

# Quantifying the Security of Preference-based Authentication

Markus Jakobsson<sup>\*</sup>  
Palo Alto Research Center  
Palo Alto, CA 94304  
mjakobss@parc.com

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

## ABSTRACT

We describe a technique aimed at addressing longstanding problems 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%.

## Categories and Subject Descriptors

K.6.5 [Management of Computing and Information Systems]: Security and Protection—*Authentication*

## General Terms

Security, Design, Experimentation

## Keywords

Password reset, preference-based authentication, security question, simulation

## 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 [16]. To exacerbate the problem, many

<sup>\*</sup>Work performed for RavenWhite Inc., and while the author was with Indiana University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DIM'08, October 31, 2008, Fairfax, Virginia, USA.

Copyright 2008 ACM 978-1-60558-294-8/08/10 ...\$5.00.

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 his passwords. Unfortunately, the cost of a customer-service mediated password reset—currently averaging \$22 [15]—is much too expensive for most service providers.

In a recent paper by Jakobsson, Stolterman, Wetzel and Yang [9], an alternative method was introduced. Therein, a system based on user preferences was proposed in order to reduce the vulnerability to data-mining and maximize the success rate of legitimate reset attempts. The viability of such an approach is supported by findings in psychology [2, 13], showing that personal preferences remain stable for a long period of time. However, in spite of the desirable properties of the work by Jakobsson et al., its implementation remained impractical: To obtain a sufficient level of security against fraudulent access attempts—which for many commercial application is set below 1% false positive—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 the user interface of the setup phase 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 a *classification of all* preferences (as was done in [9]) to a *selection of some* preferences—those 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 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 user profiles are emulated from the initial distributions. These are then 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 from 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. While man-in-the-middle attacks remain possible, these are harder to carry out due to the real-time traces of traffic; these allow service providers to heuristically detect and block attacks based on commonly deployed techniques.

It is worth mentioning that if a server were to be compromised and user preference data was 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 note that our technique can be combined with a technique that requires a user’s ability to access an email account, or a registered phone number, etc. when he requests a password reset. In that case, a user is not allowed to even see the reset questions unless he accesses his email account, or phone, etc. Such a hybrid solution will make it harder for an attacker to capture the reset questions and impersonate a user, although still not impossible, as the current trend in theft of email credentials indicate.

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 businesses for fallback authentication. Financial institutions are well-motivated to secure the accounts of their customers, both to limit losses due to fraud (and thus poor PR), and to comply with regulations [19]. Yet, it is commonly agreed that the state of the art in security question-based authentication corresponds to a worrisome vulnerability [17]. A recent survey conducted by Rabkin [16] supports the common belief that many security questions suffer from weaknesses related to 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 [12]. The 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 [14]. The authors specified requirements for questions and answers and described how QDP can be combined with other techniques like PINs, addresses 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 is expected to have the same high cost [15] 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 by means of a user’s answers to a number of questions [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]. Preference-based authentication has the property of error-tolerance but achieves 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, preference-based authentication does not protect the profile information of users against the server; it may be possible to extend preference-

based authentication 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 named *preference-based authentication*. The underlying insight for their approach is that preferences are stable over a long period of time [2, 13]. Also, preferences are less likely to be publicly recorded than fact-based security questions, e.g., name of high school, 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, the system in [9] requires a user to provide his answers to a large number of questions when registering an account. This makes the previous preference-based system in [9] impractical. In this paper, we show that a redesign of how questions are selected can drastically reduce the number of questions needed 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, 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 20 questions for authentication, and a system asking too many questions for authentication purposes 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) on a small portion of the available items. Thus, instead of classifying each available item according to a 3-point Likert scale (like, no opinion, dislike), the new interface lets users select items that they either like or dislike from several categories of items which are dynamically selected from a big candidate set and are presented to a user in random order, as is shown in Figure 1. The majority of items are not selected and thus require no user action. The *authentication* interface is designed to only require a classification of

preferences (like or dislike) for the selected items displayed to the user in a random order.

#### Questions.

The selection of questions is a delicate task. In order to maximize the security of the system, it is important that the entropy of the distributions of answers for the questions used is large and that the correlation between answers is low. It is also important that the correlation to geographic regions and other user demographics is low. It is clear that users in different countries and user classes may exhibit different distributions. Thus, it may be of value to develop questions specifically to various countries and demographics. (Our current set of questions has been optimized for a general U.S. population.)

Moreover, it is of practical relevance that the questions used in the system do not evoke extreme opinions (of the kind that may cause users to expose their opinions in other contexts such as, e.g., in social networks), but that most users still can find reasonably strong opinions reasonably easily. The development of appropriate questions is just as much an art as a science, and it is an area with promising opportunities for more in-depth research.

#### Setup.

During the setup phase, a user is asked to select  $L$  items he likes and  $D$  items he dislikes from several categories of topics (e.g., *Playing baseball, Karaoke, Gardening, etc.*). For each user, only a subset of items is presented for selection. The subset is chosen in a random way from a larger candidate item set, and the order of the items in each category is randomized, as is the order of the categories. This avoids a static view of the questions, which would otherwise have introduced a bias in terms of what questions were typically selected. 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 which is 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. See [www.blue-moon-authentication.com](http://www.blue-moon-authentication.com) for a live system.

#### Authentication.

During the authentication phase, the user first presents his username for which the server then 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 answers: like or dislike. The correctness of the answers is scored using an approach described in [9], so as to assign some positive points to each correctly answered question and some negative points to each incorrectly answered question; the exact number of points depends on the entropy of the distribution of answers to these questions among the population considered. The authentication succeeds if the total score is above a preset threshold.

Returning to the differences in user interfaces, we see that the user interface we propose represents a usability improvement over the interface proposed in [9] where users have to classify a much larger number of topics for an equivalent security assurance. In our version, a user selects what to classify during the setup phase and only classifies these topics during authentication. Our proposed system requires a

Items		
TV	Interests	Food
Places	Music	Sports
Gardening	<input type="button" value="Like"/>	<input type="button" value="Dislike"/>
Cats	<input type="button" value="Like"/>	<input type="button" value="Dislike"/>
Crafts	<input type="button" value="Like"/>	<input type="button" value="Dislike"/>
Motorcycles	<input type="button" value="Like"/>	<input type="button" value="Dislike"/>
Video games	<input type="button" value="Like"/>	<input type="button" value="Dislike"/>
Reading comics	<input type="button" value="Like"/>	<input type="button" value="Dislike"/>

Likes	
1. Going to garage sales	<input type="button" value="x"/>
2. Classical music	<input type="button" value="x"/>
3. Poetry	<input type="button" value="x"/>
4.	
5.	
6.	
7.	
8.	

(Choose 5 more Likes)

Dislikes	
1. Watching auto racing	<input type="button" value="x"/>
2. Game shows	<input type="button" value="x"/>
3.	
4.	
5.	
6.	
7.	
8.	

(Choose 6 more Dislikes)

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

total of 16 topics to be selected and classified. 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.

### 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 the ratio  $S_A/S_S$ , 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 the uncertainty of its answer for a random guess, which is measured by its information entropy [18]. 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 penalized (by receiving negative points). The penalty 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 penalty for providing an incorrect one. (If it was true that a legitimate user would always answer all questions correctly during authentication, then the optimal parameter choice for the weights would be set to negative infinity. However, since we must allow users to make a small number of mistakes, that is not the parameter choice we make.)

## 4. ADVERSARIAL MODEL

We study the security of the scheme by investigating how likely it is that 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 authentication site. In the following, 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 know that users are asked to select  $L$  items they like and  $D$  items they dislike during the setup phase. However, it is assumed

that the adversary knows nothing of the relative selection frequencies of the available items. To impersonate a user, the adversary randomly selects the choice *like* for  $L$  items and the choice *dislike* for  $D$  items during an authentication attempt. This is a realistic assumption for most real-life adversaries who have limited information or expertise of the targeted systems. As a case in point, most current phishing attacks do not use advanced 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 the parameters  $L$  and  $D$ , an adversary knows the distributions of answers to the questions used by the system. In particular, for each item used during the authentication phase, the adversary knows the percentages of users who chose *like* and *dislike* respectively. We call these percentages the *like rate* and the *dislike rate*, denoted by  $p$  and  $q$ . The like rates and dislike rates used in this type of attack were obtained from an experiment in [9]. 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 the choice *like* for  $L$  items and the choice *dislike* for  $D$  items such that  $p_{i_1} \times \dots \times p_{i_L} \times q_{j_1} \times \dots \times q_{j_D}$  is maximized, where  $(i_1, \dots, i_L, j_1, \dots, j_D)$  is a permutation of the indices  $(1, 2, \dots, L + D)$  for the  $L + D$  items.

The strategy of both our adversaries differs from that of 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 that adversary 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 achieve 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.

### 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.

### Question-Cloning Attack.

In a question-cloning attack, the adversary poses a victim with a set of questions, and asks for the answers to these. The pretense may be that the victim user is setting up an account with a site controlled by the attacker, not knowing that this is a malicious site. The adversary succeeds if he learns the answers to questions used by victim user at another site; we refer to this attack as a *question-cloning* attack, since that is exactly the circumstance when the attack is successful: when the adversary asks the same questions as are used elsewhere.

## 5. QUANTIFYING THE SECURITY

The security features of our approach have been evaluated in three ways: user experiments, user emulations, and attacker simulations. The goal of the experiments was to obtain user data to be used to assess error rates. Due to a 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 (naive or strategic) 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 6800 user profiles were 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 and dislikes, our experiments do not consider only this parameter choice. However, our exposition in the paper 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

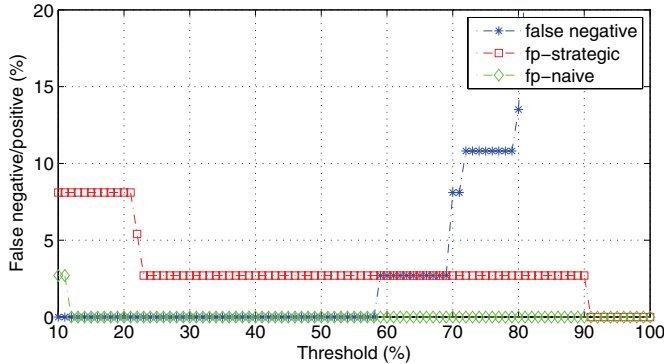
We conducted an experiment involving 37 human subjects. Unlike our final system shown in Figure 1 which asks

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 phases. Each user was allowed to perform one authentication attempt. All participants completed both the setup and authentication phases. Tests (of a small sample size) showed that it takes a user approximately two minutes on average to complete the setup, and about 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 between the number of unsuccessful authentication attempts (i.e., attempts resulting in a score lower than  $T$ ) and 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 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 user profiles being attacked. As described in Section 3, the parameter  $c$  is used to adjust the quantity of punishment for incorrect answers. From the point of view of system design, choosing a high value of  $c$  can severely penalize incorrect answers 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 answers 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 shows the variation of false negative and false positive rates with respect to the value of the 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 the naive and strategic attack for the 37 user 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. A 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%<sup>1</sup>. This finding led us to consider increasing  $L$  and  $D$  in order to obtain lower false positive rates. As we will see later, the false positive rate for the strategic and

<sup>1</sup>While the values we determined for  $c$  and  $T$  are suitable, they may not be optimal. I.e., there may be parameter choices that lead to lower error rates.



**Figure 2: The false positive and false negative rates as a function of the threshold  $T$  for  $c = 6$ , when users were asked to select 5 items they like and 5 items they dislike during the setup phase.**

naive attack decreased to 0 and  $0.011 \pm 0.025\%$ <sup>2</sup> respectively when users were required to select 8 items they like and 8 items they dislike.

## 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 and drawing on preference data of 400+ subjects (see [9].) We denote this process by *EmulSetup*. Executing *EmulSetup* once will generate a user profile for a hypothetical user, where the profile contains  $L$  items liked and  $D$  items disliked by the hypothetical user. The profiles generated by *EmulSetup* are believed to have the same distribution as the profiles of real users in real experiments. This will be explained further in the following subsections of this paper. By repeatedly executing *EmulSetup*, we generated a large number of hypothetical user profiles. The second step of the simulation is to apply both the naive and strategic attack to the hypothetical 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 *EmulSetup* function emulates how users perform the setup using the interface described in Section 3. In *EmulSetup*, a profile is generated by presenting several lists of items to a hypothetical user who then selects items according to the known probability distributions, as observed in [9]. 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 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

<sup>2</sup>The 0.025% denotes the precision of the estimate. Further details are provided in Section 5.3

a probability of  $0.3/0.6 = 50\%$ , select *Rap music* with 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 this approach does not take correlations into consideration, that is not a limitation in the context of the adversaries we consider.)

### 5.2.2 Mathematical Description

Now we provide 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  $p_1, p_2, \dots, p_m$  and the corresponding dislike rates are  $q_1, q_2, \dots, q_m$ . The like rates and dislike rates for all items were obtained from an experiment involving 423 participants in [9]. Assume the selections of items are independent (which is reasonable when the size of the candidate set is large). Then, a hypothetical user will select like for the  $i$ th item in the list with a probability of

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

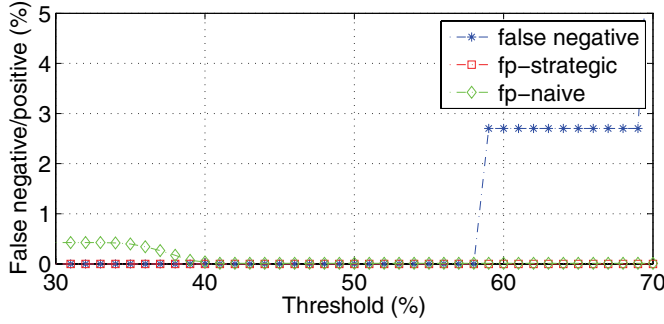
where  $X$  denotes the index of an item in this list.

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  $I_i = [S_{i-1}, S_i)$  it falls into for  $i = 1, \dots, m$  where  $S_i = \sum_{j=0}^i P_j$  and  $P_0 = 0$ . If the random value falls into  $I_i$ , then the  $i$ th 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 profiles generated by real users in Section 5.1 and the profiles generated by *EmulSetup*. The results confirm that they are not statistically different, with a significance level of 0.05. This provides further evidence that the profiles generated by *EmulSetup* 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 profiles generated by *EmulSetup* 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 [3], 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 profiles for hypothetical users in order to obtain high accuracy in our evaluation. The number of profiles we generated was 6800. How this number was determined will be discussed later. In our emulation, each of the 6800 hypothetical users picks 8 items he likes and 8 items he dislikes as his setup. Then, we applied the naive and strategic attacks to the generated 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  $c = 4$ . For any threshold value between 23% and 58%, the false positive rate for the strategic attack is 0. For the naive attack, the false positive rate is  $0.011 \pm 0.025\%$ . The significance level of our estimates is 5%.



**Figure 3:** The relationship between the false positive rates and the threshold of scores when 6800 profiles were simulated ( $c = 4$ ), where a hypothetical user is asked to select 8 items he likes and 8 items he dislikes.

By comparing the false positive rates in Figure 2 and Figure 3, one can observe that when  $L = D$  then  $f_p$  corresponding to the strategic attack can be bounded above by  $\frac{1}{2L}$ . For example, in Figure 2 where  $L = 5$ , the estimated  $f_p$  for the strategic attack is 2.7% (i.e., less than  $\frac{1}{25}$ ); in Figure 3 where  $L = 8$ , the estimated  $f_p$  for the strategic attack is 0 (i.e., less than  $\frac{1}{28}$ ).

**Remark.**

It is not a priori evident that  $L = D$  leads to the lowest error rates. We performed simulations of 19 different parameter choices for  $L$  and  $D$ , such that  $L + D = 16$ , and estimated the error rates for these. Whereas this may depend on the total number of questions selected (i.e., the sum  $L + D$ ), we found that setting  $L = D$  leads to the most favorable rates for  $L + D = 16$ .

### 5.3 The Accuracy of the Analysis

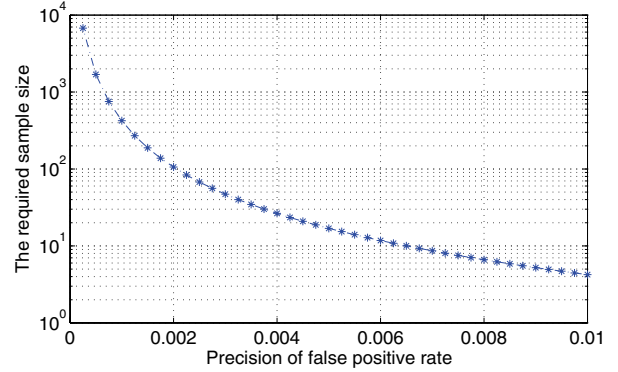
We now discuss the precision of our estimates on the false positive rate  $f_p$ . If the error of the estimate is denoted by  $\epsilon$ , then  $f_p$  can be expressed by  $f_p = \hat{f}_p \pm \epsilon$ , where  $\hat{f}_p$  is the estimated value of  $f_p$ . We assume that the false positive rate has 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  $\epsilon$  can be computed as

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

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

$$n = \frac{z_{\alpha/2}^2 \hat{f}_p(1 - \hat{f}_p)}{\epsilon^2}. \quad (3)$$

Equation (3) determines the required number of profiles to reach a certain precision  $\epsilon$  for the estimated  $f_p$ . Figure 4 visualizes the relationship between the  $\epsilon$  of the estimated  $f_p$  (for the naive attack) and the required number of profiles when  $\hat{f}_p = 0.011\%$  (computed in Section 5.2). It shows that in order to make  $\epsilon = 0.025\%$ , at least 6771 profiles are



**Figure 4:** The relationship between the required number of profiles and the precision of the estimated false positive rate for the naive attack when  $\hat{f}_p = 0.011\%$  (computed in Section 5.2). For the strategic attack, the  $\hat{f}_p = 0$  causes the denominator of equation (3) to be zero. Thus, the required sample size cannot be determined in this special case. However, we strongly believe that the number of profiles which assures sufficient precision for  $f_p$  for the naive attack also provides reasonable precision in the case of the strategic attack.

needed. Thus, using 6800 profiles in Section 5.2 provides sufficient precision, resulting in an error of the estimate less than 0.025%.

### 5.4 Security against Question-Cloning

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. Thus, while a normal attack would focus on learning a victim’s answers, this attack would aim at learning the *questions* asked to a victim—in order to ask the victim these questions and *then* learn the answers. We may refer to this as a *two-phase* attack. Given the assumption that the victim is willing to set up a profile with the pirate site (not knowing of its bad intentions), it is clear that the second phase of the attack is easy to perform, and the system must stop the attacker from performing the first phase. The first phase is trivial for most current systems, as there is a very limited number of questions used, and the victim can be posed with all of these. To carry out the first phase of the attack on our system, it is not sufficient to know what questions *can* be asked, since it is a very large number. An attacker needs to know what questions *will* be asked. To do that, the attacker has to attempt to reset the victim’s password—only then will he learn the questions. If we require access to a registered email account or phone as an orthogonal security mechanism, then this makes this type of attack very difficult to perpetrate. This is a benefit that is derived from the user interface we propose, 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 security of this feature will increase with the number of selectable topics.

## 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 that 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. Extending this protection towards more aggressive types of attack is an interesting open problem.

We have evaluated the security of our proposed system against two types of realistic 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.

We have considered an adversarial model in which all distributions are known, but correlations are not used by an attacker. Preliminary experiments suggest that most of the proposed questions have relatively low pairwise correlation, and the removal of a few questions is likely to curtail the effects of stronger adversarial models. However, this is not the only type of model worth studying in more detail. For example, it is also worth considering attackers with partial personal knowledge of their victims. 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 affected in other cases. However, it is important to recognize that other password reset methods would exhibit similar behavior. Also, it is important to recognize that most of these attacks would be addressed in a satisfactory manner by methods in which a user needs to show access to a registered email account, phone number or other personal account. These are techniques that are currently in use in real-world password reset applications. A further study of the practical security of such hybrid systems would be of high interest, but it is not evident how to study non-deployed systems in such contexts.

To protect against friends and colleagues, one could add questions that are difficult to guess the answers by people close to the victim. There exist a lot of questions for which the answers are difficult to guess even by friends or colleagues. Examples include *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, as it becomes more difficult for an adversarial site to get overlapping sets of answers by sheer luck. 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.

## Acknowledgments

The authors wish to thank Erik Stolterman, Ellen Isaacs, Philippe Golle, and Paul Stewart for insightful discussions; Ariel Rabkin, Mark Felegyhazi, Ari Juels, Sid Stamm, Mike Engling, Jared Cordasco, and John Hite for feedback on previous versions of the manuscript. Thanks to Susan Schept for helpful discussions on the stability of preferences.

## 7. REFERENCES

- [1] F. Asgharpour and M. Jakobsson. Adaptive Challenge Questions Algorithm in Password Reset/Recovery. In *First International Workshop on Security for Spontaneous Interaction: IWISI'07*, Innsbruck, Austria, September 2007.
- [2] D. W. Crawford, G. Godbey, and A. C. Crouter. The Stability of Leisure Preferences. *Journal of Leisure Research*, 18:96–115, 1986.
- [3] J. L. Devore. *Probability and Statistics for Engineering and Sciences*. Brooks/Cole Publishing Company, 1995.
- [4] C. Ellison, C. Hall, R. Milbert, and B. Schneier. Protecting Secret Keys with Personal Entropy. *Future Gener. Comput. Syst.*, 16(4):311–318, 2000.
- [5] N. Frykholm and A. 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] V. Griffith and M. Jakobsson. Messin' with Texas, Deriving Mother's Maiden Names Using Public Records. *RSA CryptoBytes*, 8(1):18–28, 2007.
- [7] W. J. Haga and M. Zviran. Question-and-Answer Passwords: an Empirical Evaluation. *Inf. Syst.*, 16(3):335–343, 1991.
- [8] M. Jakobsson, T. N. Jagatic, and S. Stamm. Phishing for Clues. <https://www.indiana.edu/~phishing/browser-recon/>, last retrieved in August 2008.
- [9] M. Jakobsson, E. Stolterman, S. Wetzel, and L. 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] A. Juels and M. 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] [www.rsa.com/blog/blog\\_entry.aspx?id=1152](http://www.rsa.com/blog/blog_entry.aspx?id=1152), last retrieved in August 2008.
- [12] M. Just. Designing and Evaluating Challenge-question Systems. *IEEE Security and Privacy*, 2(5):32–39, 2004.

- [13] G. F. Kuder. The Stability of Preference Items. *Journal of Social Psychology*, pages 41–50, 10 1939.
- [14] L. O’Gorman, A. Bagga, and J. L. Bentley. Call Center Customer Verification by Query-Directed Passwords. In *Financial Cryptography*, pages 54–67, 2004.
- [15] [www.voiceport.net/PasswordReset.aspx](http://www.voiceport.net/PasswordReset.aspx), last retrieved in August 2008.
- [16] A. Rabkin. Personal Knowledge Questions for Fallback Authentication: Security Questions in the Era of Facebook. In *SOUPS*, 2008.
- [17] [www.schneier.com/blog/archives/2005/02/the\\_curse\\_of\\_th.html](http://www.schneier.com/blog/archives/2005/02/the_curse_of_th.html), last retrieved in August 2008.
- [18] D. Stinson. *Cryptography: Theory and Practice*. CRC Press, 3rd edition, November 2005.
- [19] [www2.csoonline.com/article/221068/Strong\\_Authentication\\_for\\_Online\\_Banking\\_Success\\_Factors?page=1](http://www2.csoonline.com/article/221068/Strong_Authentication_for_Online_Banking_Success_Factors?page=1), last retrieved in August 2008.