knows aggregate distributions). We have not studied demographic differences, whether these are broken down by cultural background or by age group, gender, etc. It would be interesting to study these topics, and how to adjust what questions to use to maximize security given such insights. This is beyond the scope of this paper, though.

Moreover, it would be of great value to study how correlations between questions may affect the security; this does not correspond to an attack on the proposed system, it should be noted, but suggests that there is room for exploration as it comes to stronger adversaries. Such studies could also analyze how adversaries with partial knowledge of their victims may benefit, and how to modify the system to improve the security against such attacks. We have performed small-scale studies in which acquaintances, good friends, and family members attempt to impersonate a user, and observed that security is severely affected when a family member is the attacker, but only slightly so in other contexts. We note that the proposed system can be combined with other, orthogonal measures that increase the security against people in close contact with the intended victim of an attack. These measures do not have to provide any security against strangers. We note even though some of the questions have answers that can be guessed by friends or colleagues of a user, the same structure of the approach can be used without the same problem. There exist a lot of questions even friends or colleagues feel hard to guess the answers, examples including *Do you sleep on the left or right side of the bed?, Do you read the newspaper while eating breakfast?, etc.* (This would change the answers from "like" and "dislike" to "yes" and "no", with the third category being that the user does not select either during the setup phase.)

An important area of follow-up research is to study other adversarial models and analyze the security of the system in those contexts. Such studies may also suggest possible modifications to the design of the system that will let it withstand harsher

attacks or allow the server to detect attacks more easily.

Finally, another challenging problem is how to develop a large number of additional questions. It is evident that the security of the final system would be further enhanced with the addition of more questions, at the very least as far as protection against cloning attacks goes. This is not a trivial matter, nor is the automation of the whole process, and it remains an open question how best to address this issue.

We believe that the area of research on which we have embarked has a great potential for future improvement. Password reset, in our view, is one of the most neglected areas of security to date, and we hope that our enthusiasm will inspire others to make further progress.

Acknowledgements

The authors wish to thank Erik Stolterman, Ellen Isaacs, Philippe Golle, and Paul Stewart for insightful discussions, and Ariel Rabkin, Mark Felegyhazi, Ari Juels, Sid Stamm, and Mike Engling for feedback on previous versions of the manuscript.

References

[1] Farzaneh Asgharpour and Markus Jakobsson. Adaptive challenge questions algorithm in password reset/recovery. In *First International Workship on Security for Spontaneious Interaction: IWIISI'07*, Innsbruck, Austria, September 2007.

[2] Duane W Crawford, Geoffrey Godbey, and Ann C Crouter. The Stability of Leisure Preferences. *Journal of Leisure Research*, 18:96–115, 1986.

[3] Jay L. Devore. *Probability and Statistics for Engineering and Sciences*. Brooks/Cole Publishing Company, 1995.

[4] Carl Ellison, Chris Hall, Randy Milbert, and Bruce Schneier. Protecting secret keys with personal entropy. *Future Gener. Comput. Syst.*, 16(4):311–318, 2000.

[5] Niklas Frykholm and Ari Juels. Error-tolerant password recovery. In *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 1–9, New York, NY, USA, 2001. ACM.

[6] Virgil Griffith and Markus Jakobsson. Messin' with Texas, Deriving Mother's Maiden Names Using Public Records. *RSA CryptoBytes*, 8(1):18–28, 2007.

[7] William J. Haga and Moshe Zviran. Question-and-answer passwords: an empirical evaluation. *Inf. Syst.*, 16(3):335–343, 1991.

[8] Markus Jakobsson, Tom N. Jagatic, and Sid Stamm. Phishing for clues. `https://www.indiana.edu/~phishing/browser-recon/`, Last retrieved in June 2008.

[9] Markus Jakobsson, Erik Stolterman, Susanne Wetzel, and Liu Yang. Love and authentication. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 197–200, New York, NY, USA, 2008. ACM.

[10] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36, New York, NY, USA, 1999. ACM.

[11] `http://www.rsa.com/blog/blog_entry.aspx?id=1152`, last retrieved in June 2008.

[12] Mike Just. Designing and evaluating challenge-question systems. *IEEE Security and Privacy*, 2(5):32–39, 2004.

[13] Lawrence O'Gorman, Amit Bagga, and Jon Louis Bentley. Call center customer verification by query-directed passwords. In *Financial Cryptography*, pages 54–67, 2004.

[14] `http://www.voiceport.net/PasswordReset.aspx`, last retrieved in June 2008.

[15] Ariel Rabkin. Personal knowledge questions for fallback authentication: Security questions in the era of Facebook. In *SOUPS*, 2008.

[16] Arthur E. III Stamps. Of Time and Preference: Temporal Stability of Environmental Preferences. *Perceptual and Motor Skills*, Vol 85(3, Pt 1):883–896, December 1997.

[17] Douglas Stinson. *Cryptography: Theory and Practice*. CRC Press, 3rd edition, November 2005.