Password Portfolios and the Finite-Effort User: Sustainably Managing Large Numbers of Accounts*

Dinei Florêncio and Cormac Herley
Microsoft Research, Redmond, USA

Paul C. van Oorschot
Carleton University, Ottawa, Canada

Abstract. We explore how to manage a portfolio of passwords. We review why mandating exclusively strong passwords with no re-use gives users an impossible task as portfolio size grows. We find that approaches justified by loss-minimization alone, and those that ignore important attack vectors (e.g., vectors exploiting re-use), are amenable to analysis but unrealistic. In contrast, we propose, model and analyze portfolio management under a realistic attack suite, with an objective function costing both loss and user effort. Our findings directly challenge accepted wisdom and conventional advice. We find, for example, that a portfolio strategy ruling out weak passwords or password re-use is sub-optimal. We give an optimal solution for how to group accounts for re-use, and model-based principles for portfolio management.

1 Introduction

Due to the growth in online services, many users now manage dozens of password-protected accounts. Many service providers, awareness campaigns (US DHS [1]), and government entities (US-CERT [2]) stress two foundations for password security:

A1: Passwords should be random and strong; and

A2: Passwords should not be re-used across accounts.

Despite this, users have long been observed to choose weak passwords. Leaked datasets, such as the 32 million plaintext passwords from Rockyou, reveal that most users fall far short of following "traditional" advice on password strength. Evidence also indicates widespread password re-use [21]. While admonitions against this are almost universal, ignoring that advice seems equally universal. Clearly, users find managing a large password portfolio burdensome. Both password re-use, and choosing weak passwords, remain popular coping strategies.

Numerous efforts have been made to address the neglect of password strength by users. Many sites stress the importance of, and offer tips on how strong passwords can be made easier to construct and remember; e.g., US-CERT [2] and others commonly suggest passphrase-based and other mnemonic approaches. But while significant attention has been devoted to motivating and helping users choose strong individual passwords, there is little guidance on how to choose and manage large numbers of them. We aim to give, and justify, such guidance.

We explore how a large portfolio of passwords can be maintained without ignoring that users have limited abilities. Can password re-use be part of sensible portfolio management, or is it never justifiable? Is a unique strong password for every account, including blog sites and throw-away accounts, truly the best use of limited human memory resources? In practice, many users gather accounts into groups that re-use a password, but little guidance exists on choosing appropriate groups. Given that re-use does and will happen, we explore how to do so in a principled way, and answer these questions.

Our findings directly challenge some conventional wisdom. For example, we find: *strategies that rule out password re-use or the use of weak passwords are sub-optimal*. Both are valuable tools in balancing the allocation of effort between higher and lower value accounts.

We first review password-related demands on users, and consider users' options under the reasonable but too-rare assumption of *finite user effort*. This realism yields an inherent trade-off between two desired outcomes: greater password strength and avoiding re-use. Acknowledging fixed user effort budgets, more of one means less of the other.

We explore the implications of password re-use, and outline an optimal password-sharing strategy: for a fixed number of passwords and a given set of accounts, how to partition accounts to minimize total expected loss. Loss analysis is greatly complicated by cross-contamination issues due to password re-use. We address this by a novel

*USENIX Security 2014, August 20-22.

partitioning of attacks into three broad classes covering the major threat vectors, itself of independent interest.

2 Related Work

In 2000, Dhamija and Perrig [18] interviewed 30 participants reporting 1–7 unique passwords for 10–50 web sites. Circa 2001, Sasse and Brostoff [45] surveyed 144 employees reporting on average 16 passwords including non-online activity. A 2004 survey of 218 college students by Brown et al. [11] indicated on average 8.18 password accounts serviced by 4.45 unique passwords. In 2006 Gaw and Felten [24] surveyed 58 (mainly student) participants by online questionnaire with in-lab follow-up of 49, exploring how users manage online passwords, the extent of reuse and how users justify it, and the use of related passwords; they reported on average 13 passwords and found reuse increased over time—new accounts accumulated faster than new passwords. Riley's 2006 survey [44] of 315 college students (8.5 accounts on average) reported: 74.9% have a set of predetermined passwords they frequently re-use; 54.6% very frequently or always use a same password for multiple accounts; 33% use some variation of a same password for multiple accounts; and 60% do not vary the complexity of their passwords with the nature of a site. In a 2007 study of password use/re-use across three months by over a half million users, Florêncio and Herley [21] reported on average 25 accounts serviced by 6.5 unique passwords, re-used passwords used on average at 5.7 sites, and strong passwords re-used less.

Notoatmodo's 2007 thesis [42] explored password re-use and users' perspectives of their real-world passwords—and especially relevant to our work, how users mentally group both accounts and passwords into categories, relationships between account and password groups, and details of users' reasons both for, and for not, reusing passwords. The 26 participants surveyed had on average 12.9 accounts and 8.1 passwords; most reused passwords (132 of 336 accounts had unique passwords). Reuse was found again (see above) to increase with number of accounts. A hypothesis progressed was that users manage their accounts and passwords by mentally separating both into categories based on perceived account similarities and password similarities,[1] Regarding grouping accounts, and which accounts they felt were "high importance", most participants had only one high importance account group (1.54 such groups on average), and high importance groups were found to be smaller (fewer accounts per group: mean 1.84 vs. 2.78 for low importance groups). 45% reported reusing at least one password from a high importance group vs. 96% reusing at least one password from a low importance group; 70% had passwords exclusively used for an account in high importance groups (2.9 such passwords on average). In line with our views, Notoatmodo suggests *"reusing passwords on unimportant accounts which contain no sensitive information should not be discouraged ... Expecting users to create unique, strong passwords for all their accounts is ... unreasonable ... Instead, users should be educated to identify which accounts [not to] reuse passwords on."* While granting that password re-use is dangerous, Karp [36] also argues for re-use (*"human nature being what it is, not reusing passwords is equally dangerous"*) but in a different direction: by a password manager tool re-using a user password as a master password combined with details of a target site (e.g., site name) for site-specific passwords.

The "domino effect" of password re-use is well-documented (e.g., Ives et al. [32]; Gouda et al. [25]). The need for re-use is exacerbated by large numbers of passwords consuming user's memory capacity [3]. In scarce empirical work on implications of password re-use, Bonneau and Preibusch [9] analyze password implementations across 150 free websites, explaining technical means by which password re-use allows low-security sites—often unmotivated to spend effort or user experience securing passwords—to compromise high-security sites. The same authors [43] explore this question as a negative externality of password policies, finding a tragedy of the commons whereby sites with the lowest security needs can endanger those with the highest. Florêncio and Herley [22] find that the imposition of stringent password policies is better correlated with insulation from the consequences of poor usability than the need for greater security.

While the degree of password re-use naturally varies with the users studied, their circumstances and environment at a give time, evidence clearly shows it is widespread. The accuracy of self-reported re-use statistics is debatable, but a lower bound on re-use in real life is possible from leaked password databases from two different sites: from each database, recover a (userid, password) list, find userids common to both (e.g., re-used email addresses), then count password re-use instances. Das *et al.* [17] estimate that 43-51% of users re-use passwords across sites, and give algorithms that improve an attacker's ability to exploit this fact; this exceeds the 12-20% rate of some earlier studies noted above [24, 21]. Lemos [38] reports that intersection of the breached database pair (Yahoo Voices, Sony online) and (Sony online, Gawker) found usernames had re-used passwords across two sites 59% and two-thirds of the

[1] Examples of similarities for grouping passwords: "school stuff", email accounts, online banking, and semantic properties related to security (e.g., overall length, number of letters). Examples for account grouping: type of service related to the account (e.g., financial, education, communication), similar levels of risk or importance.

time in the two pairs. RockYou's leaked dataset [30] was explored by Bonneau [7, p.83] and Weir et al. [48]. Zhang et al. [49] easily predict new passwords from old when password aging policies force updates.

Over a 2011 two-week diary study of password use by Hayashi et al. [28], 20 participants reported 8.6 accounts on average, and to not use any memory aids for 60% of accounts; 19 of 20 said they reused passwords for multiple accounts. This may indicate under-estimating password re-use risk vs. writing passwords down. From a 2010 one-week diary-based study wherein 32 staff from two organizations produced just over 6 passwords each, Inglesant et al. [31] suggest that password polices be designed not to maximize password strength but rather to aid users in setting strengths appropriate to specific use contexts. Grawemeyer et al. [26] explore re-use among coping strategies in managing collections of passwords, in a detailed 2011 diary study of 22 participants over 7 days. In 2014, Stobert et al. [47] also explore user coping strategies for managing passwords, with guided interviews and questionnaires on 27 participants—noting as a user concern "rationing effort to best protect important accounts", and that "many participants [reported] having a specific password that they reused widely on accounts of low interest, low importance, or infrequent use".

The idea of grouping passwords, e.g., by level of importance, has seen little academic study, but Cheswick et al. [15, pp.140-141] suggested four categories: worthless, slightly important, quite secure, and top security. Cheswick more recently [13] suggests three classes: those that (a) have no importance; (b) are inconvenient if stolen; or (c) result in a major problem if abused. Five categories each are given by Grosse et al. [27] (based on account value) and Florêncio et al. [23] (based on consequence of compromise). Cheswick [14] also reviews common password guidance, and the ongoing suitability of circa-1985 U.S. government password guidelines. Florêncio and Herley suggest defender goals are well modelled by minimizing loss plus effort [20].

Nithyanand et al. [41] explore issues related to password re-use, under an attack model focused on server-side breakin (excluding phishing, client-side malware); seek solutions to maximize "remaining value" (i.e., minimize loss, vs. loss plus effort herein); show their password allocation problem is NP-complete; and find heuristic solutions to special cases (for accounts of equal value, with identical compromise probabilities, etc.).

3 The Difficulty of Managing a Portfolio

Issues related to complexities of human memory, encoding and recalling information (see [10]) currently preclude a satisfactory cognitive model or measure of the load passwords place on users. Nonetheless we begin with a naive model to highlight impossible-to-meet assumptions, and to position and motivate later discussion. We stress that *our later modeling* (Sections 4 onward) *abandons these assumptions and this naive model*, tackling a more realistic setting. We acknowledge that our equations in this Section, e.g., for the difficulty of associating passwords with accounts, give at best crude estimates. We emphasize also that this paper considers ordinary text passwords, not text or graphical variations using cues; we make no claims regarding such schemes.

An active web-user may have a hundred or more password-protected accounts. Ideally a user with N accounts chooses N strong passwords. If passwords were random collections of equi-probable characters, the difficulty of remembering them would be related to their length. Assume each such password is $\lg S$ bits. The effort required to manage the portfolio might naively appear to be $N \lg S$. But beyond remembering N passwords, users must remember which matches which account. We now explicitly consider this often overlooked sub-task.

3.1 Matching Passwords to Accounts

There are $N \cdot (N-1) \cdots 1 = N!$ possible mappings of N unique passwords to accounts; no encoding of this information uses less than $\lg(N!)$ bits, unless passwords contain clues as to which site they serve, violating A1 above. Thus the number of bits to be remembered to manage a portfolio of N passwords, each of $\lg S$ bits, is at least:

$$E(N) = N \cdot \lg S + \lg(N!). \tag{1}$$

(As noted above, this approximation fails to address the complexities of human cognition, but suffices for the argument below.) Clearly this grows rapidly with N, the second term super-linearly by Stirling's approximation ($\ln N! \approx N \ln N - N$). Consider a conscientious user, with $N = 100$ accounts. Choosing unique random passwords of 40 bits for each account rewards him with the obligation to remember $100 \times 40 + \lg(100!) = 4525$ bits (equivalent to 1362 random digits or 170 random 8-digit PINs). This burden far exceeds what users can manage by memorization (*i.e.*, without other aids); for most it is insupportable. How can users reduce it? An obvious shortcut, with significant side effect, is to choose weaker (less random) passwords; the linear dependence on $\lg S$ suggests reducing strength as much as possible.

While using weaker passwords clearly reduces the first term of (1), no matter how weak N distinct passwords are, the second term is unaffected. Considering that term alone, $N = 100$ yields $\lg(N!) = 525$. This is double the $\lg(52!) = 226$ bits required to memorize the order of a shuffled card deck, and equivalent to remembering 158 random digits—random since as noted, no encoding of an $N \times N$ assignment takes fewer than $\lg(N!)$ bits. Thus,

the assignment burden alone, including the problem of *password interference* [16], is evidently beyond a reasonable expectation of users.

So the two staples A1, A2 of password advice appear impossible to meet individually, let alone simultaneously. How do users proceed? They "cheat" on A1 by choosing passwords far weaker than advised. But this isn't enough—no matter how weak the passwords, a user must still remember $\lg(N!)$ random bits for password assignment. A further coping strategy is needed.

3.2 Password Re-use as a Coping Strategy

Consider next a user with N accounts using $G \leq N$ passwords to cover them. Assume for now each password is used at $n = N/G$ accounts, that the password-to-group assignment is random and (for simplicity) that G divides N. The burden of remembering passwords drops to $G \cdot \lg S$ bits. What of the further burden of remembering which password goes where? The G groups of accounts each have $n = N/G$ elements. There are C_n^N possible combinations for the first group, C_n^{N-n} for the second, etc., so the number of possible assignments of N accounts to G equal-sized groups is:

$$\left(\begin{array}{c} N \\ n \end{array}\right) \cdot \left(\begin{array}{c} N-n \\ n \end{array}\right) \cdots \left(\begin{array}{c} n \\ n \end{array}\right) = \frac{N!}{(n!)^G}.$$

Thus the user effort (memory burden in bits) drops to:

$$\begin{aligned} E_G(N) &= G\lg S + \lg(N!) - G \cdot \lg(n!) \qquad (2) \\ &\approx G\lg S + N\lg G \end{aligned}$$

the last line following by Stirling's approximation again.

Now compare the burden of managing a portfolio with and without password re-use. For example, even if $\lg S$ is as low as 20, from (2), the burden of managing 100 accounts with 10 passwords is $E_{10}(100) = 506$ bits, while from (1) the burden of doing so with 100 is $E(100) = 2525$ bits. Thus, in this instance, password re-use reduces the memorization burden by a factor of five.

3.3 Tradeoff: Re-use & Password Strength

What other solutions use the same effort? A portfolio of N passwords can be managed in many ways. If $E_G(N)$ is fixed then $\lg S \approx (E_G(N) - N\lg G)/G$. So, $\lg S$ falls faster than $1/G$: doubling the number of passwords more than halves the number of bits per password. So, if $G = N$ (no password re-use), then $\lg S$ must be small.

Fig.1 shows the locus of solutions in the G-$\lg S$ plane when $N = 100$ and the budget is $E_G(100) = 400, 550$ and 700 bits. This reveals the essential tradeoff: less re-use (*i.e.*, increasing G) implies weaker passwords. For example, at fixed effort $E_G(N) = 400$, two possible operating points are $(G = 4, \lg S = 52.5)$ and $(G = 5, \lg S = 36.2)$. At fixed effort, the question is not whether achieving password strength and avoiding re-use are good, but how these relative goods are best traded off. Deciding, e.g., between these two operating points depends on whether reducing password re-use (by increasing G from 4 to 5) reduces the risk of harm by more or less than reducing password strength (from 52.5 to 36.2 bits). The rapid decline of $\lg S$ in Fig.1 as G increases suggests that, far from being unallowable, password re-use is a necessary and sensible tool in managing a portfolio. Re-use appears unavoidable if $\lg S$ must remain above some minimum (and effort below some maximum). Fig.1 further suggests that G should be small: high values of G seem to imply very low values of $\lg S$. This enormous saving in user effort that password re-use provides may explain its ongoing prevalence in practice [24, 19, 21, 45].

Note on entropy: caution is needed to avoid historical pitfalls such as assuming particular ranges for $\lg S$ based on metrics appropriate only for *random* passwords, or misleading rules-of-thumb on what is necessary to withstand attack. We intend $\lg S$ to represent user effort to remember a password, not attacker guessing difficulty; the two may be correlated but should not be used in place of each other. For example, if a user has 7 unique passwords, and each names a major Hawaiian island, then $\lg S = \lg 7 \approx 2.8$ bits. But this offers little guide on how hard these passwords are to guess. It is also well understood [48, 8, 37, 39] that $t \cdot \lg C$ and NIST's crude password entropy estimate [12], significantly overestimate the difficulty of guessing *user-chosen* length-t passwords from C-character alphabets. Equally, such metrics must not be mis-used to estimate users' capabilities or effort, lest we drastically over-estimate what a reasonable cognitive burden is.

4 Objective Function: Loss + Effort

Suppose a user has N password-protected accounts. Advice such as A1, A2 implicitly assume the goal is to minimize loss. Let P_i be the probability of compromise in a given period (e.g., per year, under the current password management strategy), and L_i the loss endured upon such compromise. We intend that P_i capture the probability that an attacker gains the means to access account i, whether or not that means is used or results in loss. We intend L_i to capture the *expected value* of the consequences of account compromise (regardless of attack vector), including direct losses and any indirect costs involved in remediation. The total expected loss is

$$L = \sum_{i=1}^{N} P_i \cdot L_i. \qquad (3)$$

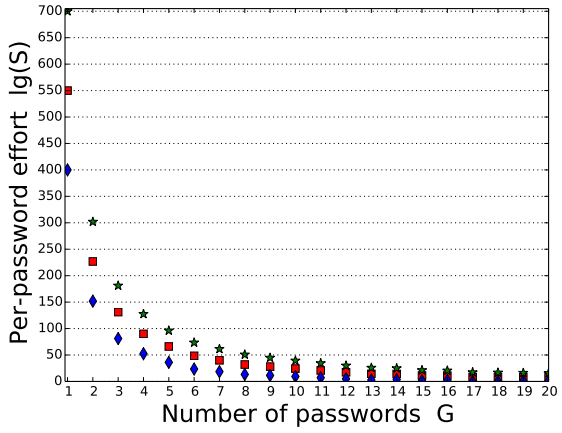

Figure 1: Locus of achievable solutions trading off re-use and password strength for fixed effort $E_G(N) = 400, 550, 700$ in (2) at $N = 100$. Note that when effort is kept constant, lower levels of re-use are only achieved by having weaker passwords.

A major complication we will find—and defer to Section 5—is that some attacks affect more than one account; e.g., malware and system attacks affect all accounts, and if passwords are re-used then an attack against one account can affect many others. But for now, suppose that attacks are only against individual accounts. Then the P_i depend on effort E_i devoted to account i but not to E_j. The probability of compromise $P_i = P_i(E_i)$ is presumably monotonically non-increasing with effort. Investing more effort generally reduces P_i; a stronger password reduces the risk that it falls to password-guessing attacks.

If aiming to minimize expected loss L, the solution occurs when L has derivative zero with respect to $E = \sum_i E_i$, which from (3) gives the system of equations

$$\frac{dP_i}{dE_i} = 0 \quad \text{for} \quad i = 1, 2, 3, \cdots, N. \tag{4}$$

The solution is trivial: the optimum is achieved when further effort can't reduce the probability of loss for any of the N accounts. Thus to minimize L, we should increase each E_i until no further reduction is possible (further effort does not affect P_i). If $P_i(E_i)$ is monotonically decreasing—so further effort always reduces P_i—then expected loss is minimized at infinite effort. Thus *the lack of a constraint on effort leads to an unrealistic solution.* Note also that users gravitate toward a solution very different from this effort-maximizing one. From this we infer: *the objective function users minimize is not merely expected loss*—if it were they'd always invest effort that could reduce loss and always follow A1, A2.

This optimization has an obvious flaw: minimizing L implicitly values user effort at zero. Users presumably care about loss due to account compromise, *but they also factor in the effort they must spend to reduce that loss.* They may be willing to spend effort to reduce loss, but at some point there are diminishing returns; and it is wasteful to continue after the cost of further effort exceeds expected reduction in loss. Thus, rather than an unconstrained optimization [29] ignoring reluctance to increase effort, we should solve a constrained problem explicitly including cost of user effort.

One way to incorporate a constraint is to minimize loss subject to a bound—e.g., say $\sum E_i < E_{max}$ to model users with an upper limit on the effort they are willing to exert. This is reminiscent of the compliance budget [5, 4]; it can be achieved with Lagrangian multipliers. A more general approach, which we follow, is to minimize not the expected loss, but the sum of effort plus loss, $L + E$. As a precedent for this approach, economics Nobel laureate Becker notes [6] that attempts to minimize crime lead to perverse results, and it is preferable to minimize the costs of crime plus the costs of detecting, prosecuting and punishing it. For example, it makes little sense to spend \$1 more on policing effort if that reduces the effects of crime by less than \$1.

To illustrate the importance of the objective function we revisit the question of finding optimum allocation of effort[2] when $dP_i/dE_j = 0, i \neq j$ (*i.e.*, no cross-account attacks). The optimum occurs when the derivative (with respect to E) of objective function $L + E$ is 0: $dL/dE + 1 = 0$. Using (3) to substitute for L gives the system

$$L_i \cdot \frac{dP_i}{dE_i} = -1 \quad \text{for } i = 1, 2, 3, \cdots, N. \tag{5}$$

Thus, at optimum effort allocation,[3] the marginal return on effort to reduce P_j is a factor L_i/L_j higher than that to reduce P_i :

$$\frac{dP_j}{dE_j} = \frac{L_i}{L_j} \cdot \frac{dP_i}{dE_i}. \tag{6}$$

If the loss for the most important account is, say, 10^4 times that for the least important ($L_1 = 10^4 L_N$) then the marginal return on effort should differ by that factor. Thus, *effort should not be spent equally on all accounts*.

While we are unlikely to find an exact form for how the probability of harm varies with effort, using a parametric form can help illustrate the relations. Basing an example on Shamir's quote "to halve your vulnerability you have to double your expenditure" [46], we examine what happens when there is a reciprocal relation between them, *i.e.*, $P_i(E_i) \propto 1/E_i$. This gives $dP(E_i)/dE_i \propto -1/E_i^2$. Substituting into (6) indicates how the relative effort for two accounts should depend on the relative losses:

$$E_j = \sqrt{\frac{L_j}{L_i}} \cdot E_i.$$

[2] This includes max cumulative effort and where/how to allocate it.

[3] This is also true at other points, though not proven here.

So if two accounts differ in value by factor 10^4, ideally the effort expended would differ by a factor 100. We reiterate: this analysis, as it depends on the parametric form for $P_i(E_i)$, is for illustrative purposes only.

Now contrast this solution minimizing $L+E$, with that found by minimizing L alone (system (4) above). First, if minimizing L, all passwords should be as strong as possible, meaning that (at the optimum) no additional effort can reduce the risk for any account. When minimizing $L+E$ this isn't the case: (5) says that (at the optimum) additional effort may still reduce risk for every account, but it is sub-optimal to spend it. Second, when minimizing L, the optimum protection given to an account is independent of L_i. When minimizing $L+E$ some accounts should be (possibly far) less protected than others: (5) shows that the rate of return on effort should be *inversely related to the account value*.

Thus, using objective function $L+E$ (not L) makes an enormous difference in solutions. We posit that much of the advice directed at users aims to minimize L only, and is ignored as users implicitly care about E also and have found operating points attempting to minimize *their* objective function; these points may or may not be optimal, but have been arrived at by *ad hoc* methods. We note that in minimizing $L+E$ we neglect the non-linear response to probabilities predicted by Prospect Theory [35]. We believe that the rational model which offers (Kahneman [34]) "great precision in some situations and good approximation in many others" is the most realistic one that we can currently make progress on, and significantly advances a model that neglects E. Finally, use of the term *portfolio* is not accidental. Since 1952 [40] it has been recognized that managing a portfolio of equities raises issues drastically different from managing individual securities. In an analogous situation for passwords, due to cross-account attacks, the security of accounts cannot be considered in isolation, yet the literature has given little attention to the portfolio problem.

5 Modeling Loss, Effort, Attack Classes

While (5) offers to guide effort allocation when minimizing $L+E$, it assumed $dP_i/dE_j = 0$ for $i \neq j$; we postponed issues of cross-account attacks. This might be reasonable if guessing were the only attack and account passwords were unique; the probability P_i of compromise of account i would then depend only on how passwords withstood attack. But that over-simplifies. With password re-use, compromise of one account can leak to others, and client-side malware affects all accounts. Such attack vectors are too important to ignore. P_i depends on effort not just devoted to account i but also, e.g., to address client malware or avoid phishing, and the security of other sites the password is re-used on.

If we can't assume partial derivatives of zero, then on minimizing $L+E$, instead of (5) we get the system

$$\sum_{i=1}^{N} \frac{\partial P_i}{\partial E_j} \cdot L_i = -1 \quad \text{for } j = 1,2,3,\cdots,N. \qquad (7)$$

This is not simply a linear system. The N unknowns E_j, specified implicitly by the constraint on N^2 partial derivatives, relate non-linearly to the L_i. The intuition of (5) is now lost. A simple interpretation (e.g., marginal return on effort should be inversely related to loss) is no longer discernible, as instead of appearing singly, the partial derivatives are now constrained by a sum.

Note that if we minimize L instead of $L+E$ we get a system similar to (7), but with zero on the right side. Since losses must be non-negative and the partial derivatives are non-positive, the solution is achieved by setting $\partial P_i/\partial E_j = 0$ for all i, j. This would again indicate optimality occurs when no further effort can reduce any of the loss probabilities. Thus, the fully general system is tractable if we use the wrong objective function. Alternatively, a simplified system (*i.e.*, assuming $\partial P_i/\partial E_j = 0$ for $i \neq j$) is tractable using a realistic objective function. However, the general system using the realistic objective function is challenging. Our way forward is to re-structure the problem to isolate types of attack affected by different types of effort. By including the major attack vectors, the model is necessarily more complicated than that yielding (5), but will allow insight on how to manage a portfolio when minimizing $L+E$.

5.1 Attack Classes and Attack Vectors

We partition attacks into three classes:

- **Class I attacks (*FULL*)**: these compromise all password-protected accounts of a user. They involve general attack vectors targeting the client machine. Upon success, the attacker acquires actual passwords. Example: client-side malware (e.g., persistent keyloggers), which we assume provides attacker access to all of a user's passwords.

- **Class II attacks (*GROUP*):** these compromise all of a user's accounts protected by the same shared ("group") password, with the attacker obtaining that password; this includes singleton groups. Examples: phishing, brute-force and other guessing, shoulder-surfing, server break-ins to obtain password files, network channel compromise. We assume the attacker will try appropriate credentials with this password on all relevant sites (a finite number), determining associated account userids from public information or otherwise, and gain access to all accounts that use this password. Com-

	Class I (Full, direct)	Class II (Group, direct)	Class III (Single, indirect)
Attack Vectors	Client-side malware (keyloggers, *etc.*)	Phishing, password guessing, shoulder-surfing, system-side database compromise, network channel compromise	Session hijacking, cross-site scripting, password reset mechanisms
Effort elements addressing attack	Run AV, disable unused apps/interfaces, run up-to-date software (apply patches), avoid suspicious links, don't click on email attachments	Choose strong passwords, don't re-use passwords, change passwords often, don't write down, avoid phishing sites	*little advice*

Table 1: Attack classes for password-protected accounts, attack vectors, and relevant user effort elements (defensive actions).

promising one account thus may imply losses in all same-password accounts of the user.

- **Class III attacks (*SINGLE*):** these compromise only a target account, without obtaining the actual password.[4] Example attack vectors: cookie stealing, single-session hijacking (e.g., by cross-site request forgery), exploiting password reset vectors (but not those that mail-back original passwords). The attacker may gain account access, but cannot leverage this to access other accounts, even same-password accounts.

While this classification is still a simplification—e.g., some passwords are easily derived from related passwords [49]—it allows us to model cross-contamination. To handle the case where passwords are modified-and-shared rather than simply shared between groups, as observed by Das etal [17], would require an adjustment to this model (e.g. by modifying Class II). Table 1 synopsizes the attack classes, their principal vectors and user effort that addresses them. Note that the user effort related to passwords (*e.g.*, strength, avoiding re-use, avoiding phishing sites) is concentrated in Class II. Class I deals with system-wide attacks. Class III deals with attacks affecting only a single account, not others sharing the same password.

The probability of individual account compromise can now be split as:

$$P_i \approx P^I + P_i^{II} + P_i^{III} \tag{8}$$

where superscripts denote attack class. Here, and throughout the paper, the compromise probabilities are assumed small enough that the well-known approximation $(1-\prod_i(1-P_i)) \approx \sum_i P_i$ can be used. For Class I we omit the subscript from attack probability P^I, since it has the same value for all accounts. Now, if a user has $G \leq N$ unique passwords, sharing password w_j across a set $\mathscr{A}_J$ of accounts, the expected loss becomes:

$$\begin{aligned} L &= P^I \sum_{i=1}^{N} L_i + \sum_{J=1}^{G} \Big(\sum_{i\in\mathscr{A}_J} P_i^{II} \Big) \Big(\sum_{i\in\mathscr{A}_J} L_i \Big) + \sum_{i=1}^{N} P_i^{III} L_i \\ &= P^I \sum_{i=1}^{N} L_i + \sum_{J=1}^{G} P_J \cdot L_J + \sum_{i=1}^{N} P_i^{III} L_i. \end{aligned} \tag{9}$$

To distinguish, e.g., account i from password-sharing group J, we abuse notation with upper-case indices; and similarly subscripts to denote sums over groups, so

$$L_J = \sum_{i\in\mathscr{A}_J} L_i \quad \text{and} \quad P_J = P_J^{II} = \sum_{i\in\mathscr{A}_J} P_i^{II}, \tag{10}$$

dropping P_J^{II}'s superscript as this is for Class II only.

The three terms on the right side of (9) match the three attack classes. The first term is the probability of a Class I attack, weighted by the entire portfolio value. The second term is the sum across the G password-sharing groups, each weighted by the value of the accounts in that group. This highlights the drawback of password re-use: a compromise is not isolated to one account, but spreads to others. The third term is the sum of probability of individual account compromise weighted by the account value.

5.2 Modeling Effort Allocation and Effectiveness

To minimize an objective function that includes both loss and effort, both must be mapped to the same dimension. For simplicity, we assign a monetary value E for the time and effort—a mapping that is naturally user-dependent. The cost of this management has different components; preventing different attacks often requires different mechanisms. Thus again, this is split based on the class of attack the effort addresses:

$$\begin{aligned} E &= E^I + E^{II} + E^{III} \\ &= E^I + \sum_{J=1}^{G} E_J^{II} + \sum_{i=1}^{N} E_i^{III}. \end{aligned} \tag{11}$$

Under the assumption that effort is applied independently across classes, from (11) we also have: $\partial E/\partial E^I =$

[4] In the case of password resets mentioned next, the attack may recover a new temporary reset password, but not the original password possibly shared across other accounts.

$\partial E/\partial E^{II} = \partial E/\partial E^{III} = 1$. E^I is the cost of defensive effort related to Class I attacks—including, e.g., the total cost and time/effort associated with purchasing/running anti-virus software, and all effort related to keeping a computer malware-free. E_J^{II} is the cost of effort involved in combating Class II attacks on a group that share the same password (brute force, social engineering, etc.). Clearly, $E_G(N)$ given in (2), the cost of managing the password portfolio, is a portion of E^{II}. However, E^{II} also includes effort devoted to other Class II attacks, such as phishing [33]. E^{III} relates to account-specific efforts, which may include, e.g., managing one-time passwords or second-factor authentication devices.

Assuming the three types of efforts can be controlled independently, objective $L+E$ is minimized when

$$\frac{\partial(L+E)}{\partial E^I} = \frac{\partial(L+E)}{\partial E^{II}} = \frac{\partial(L+E)}{\partial E^{III}} = 0 \qquad (12)$$

which simplifies to:

$$\frac{\partial L}{\partial E^I} = \frac{\partial L}{\partial E^{II}} = \frac{\partial L}{\partial E^{III}} = -1. \qquad (13)$$

Substituting our expression for loss (9) into each of these three equalities, the parade of equations concludes with:

$$\left(\sum_{i=1}^{N} L_i\right) \frac{\partial P^I}{\partial E^I} = -1 \qquad (14)$$

$$L_J \cdot \frac{\partial P_J}{\partial E_J} = -1, J = 1 \cdots G \qquad (15)$$

$$L_i \cdot \frac{\partial P_i^{III}}{\partial E_i^{III}} = -1, i = 1 \cdots N. \qquad (16)$$

Note that we have used the fact that the effort devoted to group J does not affect either the probability of loss for group K (*i.e.*, $\partial P_K^{II}/\partial E_J^{II} = 0$ when $K \neq J$) or the effort devoted there (*i.e.*, $\partial E_K^{II}/\partial E_J^{II} = 0$ for $K \neq J$).

5.3 Implications of the Model

Equations (14)-(16) help formalize the concept of optimization of defensive investment (i.e., effort) related to expected loss. We briefly discuss each further.

Class I equation. Eqn (14) relates to Class I attacks, e.g., client-end malware like keyloggers. It isolates the cost of avoiding such attacks from efforts directly related to password management. The sum over all L_i reflects the definition: Class I attacks compromise *all* of a user's passwords—thus the loss may be quite large, especially if the sum strongly dominates individual L_i values. The absence of individual P_i in (14) reflects that defensive effort (cost) related to reducing likelihood of Class I losses is unrelated to costs associated with managing individual passwords. This is notable as current password advice to end-users *is* predominantly related to managing individual passwords (e.g., choosing stronger, more complex passwords, not re-using across accounts), none of which is related to (14).

Common advice related to (14) includes (see Table 1): keeping software up-to-date with patches; using AV (anti-virus) protection; disabling unused applications and interfaces; "hardening" the platform OS.

Regarding overall investment in client-end protection, (14) informs us that effort expended defending Class I attacks should be driven by: (i) the total value of all accounts the user accesses from the client device—the larger this value, the more worthwhile even small defensive efforts which reduce the probability of losses; and (ii) the degree to which incremental defensive effort reduces the probability of Class I attacks. Note that, counter-intuitively, the effort optimally expended is *not* driven by the absolute probability of Class I attacks—since effort spent doesn't necessarily reduce the probability of successful attack, even if P_i is large.

Class II equation. Note that (15) is a set of equations, one for each password-sharing group J. L_J accumulates losses over the accounts sharing a password, based on the assumption that once a password is compromised, all accounts sharing it may suffer. P_J sum probabilities over all accounts in the group, for a similar reason.

Regarding overall investment in defenses against Class II attacks, (15) informs us that the allocation of such effort should be driven by the following, considered now *for each group:* (i) the total value of all accounts in the shared-password group—the larger this value, the more worthwhile defensive efforts which reduce the P_J; and (ii) the cumulative sum, across all groups accounts, of the degree to which incremental defensive effort reduces the probability of Class II attacks. As above for Class I, the optimal effort expended is *not* driven by the absolute probability of Class II attacks; the same is true for (16) and Class III.

The similarity between (15) and (5) should be obvious: we again have a constraint involving a single partial derivative. A few conclusions can be drawn that mirror those drawn about the simpler model in Section 4. First, all passwords should *not* be equally strong (that would be wasteful, allocating excessive effort to low-value account groups at the expense of high-value ones). Second, the rate of change of P_J with respect to effort should be inversely proportional to L_J. This means that (unless a user has excess capacity of effort they wish to spend, and no higher-value groups to spend it on) groups with $L_J \approx 0$ *should* be very exposed and *should* have weak passwords, since as $1/L_J \to \infty$, they should be at the point where $\partial P_J/\partial E_J$ is extremely high; thus even tiny invested effort would reduce P_J significantly, but spending effort there would be wasteful as we care not about

P_J but $P_J \cdot L_J$. Effort is better spent on an account group with high L_J (even if $\partial P_J/\partial E_J$ is very low). It makes no sense to invest at all on accounts where $L_J = 0$, so long as any other account has $L_J > 0$.

Toy example. To illustrate (15), suppose two bank accounts sharing a common password have loss values 10 and 12. Assume that the first account is phished, and thereafter an attacker tries the same password with appropriate obtained userid on all banks. Assume further that additional effort $\delta E = 3$ units (e.g., a stronger group password) reduces individual account compromise probabilities from 0.1 to 0.09 (first account) and from 0.05 to 0.03 (second). Then the initial expected loss (see (9)) of $(10+12)(0.1+0.05) = 3.3$ is reduced, by extra effort, to $(10+12)(0.09+0.03) = 2.64$. Thus extra effort of 3 units reduced loss by only 0.66. This can also be observed by looking at the differences (or derivatives, as in (15)); the change is $(10+12)(-0.01/3 - 0.02/3) = -0.22$. And, in this example, as -0.22 is less negative than -1, we have higher investment than optimal—the cost of effort invested exceeds the reduction in loss it provides. The equations thus confirm our expectations, despite the "units of measure" carrying little meaning.

Class III equation. Finally, (16) reminds us that, regardless of password policies, we must keep in mind and beware reset mechanisms and alternative access paths. Class III attacks involve only a single account and are unrelated to group sharing of passwords, being unrelated to the actual choice of passwords. As noted in Table 1, users get little advice related to Class III attacks (and hence $\partial P_i^{III}/\partial E_i^{III} \approx 0$). In the sequel, (16) is discussed little, as risks associated with these attacks are largely impervious to user effort, our present focus. Regarding overall effort defending Class III attacks, (16) tells us that, considering now *each account individually,* the allocation of such effort should depend on: (i) the account value; and (ii) the degree to which new effort reduces the probability of Class III attacks on it.

6 Account Grouping for Password Re-use

We saw in Section 3 that, without additional coping mechanisms, re-use is unavoidable for large N. We now show that it can help, even for smaller portfolios. Since we seek to minimize L+E there are two components to consider: changes in effort, and in expected loss. For loss, we need consider only Class II attacks, as Class I and III attacks are unaffected by re-use.

Consider the case of three accounts, two relatively low-value (L_1, L_2), one high-value (L_3) so $L_3/(L_1 + L_2) = m >> 1$. For simplicity assume further $P_1^{II} \approx P_2^{II}$ (we will drop superscripts II, as only Class II attacks are relevant). Now compare Case A (using three unique passwords) vs. Case B (re-use one password across low-value accounts, with unique password for high-value). For Case A, expected Class II losses are: $P_1L_1 + P_2L_2 + P_3L_3$. For Case B, re-use increases the expected loss over the first two accounts by $\Delta L = (P_1+P_2)(L_1+L_2) - (P_1L_1 + P_2L_2)$; as $P_1 = P_2$ now, this is $P_1L_2 + P_2L_1 = P_1(L_1+L_2) > 0$, but the user manages one fewer password. Assume the saved effort ΔE is used to strengthen the high-value password[5] reducing the expected loss related to the third account from P_3L_3 to $(P_3(1-e))L_3$ where $0 < e < 1$. So Case B is preferable (has lower expected loss) provided the increase ΔL in expected loss over the first two accounts is less than the expected decrease on the third, i.e., provided: $P_1(L_1+L_2) < eP_3L_3$, or equivalently,

$$m > P_1/(eP_3) \tag{17}$$

We expect (17) often holds—e.g., if $m = 50$ (a financial account with value 100 times that of a free or low-value subscription site) and $P_1 \approx P_3$, then (17) is true for $e > 1/50 = .02$, i.e., a 2% or greater reduction in probability of loss due to a strengthened password. The right side of (17) becomes even smaller if $P_3 > P_1$, and if $P_3 < P_1$ then (17) still holds for a correspondingly larger e. Thus certainly, re-use can be beneficial.

Of course, guessing is but one possible Class II attack; some others also increase the consequences of re-use. The risks of some, like phishing, can be reduced by the user, while that of others, like server-side attacks, are largely impervious to user effort (see 7.3).

6.1 Share among Accounts of Similar P/L

We now explore how to re-use passwords "properly". Based on the loss model, we give an optimal password re-use strategy in the following sense: for a fixed number of passwords, and a given set of accounts (thus effort is fixed), find how to group accounts to minimize total expected loss.

As before, assume a user splits N accounts into G groups each sharing a unique password. Per the second term on the right of (9), the total Class II loss is:

$$L^{II} = \sum_{J=1}^{G} (\sum_{i \in \mathscr{A}_J} P_i^{II})(\sum_{i \in \mathscr{A}_J} L_i) \tag{18}$$

Is there an optimal way to partition this set of accounts into shared-password groups $\mathscr{A}_J$?

We first address the case of adding a new account to an existing portfolio; i.e., we have G groups and must decide to which group a new account is best added. From (18), adding a new account with (P_i, L_i) to group J, the incremental loss is (with L_J, P_J as in 5.1):

$$\Delta L = P_iL_J + L_iP_J + P_iL_i \tag{19}$$

[5] If users do not do this, the case for re-use is lost; this is critical.

From (2), the incremental effort is $\Delta E \approx \lg G$. Since neither ΔE, nor the third term of ΔL depend on the group J, the objective function, $L+E$, depends on the group assignment only through the first two terms of (19). Thus the new account should be added to the group $\mathscr{A}_J$ minimizing $P_i L_J + L_i P_J$. This brings an interesting insight: if any group J exists such that $P_J < P_K$ and $L_J < L_K$ for all G (i.e., the group has both a smaller total probability and a smaller total loss than all other groups), then all new accounts should be added to that group J, until one of the two inequalities fails.

Thus without loss of generality, the remaining case is in deciding between two groups $\mathscr{A}_J$, $\mathscr{A}_K$ when $P_J < P_K$ and $L_J > L_K$. Here, new account i should be assigned to $\mathscr{A}_J$ (vs. $\mathscr{A}_K$) if and only if:

$$P_i L_J + P_J L_i \leq P_i L_K + P_K L_i \tag{20}$$

This can be rewritten as

$$\frac{L_i}{P_i} \geq \frac{L_J - L_K}{P_K - P_J} \tag{21}$$

Fig.2 illustrates this constraint graphically. Recall that a line of slope m in the PL plane is given by $L = m \cdot P + c$. Thus, (21) says that account i should be placed in group $\mathscr{A}_J$ (vs. $\mathscr{A}_K$) if and only if point (P_i, L_i) lies above a line with slope $(L_J - L_K)/(P_K - P_J)$ going through the origin. Fig.2 shows the construction of a (solid red) line with slope $(L_J - L_K)/(P_K - P_J)$; it passes through points (P_K, L_J), (P_J, L_K). The dashed red line is one of the same slope, but through the origin.

In summary, the decision boundary between adjacent groups $\mathscr{A}_J$ and $\mathscr{A}_K$ is given by the line:

$$L = \left(\frac{L_J - L_K}{P_K - P_J} \right) \cdot P. \tag{22}$$

A necessary condition for optimality is the absence of *profitable single moves* in the following sense: if a partitioning of accounts is optimal, the total loss cannot be decreased by moving any account i from group $\mathscr{A}_J$ to any other group $\mathscr{A}_K$. This can be expressed as

$$P_i L_{J^*} + P_{J^*} L_i \;\; \leq \;\; P_i L_K + P_K L_i \quad \text{for all } K. \tag{23}$$

Here (P_K, P_{J^*}) are the total loss probabilities, and (L_K, L_{J^*}) the total losses, resp., for groups K and J^* where J^* denotes group J after removing account i. Similar to (21), we can rewrite (23) as

$$\frac{L_i}{P_i} \geq \frac{L_{J^*} - L_K}{P_K - P_{J^*}} \quad \text{for all } K. \tag{24}$$

Consider the case when the number of accounts N becomes large, in which case P_i and L_i are typically small relative to P and L. We can then assume the total loss and probability of each group does not change much by adding or removing a single account. Thus $P_{J^*} \approx P_J$ (i.e., $P_J \approx (P_J + P_i)$).

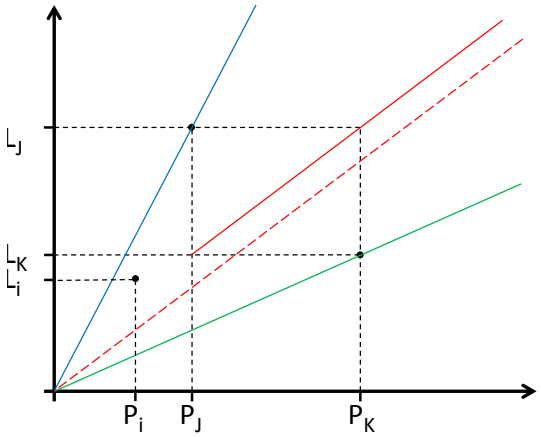

Figure 2: Optimal assignment of a new account (P_i, L_i) between two groups J and K. If the new account falls above the dashed red line, total loss will be smaller when the account is assigned to group J.

We first show that given an optimal grouping, for any groups J and K the decision boundary is bounded by:

$$\frac{L_K}{P_K} \leq \frac{L_J - L_K}{P_K - P_J} \leq \frac{L_J}{P_J}. \tag{25}$$

The decision boundary slope (Fig.2, dashed red line) thus must be between that of the green and blue lines.

To show this, note that group K must contain at least one account i with $L_i/P_i \geq L_K/P_K$ (since all of the L_i and P_i are ≥ 0). Thus (21) holds, implying account i belongs in group $\mathscr{A}_J$ rather than $\mathscr{A}_K$ unless the righthand inequality of (25) holds. The reverse argument applies to show the lefthand inequality in (25).

Now (22) tells us that the decision boundaries are lines through the origin; so each group has at most two neighbors. Further, (25) when applied to every pair of "adjacent" groups in the PL plane, implies the same ordering applies to not only the ratio of L and P differences as in (22), but also the ratio of their values:

$$\frac{L_1}{P_1} \geq \frac{L_2}{P_2} \geq \cdots \geq \frac{L_G}{P_G} \tag{26}$$

where, without loss of generality, the groups have been ordered clockwise, according to their order in the PL plane. In general for groups $\mathscr{A}_J$, $\mathscr{A}_K$, recall that $P_J < P_K$ implies $L_J > L_K$. From this it follows that, given an ordering for the ratio, the same ordering must apply to the expected loss and the reverse ordering for probability, i.e.,

$$P_1 \leq \cdots \leq P_G \quad \text{and} \quad L_1 \geq \cdots \geq L_G. \tag{27}$$

Thus ordering the account groups by decreasing total loss, they have increasing total probability; due to the possibility of equality, none of the orderings is strict.

6.2 Groups Similarly Weighted by PL

Consider next how large and how disparate different groups will be. We show that under certain conditions, the groups formed have similar individual products PL. With focus again on the outcome as G increases, from Section 6.1 the groups obey an ordering in terms of P, L, and L/P, and the decision line slope (dashed red line in Fig.2) must be between the slopes of the two adjacent groups. Thus, assuming accounts exist around every point in the PL plane, as G increases the adjacent groups have increasingly similar slopes, with bounds on the decision boundary slope per (25). Since, from (27), the L_i are non-increasing, and from (27), the P_i are non-decreasing, we have $L_J \geq (L_J + L_K)/2 \geq L_K$ and $P_J \leq (P_J + P_K)/2 \leq P_K$. It follows from (25) that, as G increases:

$$\frac{L_J - L_K}{P_K - P_J} \approx \frac{(L_J + L_K)/2}{(P_J + P_K)/2}. \tag{28}$$

Re-arranging yields:

$$(L_J - L_K)(P_J + P_K) \approx -(P_J - P_K)(L_J + L_K) \tag{29}$$

Expanding products and eliminating common terms,

$$P_J L_J \approx P_K L_K \tag{30}$$

Thus the product of probability and loss for adjacent groups is about equal, increasingly so as the numbers of groups G and accounts per group increase.

6.3 Pedagogical Illustration through Two Generated Datasets

To illustrate, we generated two datasets, assigning accounts to groups with an optimization program obeying the "no profitable moves" rule. The simulation models 100 accounts, with randomly assigned P_i and L_i, to be divided in five groups (shown by different colors in the figures). The program assigns accounts to a group one at a time. After each assignment, it tests all possible single moves and swaps, performing any profitable moves before moving on to assign the next account. In the first dataset, corresponding to Fig.3 and Table 2, P_i and L_i are independently drawn from uniform distributions.

In practice, the combination of (23), (25), and (30) means that whenever passwords are to be re-used across accounts, the optimum strategy is to do so across accounts with similar P/L ratio, and add enough accounts per group to achieve similar total PL products for each group. The resulting account assignments split the PL plane into slices (see Fig.3). This implies that most high-value accounts end up in the same group (particularly if they have low compromise probability), and most low-value accounts end up in another group (particularly if they have high compromise probability)—apparently in line with what many users currently do. Table 2 reports selected characteristics of the 5 password groups; note the similar values of PL across groups, strictly decreasing L, and strictly increasing P and P/L.

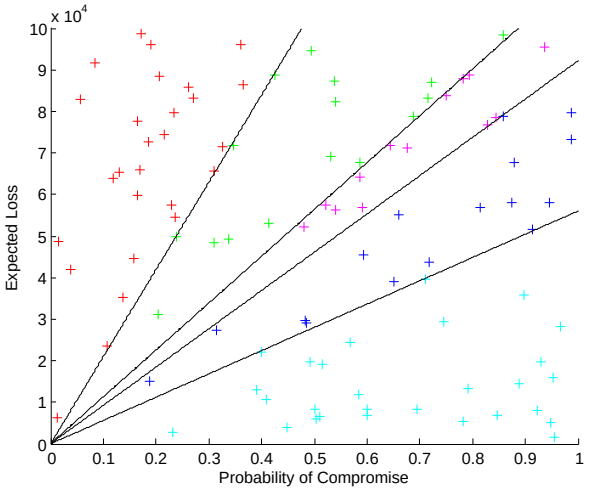

Figure 3: Password grouping, under the "no profitable moves" strategy. For this example, 100 accounts are uniformly placed at random in the PL plane, and optimally assigned to one of 5 groups. Note the linear decision boundaries, corresponding to P/L ranges (slices).

While the dataset used to produce Fig.3 allows visualization of the linear decision boundaries, such a dataset with independent distribution over P and L is not what we would expect in practice. We thus generated a second dataset (see Fig.4 and Table 3) where L_i follows a power law distribution and the expected value of P_i is inversely proportional to (the square of) L_i. While all observations on the previous dataset still hold, further insights are evident. As on this dataset high-value accounts are less likely to have high P_i, the high-value accounts end up grouped together. Indeed, group 1 includes 53 accounts, more than half of the set, while group 5 has only 4 accounts (see Table 3).

The total resulting loss across all five groups is 7.94×10^4. To see how this optimal assignment compares to a random assignment, we computed total loss on the same dataset on randomly assigning accounts to the 5 groups (in 100,000 Monte Carlo trials), finding an average PL of 1.16×10^8 (std deviation 0.24×10^8). Thus the optimal loss was 1500 times smaller than by random assignment, and 5 standard deviations below the mean.

We emphasize that both datasets are modelled examples to illustrate principles. For other datasets, the general findings will hold, but actual construction of groups may significantly differ depending on the data.

Group #	P	L	PL	P/L	max P/L	min P/L	Group Size
1	1.88e+01	3.96e+05	7.44e+06	4.75e-05	6.09e-04	1.80e-05	28
2	1.14e+01	8.08e+05	9.17e+06	1.40e-05	1.77e-05	1.09e-05	16
3	9.35e+00	9.71e+05	9.08e+06	9.63e-06	1.08e-05	8.93e-06	13
4	7.94e+00	1.14e+06	9.06e+06	6.96e-06	8.72e-06	4.79e-06	16
5	4.91e+00	1.82e+06	8.93e+06	2.70e-06	4.73e-06	3.22e-07	27

Table 2: Characteristics of each group in the grouping corresponding to Figure 3.

Group #	P	L	PL	P/L	max P/L	min P/L	Group Size
1	2.29e+01	3.24e+02	7.41e+03	7.06e-02	9.46e+02	1.44e-03	53
2	6.70e-01	1.76e+04	1.18e+04	3.80e-05	1.24e-03	4.45e-06	24
3	6.42e-02	2.40e+05	1.54e+04	2.68e-07	1.66e-06	3.01e-08	11
4	7.04e-03	2.45e+06	1.72e+04	2.88e-09	1.66e-08	4.09e-10	8
5	1.26e-03	2.19e+07	2.77e+04	5.76e-11	2.80e-10	9.29e-12	4

Table 3: Characteristics of each group in the grouping corresponding to Figure 4.

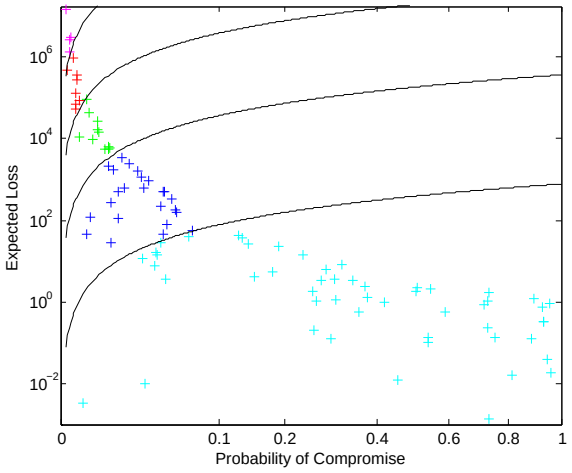

Figure 4: Password grouping, second dataset (P_i's drawn from a distribution with mean inversely proportional to L_i^2). Due to the non-linear axis, the decision boundaries are no longer linear. The number of accounts in each group differs from Fig.3.

7 Special Cases

Next, special cases illustrate how the model addresses additional assumptions and circumstances.

7.1 Case 1: Unknown P_i (Modeled as Equal)

The model highlights two variables with large effect on the problem: loss and compromise probability. Most users could give some estimate of loss that would result from compromise of a specified account—perhaps not entirely accurate, but representative of expected loss, even if only in relative terms. In contrast, user estimates of probabilities would likely be far worse, perhaps without sense of even relative P_i's. We thus consider here what results from the optimization model on assuming equal probabilities $p = P_i$ for all i. Slice-based partitioning still applies, as does the ordering—the latter now easier with all accounts on a vertical line in the P_iL_i plane. The main question is how many accounts will each group have, and how does that relate to the L_i of accounts in each group.

If group J has N_J accounts, write $P_J = pN_J$. Then (30) yields $(pN_J)(L_J) \approx (pN_K)(L_K)$, or, equivalently:

$$\frac{N_J}{N_K} \approx \sqrt{\frac{L_K/N_K}{L_J/N_J}}. \tag{31}$$

Thus groups with high-value accounts will have fewer accounts; optimally, the number of accounts N_J in a group J varies inversely with the square root of the average loss L_J/N_J in that group.

To illustrate, we re-run the optimization process on the second dataset (see Section 6.3), but now assuming ignorance of individual probabilities, modeling equal P_i. The principles discussed earlier now result in the accounts being split by strict ordering of losses. The number of accounts in the 5 groups is now (82, 11 4, 2, 1), vs. (53, 24, 11, 8, 4) in Table 3. As might be expected, total losses increase to 1.24×10^6, vs. 7.94×10^4 for optimization using known probabilities. This is 16× higher than the optimum, but still 93× smaller than the average loss from random assignment (see Section 6.3).

7.2 Case 2: Group Passwords of Unequal Strength

We showed in Section 5.3 that passwords should *not* have the same strength. We now show how the assumption made in Section 6 (that P_i did not change much when we moved account i from password group J to group K) can be relaxed, so that there is no incompatibility. Here we

briefly analyze the impact, on optimization results, when P_i is password-dependent and groups have passwords of different strength. Denote the (now group-dependent) compromise probabilities $P_{i\in J}$, $P_{i\in K}$. Then by the argument used in (20), account i should be assigned to $\mathscr{A}_J$ if and only if

$$P_{i\in J}L_J + P_J L_i \leq P_{i\in K}L_K + P_K L_i. \tag{32}$$

We again seek a bounding condition on L_i/P_i, but now using what group-neutral P_i value? We use the geometric average $P_i = \sqrt{P_{i\in J}P_{i\in K}}$ and define the squareroot ratio $r = \sqrt{P_{i\in J}/P_{i\in K}}$. Then (32) yields

$$\frac{L_i}{P_i} \geq \frac{rL_J - (1/r)L_K}{P_K - P_J}. \tag{33}$$

Note $r>1$ if group J has password weaker than K. Thus with respect to group assignment, a weaker group J password has an effect equivalent to scaling up group losses L_J, making it harder to satisfy the condition for assignment to group J. Other results regarding the slicing, P and L ordering, and so on remain as before.

7.3 Case 3: Unequal Server Break-in Probabilities

Finally, consider the effects of different levels of security at the server. The probability of server break-in is largely outside users' control, but the consequences are not: a user may decide to share a password across accounts, only to have one of the servers leak her password, compromising all accounts sharing it. While the previous analysis already takes into consideration server break-in (as a Class II attack), we now analyze how two sites with different server break-in probabilities will affect the optimum allocation.

Consider two accounts i and j, with same values $L_i = L_j$ but different probabilities, $P_i = P_j + \delta_i$, where δ_i is the added break-in probability due to a site i server poorly managed compared to j. Upon assigning account i (poorly managed) to a group, the added probability δ_i will imply a higher ratio P_i/L_i, so the account will (likely) be grouped with accounts with higher P/L, typically lower-value accounts. Furthermore, as discussed in Section 5.3, these groups may have a weaker password. Thus, for a server with higher break-in probability, optimum password grouping seems to push towards grouping the related account with lower-value accounts.

Related to this, our criteria for optimality depend on how loss probabilities change with respect to effort, but not on the magnitudes of the probabilities themselves. Consider the possible case of a threat unaddressable by user effort, swamping all others. Let $P_i = P_{i,u} + P_{i,\bar{u}}$, where $dP_{i,\bar{u}}/dE = 0$. If $P_{i,\bar{u}} > 10^3 P_{i,u}$, it may be fruitless to spend substantial user effort if such expenditure affects only the third decimal place in P_i. Nonetheless, this is what our criterion for optimality suggests. System-side or back-end (server) risks may swamp risks under user control; we simply do not know.

7.4 Case 4: Coping Alternatives including Password Managers

Despite violating long-standing password guidance, writing passwords down is, if properly done, increasingly accepted as a coping mechanism. Other strategies to cope with the human impossibility of using strong passwords everywhere without re-use include single-sign-on, use of email-based password reset mechanisms, and password managers. Such "password concentrators", a form of password re-use, allow access to many accounts from one master access point, with account passwords stored either locally or in the cloud. While not explored in detail here, each can be analyzed in our framework; we illustrate for password managers.

The main threats (recall Table 1) when re-use is employed are client-side malware (all accounts fall), and various Class II attacks such as guessing, phishing, sniffing wireless links and server breaches (all accounts in the same sharing group fall). We must modify this picture slightly if a password manager is used. For Case A (password store on a user's local machine), the main risk is still Class I attacks like client-side malware. There is a decreased risk of phishing presumably, as users remember fewer individual passwords; similarly for guessing attacks, as arbitrarily strong passwords now require no user effort, and the master password that unlocks the store resides on the client. A server-side breach compromises only a single account. Thus, a password manager with client-side store approximates our model with $G = N$. The cost, of course, is that portability across different client devices is lost as the passwords (if they are unique and random) are effectively anchored to the client on which they are stored.

Consider next (Case B) a cloud-based store, protected by a single password. Phishing and guessing attacks against any system-assigned secrets at the end-servers remain unchanged. Now however, additional guessing, phishing and server breach attacks exist against the single master password which can result in the compromise of all accounts. Class I attacks (e.g. due to malware on the client) are unchanged. A password manager with a password-protected cloud-based store approximates our system with $G = 1$. It trades one set of risks for another: the use of random and unique passwords in such a system reduces both the risks related to any single manager-chosen password being stolen and those related to re-use in the face of server compromise. However, it introduces

severe new risks: if the master password is guessed or used on any malware-infected client, or the cloud store is compromised, then all credentials are lost.

8 Discussion and Implications

Recapping, recall first the task of end-users: to choose passwords random and strong (entropy $\lg S$ bits) without re-use. The effort to manage N such passwords without re-use is modelled as $N \lg S + \lg(N!)$; as portfolio size increases, this overwhelms user capability.

M1: *Remembering random and unique passwords is infeasible for other than very small portfolios.*

Users coping strategies include weak passwords and re-use. There is a large disconnect: what standard advice mandates as essential turns out to be impossible. We suggest this is due to a failure to explicitly include user effort in the objective function. Seeking to minimize loss alone leads to unrealistic effort-maximizing solutions. While some recent work [5, 4, 29] criticizes the practice of ignoring the burden password advice places on users, it has not to our knowledge been included directly in the objective function. We make a related observation:

M2: *While advice typically minimizes L over a single or small set of sites, user best interest is to minimize $L+E$ over an entire portfolio.*

The diversity of attacks complicates our search for an optimum effort allocation. Short-cuts are tempting; we can minimize $L+E$ while ignoring cross-account attacks (as in Section 4), or consider all attack types and minimize L alone. The first scopes the problem too narrowly, the second leads to the unrealistic demand to invest unbounded effort. While both yield "solutions" that are simpler than the model in Section 5, our work suggests that realistic analysis must address a realistic attack model *and* a realistic objective function.

M3: *Realistic analysis of password effort allocation requires incorporating attack vectors affecting 1) all accounts; 2) accounts sharing a password; and 3) single accounts.*

Our segmentation of the space into Class I, II and III attacks yields interesting insights. Minimizing $L+E$ over a portfolio implies user effort be spent unequally across accounts. As can be seen from (15), all passwords should not be equally strong; equal spending overspends on low-value, and underspends on high-value accounts (or account groups). Recall that, from (27), there is an ordering of the group values L_J; the largest may be many times greater than the smallest ($L_1 \gg L_G$). Any group for which $L_J \approx 0$ should have $\partial P_J/\partial E_J$ high (meaning a weak password). If we again invoke the reciprocal relation between P_J and E_J suggested in Section 4, we'd again find $E_1 = \sqrt{L_1/L_G} \cdot E_G$. Thus a $10^4\times$ value difference between the most and least valuable groups would imply a $100\times$ difference in invested effort. In this sense, not only are weak passwords understandable and allowable, but *their absence* would be sub-optimal:

M4: *A password portfolio strategy that rules out weak passwords is sub-optimal.*

Next, while sharing a password across a group of accounts can amplify consequences if it is compromised, we find it is sub-optimal *not* to re-use. First, (1) indicates re-use becomes unavoidable when N is large. Second, (2) and Fig.1 demonstrate the tradeoff involved even if N is small enough that re-use is theoretically avoidable; i.e., re-use increases the probability of loss from certain attacks, but also reduces effort. The question then is not whether re-use is good or bad, but whether the effort required to avoid re-use can be better spent on other attack types. Section 6 gives an example.

M5: *A password portfolio strategy that rules out password re-use is sub-optimal.*

The optimal strategy places accounts with similar P/L ratio in groups sharing a password. Enough accounts are added to each group to achieve similar PL products per group. Most high-value accounts (particularly if they have low P_i) end up in the same group(s), and most low-value accounts (particularly if they have high P_i) in another group(s).

M6: *Optimal password grouping tends to (i) group together accounts with high value and low probability of compromise; and (ii) group together accounts of low value and high compromise probability.*

The above observation lines up well with anecdotal accounts of what many users actually do. Our findings also agree with the informal claim [29], that users' actual effort allocation represents an efficient operating point. Thus, actual user password-related behavior is closer to optimal than current expert advice.

Password managers (cf. Section 7.4) may improve usability and reduce some risks, but remain vulnerable to Class I attacks (e.g., client-side malware). Managers that store passwords only on the client improve resistance to Class II attacks, since they can choose better passwords and eliminate re-use. However, in storing only on the client this gives up one of the major advantages of passwords, i.e. portability. Managers that store passwords in the cloud remove this restriction, but introduce a new system-wide attack: as before if the client is infected with malware all accounts are compromised, but now this

happens also if the cloud store is breached or the master password is stolen or guessed. Thus, cloud-storage managers trade one type of vulnerability for another.

M7: *Password managers using client-only storage allow a portfolio with random passwords and no re-use, but lose cross-client portability. However, if cloud storage is used it resembles a portfolio with only one group, since a new attack on either the master password or the store itself threatens all accounts.*

Another disconnect stems from many password-related threats being unrelated to the standard advice on maintaining a portfolio: Class I attacks, server breaches and Class III attacks are not reduced by password advice staples such as A1 and A2. Since successful Class I attacks sum the losses across all accounts, the advice to protect against them is disappointingly vague, while advice to protect against the less consequential Class II attacks is far more detailed and effort-consuming. It appears that users are given the advice that is most easily given, rather than the advice that would have greatest impact. Comparing (14) and (15) shows that at optimality the marginal return on effort spent on Class I attacks should be lower than that for any Class II group (*e.g.*, effort should not be wasted strenghtening passwords for a group with low L_J if any effective Class I measure remains undone). Greater focus is needed to explore which advice, for example from Table 1, provides protection against which attack vectors:

M8: *We lack metrics for the cost to end-users, of following standard advice, and the effectiveness of following it on reducing overall expected loss.*

An important outcome of our review is that, when minimizing $L+E$, optimality depends on the losses L_i, and on how the probability of loss varies with respect to effort $\partial P_i/\partial E_i$. In contrast if one minimizes L, the solution depends on neither. Without better knowledge of real-world values for L, and especially $\partial P_i/\partial E_i$, we are unlikely to achieve optimal resource allocation in practice. Conventional user behavior appears to be based almost exclusively on L, which users may be able to estimate; $\partial P_i/\partial E_i$ values are almost entirely overlooked. This points to an important research direction: while recent work has greatly improved understanding of password guessing resistance [8], we are almost entirely ignorant on how this evolves with effort.

M9: *Without better estimates of how loss probability changes with effort, we should not expect to be able to allocate effort (even close to) optimally.*

Finally, can concrete advice for users be distilled from our findings? For example, absent knowing how P_i change as a function of various types of effort, we lack a prescriptive way to determine the optimal number of groups G. Nonetheless, the knee of the curves in Fig.1, and what we know of user behavior [24, 21, 14] points to the number of groups being below 10 if no other aids are used. The values of loss probabilities P_i are entirely unknown; expected loss values L_i, or at least relative importance, are more easily estimated or ordered. Thus the variables needed to find an optimal grouping are (and are likely to remain) unavailable to most users. We might however simplify, e.g., assuming all P_i equal, or that P_i values differ by an order of magnitude between heuristically-defined categories (*e.g.*, banks, merchants, throwaway accounts, *etc.*).

While the optimal strategy involves selective re-use and weaker passwords, benefits accrue only if the effort saved is re-deployed elsewhere for better returns. Users must not arbitrarily weaken and re-use passwords. Thus empirical studies are needed to determine if our guidelines can be followed by users.

We hesitate to give definitive advice. First, this requires more insight than our current understanding of L_J and $\partial P_J/\partial E_J$ values allows. Second, we are reminded how far bad assumptions (e.g., minimizing L vs. $L+E$) can lead us astray. Consider, however, a strategy that chooses G in the range 5 to 10, and assigns accounts to groups by value so that the number of accounts in a group is as in Section 7.1. Given the uncertainty about unknown parameters, a strategy like this may be the best we have—and may even be optimal.

9 Concluding Remarks

We have explored the task of managing a portfolio of passwords. A starting point for our analysis was the critical observation that to be realistic, efficient password management should consider a realistic suite of attacks and minimize the sum of expected loss and user effort. Our model yields detailed results; it indicates that any strategy that rules out weak passwords or re-use will be sub-optimal. We have shown that optimality requires forming groups whose accounts in sum have similar PL values ($P=\sum P_i, L=\sum L_i$). This suggests simple guidelines, such as: if P_i is similar across accounts, then optimal grouping will put high-value accounts in smaller (or singleton) groups, and low-value accounts in larger groups. Our findings are consistent with certain user behaviors (e.g., [47]) that contradict accepted advice, offering to justify the behavior and giving evidence for the model's utility. We find that optimally, marginal return on effort is inversely proportional to account values. We note that while password re-use must be part of an optimal portfolio strategy, it is no panacea. Far from optimal outcomes will result if accounts are

grouped arbitrarily.

Acknowledgements. We thank Robert Biddle, Joseph Bonneau, and anonymous referees for their comments which helped improve this paper. The third author acknowledges an NSERC Discovery Grant and Canada Research Chair in Authentication and Computer Security.

References

[1] Stop.Think.Connect. http://www.stopthinkconnect.org/.

[2] US-Cyber Emergency Response Readiness Team: CyberSecurity Tips. http://www.us-cert.gov/cas/tips/.

[3] A. Adams and M. A. Sasse. Users are not the enemy. C.ACM, pages 40–46, December 1999.

[4] A. Beautement and A. Sasse. The economics of user effort in information security. Computer Fraud & Security, pages 8–12, October 2009.

[5] A. Beautement, M. Sasse, and M. Wonham. The Compliance Budget: Managing Security Behaviour in Organisations. In NSPW, 2008.

[6] G. S. Becker. Crime and punishment: An economic approach. In Essays in the Economics of Crime and Punishment, pages 1–54. UMI, 1974.

[7] J. Bonneau. Guessing human-chosen secrets. University of Cambridge. Ph.D. thesis, May 2012.

[8] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In Proc. IEEE Symp. on Security and Privacy, pages 538–552, 2012.

[9] J. Bonneau and S. Preibusch. The password thicket: Technical and market failures in human authentication on the web. In WEIS, 2010.

[10] J. Bonneau and S. Schechter. Towards reliable storage of 56-bit secrets in human memory. In Proc. USENIX Security, 2014.

[11] A. Brown, E. Bracken, S. Zoccoli, and K. Douglas. Generating and remembering passwords. Applied Cognitive Psychology, 18(6):641–651, 2004.

[12] W. Burr, D. F. Dodson, and W. Polk. Electronic Authentication Guideline. In NIST Special Pub 800-63, 2006.

[13] W. Cheswick. Rethinking passwords. USENIX LISA, 2010. http://www.usenix.org/event/lisa10/tech/slides/cheswick.pdf.

[14] W. Cheswick. Rethinking passwords. ACM Queue, 10(12):50–56, 2012.

[15] W. Cheswick, S. Bellovin, and A. Rubin. Firewalls and Internet Security, 2/e. Addison-Wesley, 2003.

[16] S. Chiasson, A. Forget, E. Stobert, P. C. van Oorschot, and R. Biddle. Multiple password interference in text passwords and click-based graphical passwords. In Proc. ACM CCS, 2009.

[17] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The tangled web of password reuse. NDSS, 2014.

[18] R. Dhamija and A. Perrig. Deja vu: a user study using images for authentication. In USENIX Security, 2000.

[19] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley. Does my password go up to eleven? the impact of password meters on password selection. In Proc. CHI, 2013.

[20] D. Florêncio and C. Herley. Where Do All the Attacks Go? Proc. WEIS, 2011, Fairfax, VA.

[21] D. Florêncio and C. Herley. A Large-Scale Study of Web Password Habits. Proc. WWW, 2007.

[22] D. Florêncio and C. Herley. Where Do Security Policies Come From? Proc. SOUPS, 2010.

[23] D. Florêncio, C. Herley, and P. van Oorschot. An Administrator's Guide to Internet Password Research. In Proc. USENIX LISA, 2014.

[24] S. Gaw and E. Felten. Password Management Strategies for Online Accounts. In ACM SOUPS, 2006.

[25] M. Gouda, A. Liu, L. Leung, and M. Alam. Single password, multiple accounts. In ACNS (Industry Track), 2005.

[26] B. Grawemeyer and H. Johnson. Using and managing multiple passwords: A week to a view. Interacting with Computers, 23(3):256–267, 2011.

[27] E. Grosse and M. Upadhyay. Authentication at scale. IEEE Security & Privacy, 11(1):15–22, 2013.

[28] E. Hayashi and J. Hong. A diary study of password usage in daily life. In CHI (note), pages 2627–2630, 2011.

[29] C. Herley. So Long, And No Thanks for the Externalities: Rational Rejection of Security Advice by Users. Proc. NSPW, 2009.

[30] Imperva. Consumer Password Worst Practices. 2010. http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf.

[31] P. Inglesant and M. A. Sasse. The true cost of unusable password policies: Password use in the wild. In CHI, 2010.

[32] B. Ives, K. Walsh, and H. Schneider. The Domino Effect of Password Re-use. C. ACM, 47(4):75–78, 2004.

[33] M. Jakobsson and S. Myers. Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft. Wiley, 2006.

[34] D. Kahneman. Thinking, fast and slow. Macmillan, 2011.

[35] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. Econometrica: Journal of the Econometric Society, pages 263–291, 1979.

[36] A. Karp. Forum (comment). C. ACM, 47(6):11–12, 2004.

[37] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In Proc. IEEE Symp. on Security and Privacy, 2012.

[38] R. Lemos. Yahoo breach highlights password reuse threat. eWeek. July 7, 2012.

[39] J. Ma, W. Yang, M. Luo, and N. Li. A study of probabilistic password models. Proc. IEEE Symp. on Security and Privacy, 2014.

[40] H. Markowitz. Portfolio selection. The Journal of Finance, 7(1):77–91, 1952.

[41] R. Nithyanand and R. Johnson. The password allocation problem. In WPES, 2013. Nov. 4, 6 pages.

[42] G. Notoatmodjo. Exploring the 'weakest link': A study of personal password security. C.S. Dept., University of Auckland, 2007. M.Sc. thesis.

[43] S. Preibusch and J. Bonneau. The password game: negative externalities from weak password practices. In Decision and Game Theory for Security, pages 192–207. Springer Berlin Heidelberg, 2010.

[44] S. Riley. Password security: what users know and what they actually do. Usability News, 8(1), 2006.

[45] M. Sasse, S. Brostoff, and D. Weirich. Transforming the "weakest link": a human-computer interaction approach to usable and effective security. BT Tech. J., 19(3):122–131, 2001.

[46] A. Shamir. 2002 Turing Award Lecture. http://amturing.acm.org/vp/shamir_2327856.cfm.

[47] E. Stobert and R. Biddle. The password life cycle: user behaviour in managing passwords. In Proc. SOUPS, 2014.

[48] M. Weir, S. Aggarwal, M. Collins, and H. Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. In Proc. ACM CCS, 2010.

[49] Y. Zhang, F. Monrose, and M. K. Reiter. The security of modern password expiration: An algorithmic framework and empirical analysis. In Proc. ACM CCS, 2010.