2012 IEEE Symposium on Security and Privacy

The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes*

Joseph Bonneau
University of Cambridge
Cambridge, UK
jcb82@cl.cam.ac.uk

Cormac Herley
Microsoft Research
Redmond, WA, USA
cormac@microsoft.com

Paul C. van Oorschot
Carleton University
Ottawa, ON, Canada
paulv@scs.carleton.ca

Frank Stajano†
University of Cambridge
Cambridge, UK
frank.stajano@cl.cam.ac.uk

***Abstract*—We evaluate two decades of proposals to replace text passwords for general-purpose user authentication on the web using a broad set of twenty-five usability, deployability and security benefits that an ideal scheme might provide. The scope of proposals we survey is also extensive, including password management software, federated login protocols, graphical password schemes, cognitive authentication schemes, one-time passwords, hardware tokens, phone-aided schemes and biometrics. Our comprehensive approach leads to key insights about the difficulty of replacing passwords. Not only does no known scheme come close to providing all desired benefits: none even retains the full set of benefits that legacy passwords already provide. In particular, there is a wide range from schemes offering minor security benefits beyond legacy passwords, to those offering significant security benefits in return for being more costly to deploy or more difficult to use. We conclude that many academic proposals have failed to gain traction because researchers rarely consider a sufficiently wide range of real-world constraints. Beyond our analysis of current schemes, our framework provides an evaluation methodology and benchmark for future web authentication proposals.**

***Keywords*-authentication; computer security; human computer interaction; security and usability; deployability; economics; software engineering.**

I. Introduction

The continued domination of passwords over all other methods of end-user authentication is a major embarrassment to security researchers. As web technology moves ahead by leaps and bounds in other areas, passwords stubbornly survive and reproduce with every new web site. Extensive discussions of alternative authentication schemes have produced no definitive answers.

Over forty years of research have demonstrated that passwords are plagued by security problems [2] and openly hated by users [3]. We believe that, to make progress, the community must better systematize the knowledge that we have regarding both passwords and their alternatives [4]. However, among other challenges, unbiased evaluation of password replacement schemes is complicated by the diverse interests of various communities. In our experience, security experts focus more on security but less on usability and practical issues related to deployment; biometrics experts focus on analysis of false negatives and naturally-occurring false positives rather than on attacks by an intelligent, adaptive adversary; usability experts tend to be optimistic about security; and originators of a scheme, whatever their background, downplay or ignore benefits that their scheme doesn't attempt to provide, thus overlooking dimensions on which it fares poorly. As proponents assert the superiority of their schemes, their objective functions are often not explicitly stated and differ substantially from those of potential adopters. Targeting different authentication problems using different criteria, some address very specific environments and narrow scenarios; others silently seek generic solutions that fit all environments at once, assuming a single choice is mandatory. As such, consensus is unlikely.

These and other factors have contributed to a long-standing lack of progress on how best to evaluate and compare authentication proposals intended for practical use. In response, we propose a standard benchmark and framework allowing schemes to be rated across a common, broad spectrum of criteria chosen objectively for relevance in wide-ranging scenarios, without hidden agenda.[1] We suggest and define 25 properties framed as a diverse set of benefits, and a methodology for comparative evaluation, demonstrated and tested by rating 35 password-replacement schemes on the same criteria, as summarized in a carefully constructed comparative table.

Both the rating criteria and their definitions were iteratively refined over the evaluation of these schemes. Discussion of evaluation details for passwords and nine representative alternatives is provided herein to demonstrate the process, and to provide evidence that the list of benefits suffices to illuminate the strengths and weaknesses of a wide universe of schemes. Though not cast in stone, we believe that the list of benefits and their specific definitions provide an excellent basis from which to work; the framework and

*An extended version of this paper is available as a University of Cambridge technical report [1].

†Frank Stajano was the lead author who conceived the project and assembled the team. All authors contributed equally thereafter.

[1]The present authors contributed to the definition of the following schemes: URRSA [5], MP-Auth [6], PCCP [7] and Pico [8]. We invite readers to verify that we have rated them impartially.

DOI 10.1109/SP.2012.44

evaluation process that we define are independent of them, although our comparative results naturally are not. From our analysis and comparative summary table, we look for clues to help explain why passwords remain so dominant, despite frequent claims of superior alternatives.

In the past decade our community has recognized a tension between security and usability: it is generally easy to provide more of one by offering less of the other. But the situation is much more complex than simply a linear trade-off: we seek to capture the multi-faceted, rather than one-dimensional, nature of both usability and security in our benefits. We further suggest that "deployability", for lack of a better word, is an important third dimension that deserves consideration. We choose to examine all three explicitly, complementing earlier comparative surveys (e.g., [9]–[11]).

Our usability-deployability-security ("UDS") evaluation framework and process may be referred to as *semi-structured evaluation of user authentication schemes*. We take inspiration from inspection methods for evaluating user interface design, including *feature inspections* and Nielsen's *heuristic analysis* based on usability principles [12].

Each co-author acted as a domain expert, familiar with both the rating framework and a subset of the schemes. For each scheme rated, the evaluation process involved one co-author studying the scheme and rating it on the defined benefits; additional co-authors reviewing each rating score; and iteratively refining the ratings as necessary through discussion, as noted in Section V-D.

Our focus is *user authentication on the web*, specifically from unsupervised end-user client devices (e.g., a personal computer) to remote verifiers. Some schemes examined involve mobile phones as auxiliary devices, but logging in directly from such constrained devices, which involves different usability challenges among other things, is not a main focus. Our present work does not directly examine schemes designed exclusively for machine-to-machine authentication, e.g., cryptographic protocols or infrastructure such as client public-key certificates. Many of the schemes we examine, however, are the technologies proposed for the human-to-machine component that may precede machine-to-machine authentication. Our choice of web authentication as target application also has significant implications for specific schemes, as noted in our results.

II. Benefits

The benefits we consider encompass three categories: usability, deployability and security, the latter including privacy aspects. The benefits in our list have been refined to a set we believe highlights important evaluation dimensions, with an eye to limiting overlap between benefits.

Throughout the paper, for brevity and consistency, each benefit is referred to with an italicized mnemonic title. This title should not be interpreted too literally; refer instead to our actual definitions below, which are informally worded to aid use. Each scheme is rated as either offering or not offering the benefit; if a scheme *almost* offers the benefit, but not quite, we indicate this with the *Quasi-* prefix. Section V-D discusses pros and cons of finer-grained scoring.

Sometimes a particular benefit (e.g., *Resilient-to-Theft*) just doesn't apply to a particular scheme (e.g., there is nothing physical to steal in a scheme where the user must memorize a secret squiggle). To simplify analysis, instead of introducing a "not applicable" value, we rate the scheme as offering the benefit—in the sense that nothing can go wrong, for that scheme, with respect to the corresponding problem.

When rating password-related schemes we assume that implementers use best practice such as salting and hashing (even though we know they often don't [13]), because we assess what the scheme's design can potentially offer: a poor implementation could otherwise kill any scheme. On the other hand, we assume that ordinary users won't necessarily follow the often unreasonably inconvenient directives of security engineers, such as never recycling passwords, or using randomly-generated ones.

A. Usability benefits

U1 *Memorywise-Effortless*: Users of the scheme do not have to remember *any* secrets at all. We grant a *Quasi-Memorywise-Effortless* if users have to remember *one* secret for everything (as opposed to one per verifier).

U2 *Scalable-for-Users*: Using the scheme for hundreds of accounts does not increase the burden on the user. As the mnemonic suggests, we mean "scalable" only from the user's perspective, looking at the cognitive load, not from a system deployment perspective, looking at allocation of technical resources.

U3 *Nothing-to-Carry*: Users do not need to carry an additional physical object (electronic device, mechanical key, piece of paper) to use the scheme. *Quasi-Nothing-to-Carry* is awarded if the object is one that they'd carry everywhere all the time anyway, such as their mobile phone, but not if it's their computer (including tablets).

U4 *Physically-Effortless*: The authentication process does not require physical (as opposed to cognitive) user effort beyond, say, pressing a button. Schemes that don't offer this benefit include those that require typing, scribbling or performing a set of motions. We grant *Quasi-Physically-Effortless* if the user's effort is limited to speaking, on the basis that even illiterate people find that natural to do.

U5 *Easy-to-Learn*: Users who don't know the scheme can figure it out and learn it without too much trouble, and then easily recall how to use it.

U6 *Efficient-to-Use*: The time the user must spend for each authentication is acceptably short. The time

required for setting up a new association with a verifier, although possibly longer than that for authentication, is also reasonable.

U7 *Infrequent-Errors*: The task that users must perform to log in usually succeeds when performed by a legitimate and honest user. In other words, the scheme isn't so hard to use or unreliable that genuine users are routinely rejected.[2]

U8 *Easy-Recovery-from-Loss*: A user can conveniently regain the ability to authenticate if the token is lost or the credentials forgotten. This combines usability aspects such as: low latency before restored ability; low user inconvenience in recovery (e.g., no requirement for physically standing in line); and assurance that recovery will be possible, for example via built-in backups or secondary recovery schemes. If recovery requires some form of re-enrollment, this benefit rates its convenience.

B. Deployability benefits

D1 *Accessible*: Users who can use passwords[3] are not prevented from using the scheme by disabilities or other physical (not cognitive) conditions.

D2 *Negligible-Cost-per-User*: The total cost per user of the scheme, adding up the costs at both the prover's end (any devices required) and the verifier's end (any share of the equipment and software required), is negligible. The scheme is plausible for startups with no per-user revenue.

D3 *Server-Compatible*: At the verifier's end, the scheme is compatible with text-based passwords. Providers don't have to change their existing authentication setup to support the scheme.

D4 *Browser-Compatible*: Users don't have to change their client to support the scheme and can expect the scheme to work when using other machines with an up-to-date, standards-compliant web browser and no additional software. In 2012, this would mean an HTML5-compliant browser with JavaScript enabled. Schemes fail to provide this benefit if they require the installation of plugins or any kind of software whose installation requires administrative rights. Schemes offer *Quasi-Browser-Compatible* if they rely on non-standard but very common plugins, e.g., Flash.

D5 *Mature*: The scheme has been implemented and deployed on a large scale for actual authentication purposes beyond research. Indicators to consider for granting the full benefit may also include whether the scheme has undergone user testing, whether the standards community has published related documents, whether open-source projects implementing the scheme exist, whether anyone other than the implementers has adopted the scheme, the amount of literature on the scheme and so forth.

D6 *Non-Proprietary*: Anyone can implement or use the scheme for any purpose without having to pay royalties to anyone else. The relevant techniques are generally known, published openly and not protected by patents or trade secrets.

C. Security benefits

S1 *Resilient-to-Physical-Observation*: An attacker cannot impersonate a user after observing them authenticate one or more times. We grant *Quasi-Resilient-to-Physical-Observation* if the scheme could be broken only by repeating the observation more than, say, 10–20 times. Attacks include shoulder surfing, filming the keyboard, recording keystroke sounds, or thermal imaging of keypad.

S2 *Resilient-to-Targeted-Impersonation*: It is not possible for an acquaintance (or skilled investigator) to impersonate a specific user by exploiting knowledge of personal details (birth date, names of relatives etc.). Personal knowledge questions are the canonical scheme that fails on this point.

S3 *Resilient-to-Throttled-Guessing*: An attacker whose rate of guessing is constrained by the verifier cannot successfully guess the secrets of a significant fraction of users. The verifier-imposed constraint might be enforced by an online server, a tamper-resistant chip or any other mechanism capable of throttling repeated requests. To give a quantitative example, we might grant this benefit if an attacker constrained to, say, 10 guesses per account per day, could compromise at most 1% of accounts in a year. Lack of this benefit is meant to penalize schemes in which it is frequent for user-chosen secrets to be selected from a small and well-known subset (low min-entropy [14]).

S4 *Resilient-to-Unthrottled-Guessing*: An attacker whose rate of guessing is constrained only by available computing resources cannot successfully guess the secrets of a significant fraction of users. We might for example grant this benefit if an attacker capable of attempting up to 2^{40} or even 2^{64} guesses per account could still only reach

[2]We could view this benefit as "low false reject rate". In many cases the scheme designer could make the false reject rate lower by making the false accept rate higher. If this is taken to an extreme we count it as cheating, and penalize it through a low score in some of the security-related benefits.

[3]Ideally a scheme would be usable by everyone, regardless of disabilities like zero-vision (blindness) or low motor control. However, for any given scheme, it is always possible to identify a disability or physical condition that would exclude a category of people and then no scheme would be granted this benefit. We therefore choose to award the benefit to schemes that do at least as well as the incumbent that is de facto accepted today, despite the fact that it too isn't perfect. An alternative to this text password baseline could be to base the metric on the ability to serve a defined percentage of the population of potential users.

fewer than 1% of accounts. Lack of this benefit is meant to penalize schemes where the space of credentials is not large enough to withstand brute force search (including dictionary attacks, rainbow tables and related brute force methods smarter than raw exhaustive search, if credentials are user-chosen secrets).

S5 *Resilient-to-Internal-Observation*: An attacker cannot impersonate a user by intercepting the user's input from inside the user's device (e.g., by key-logging malware) or eavesdropping on the cleartext communication between prover and verifier (we assume that the attacker can also defeat TLS if it is used, perhaps through the CA). As with *Resilient-to-Physical-Observation* above, we grant *Quasi-Resilient-to-Internal-Observation* if the scheme could be broken only by intercepting input or eavesdropping cleartext more than, say, 10–20 times. This penalizes schemes that are not replay-resistant, whether because they send a static response or because their dynamic response countermeasure can be cracked with a few observations. This benefit assumes that general-purpose devices like software-updatable personal computers and mobile phones may contain malware, but that hardware devices dedicated exclusively to the scheme can be made malware-free. We grant *Quasi-Resilient-to-Internal-Observation* to two-factor schemes where both factors must be malware-infected for the attack to work. If infecting only one factor breaks the scheme, we don't grant the benefit.

S6 *Resilient-to-Leaks-from-Other-Verifiers*: Nothing that a verifier could possibly leak can help an attacker impersonate the user to another verifier. This penalizes schemes where insider fraud at one provider, or a successful attack on one back-end, endangers the user's accounts at other sites.

S7 *Resilient-to-Phishing*: An attacker who simulates a valid verifier (including by DNS manipulation) cannot collect credentials that can later be used to impersonate the user to the actual verifier. This penalizes schemes allowing phishers to get victims to authenticate to lookalike sites and later use the harvested credentials against the genuine sites. It is not meant to penalize schemes vulnerable to more sophisticated real-time man-in-the-middle or relay attacks, in which the attackers have one connection to the victim prover (pretending to be the verifier) and simultaneously another connection to the victim verifier (pretending to be the prover).

S8 *Resilient-to-Theft*: If the scheme uses a physical object for authentication, the object cannot be used for authentication by another person who gains possession of it. We still grant *Quasi-Resilient-to-Theft* if the protection is achieved with the modest strength of a PIN, even if attempts are not rate-controlled, because the attack doesn't easily scale to many victims.

S9 *No-Trusted-Third-Party*: The scheme does not rely on a trusted third party (other than the prover and the verifier) who could, upon being attacked or otherwise becoming untrustworthy, compromise the prover's security or privacy.

S10 *Requiring-Explicit-Consent*: The authentication process cannot be started without the explicit consent of the user. This is both a security and a privacy feature (a rogue wireless RFID-based credit card reader embedded in a sofa might charge a card without user knowledge or consent).

S11 *Unlinkable*: Colluding verifiers cannot determine, from the authenticator alone, whether the same user is authenticating to both. This is a privacy feature. To rate this benefit we disregard linkability introduced by other mechanisms (same user ID, same IP address, etc).

We emphasize that it would be simple-minded to rank competing schemes simply by counting how many benefits each offers. Clearly some benefits deserve more weight than others—but which ones? *Scalable-for-Users*, for example, is a heavy-weight benefit if the goal is to adopt a single scheme as a universal replacement; it is less important if one is seeking a password alternative for only a single account. Providing appropriate weights thus depends strongly on the specific goal for which the schemes are being compared, which is one of the reasons we don't offer any.

Having said that, readers wanting to use weights might use our framework as follows. First, examine and score each individual scheme on each benefit; next, compare (groups of) competing schemes to identify precisely which benefits each offers over the other; finally, with weights that take into account the relative importance of the benefits, determine an overall ranking by rating scheme i as $S_i = \sum_j W_j \cdot b_{i,j}$. Weights W_j are constants across all schemes in a particular comparison exercise, and $b_{i,j} \in [0, 1]$ is the real-valued benefit rating for scheme i on benefit j. For different solution environments (scenarios k), the relative importance of benefits will differ, with weights W_j replaced by $W_j^{(k)}$.

In this paper we choose a more qualitative approach: we do not suggest any weights $W_j^{(k)}$ and the $b_{i,j}$ ratings we assign are not continuous but coarsely quantized. In Section V-D we discuss why. In our experience, *"the journey* (the rating exercise) *is the reward"*: the important technical insights we gained about schemes by discussing whether our ratings were fair and consistent were worth much more to us than the actual scores produced. As a take-home message for the value of this exercise, bringing a team of experts to

a shared understanding of the relevant technical issues is much more valuable than ranking the schemes linearly or reaching unanimous agreement over scoring.

III. Evaluating legacy passwords

We expect that the reader is familiar with text passwords and their shortcomings, so evaluating them is good exercise for our framework. It's also useful to have a baseline standard to refer to. While we consider "legacy passwords" as a single scheme, surveys of password deployment on the web have found substantial variation in implemention. A study of 150 sites in 2010 [13], for example, found a unique set of design choices at nearly every site. Other studies have focused on implementations of cookie semantics [15], password composition policies [16], or use of TLS to protect passwords [17]. Every study has found both considerable inconsistency and frequent serious implementation errors in practical deployments on the web.

We remind readers of our Section II assumption of best practice by implementers—thus in our ratings we do not hold against passwords the many weak implementations that their widespread deployment includes, unless due to inherent weaknesses; while on the other hand, our ratings of passwords and other schemes do assume that poor user behavior is an inherent aspect of fielded systems.

The difficulty of guessing passwords was studied over three decades ago [2] with researchers able to guess over 75% of users' passwords; follow-up studies over the years have consistently compromised a substantial fraction of accounts with dictionary attacks. A survey [3] of corporate password users found them flustered by password requirements and coping by writing passwords down on post-it notes. On the web, users are typically overwhelmed by the number of passwords they have registered. One study [18] found most users have many accounts for which they've forgotten their passwords and even accounts they can't remember registering. Another [19] used a browser extension to observe thousands of users' password habits, finding on average 25 accounts and 6 unique passwords per user.

Thus, passwords, as a purely memory-based scheme, clearly aren't *Memorywise-Effortless* or *Scalable-for-Users* as they must be remembered and chosen for each site. While they are *Nothing-to-Carry*, they aren't *Physically-Effortless* as they must be typed. Usability is otherwise good, as passwords are de facto *Easy-to-Learn* due to years of user experience and *Efficient-to-Use* as most users type only a few characters, though typos downgrade passwords to *Quasi-Infrequent-Errors*. Passwords can be easily reset, giving them *Easy-Recovery-from-Loss*.

Their highest scores are in deployability, where they receive full credit for every benefit—in part because many of our criteria are defined based on passwords. For example, passwords are *Accessible* because we defined the benefit with respect to them and accommodations already exist for most groups due to the importance of passwords. Passwords are *Negligible-Cost-per-User* due to their simplicity, and are *Server-Compatible* and *Browser-Compatible* due to their incumbent status. Passwords are *Mature* and *Non-Proprietary*, with turnkey packages implementing password authentication for many popular web development platforms, albeit not well-standardized despite their ubiquity.

Passwords score relatively poorly on security. They aren't *Resilient-to-Physical-Observation* because even if typed quickly they can be automatically recovered from high-quality video of the keyboard [20]. Perhaps generously, we rate passwords as *Quasi-Resilient-to-Targeted-Impersonation* in the absence of user studies establishing acquaintances' ability to guess passwords, though many users undermine this by keeping passwords written down in plain sight [3]. Similarly, users' well-established poor track record in selection means passwords are neither *Resilient-to-Throttled-Guessing* nor *Resilient-to-Unthrottled-Guessing*.

As static tokens, passwords aren't *Resilient-to-Internal-Observation*. The fact that users reuse them across sites means they also aren't *Resilient-to-Leaks-from-Other-Verifiers*, as even a properly salted and strengthened hash function [21] can't protect many passwords from dedicated cracking software. (Up to 50% of websites don't appear to hash passwords at all [13].) Passwords aren't *Resilient-to-Phishing* as phishing remains an open problem in practice.

Finally, their simplicity facilitates several security benefits. They are *Resilient-to-Theft* as they require no hardware. There is *No-Trusted-Third-Party*; having to type makes them *Requiring-Explicit-Consent*; and, assuming that sites add salt independently, even weak passwords are *Unlinkable*.

IV. Sample evaluation of replacement schemes

We now use our criteria to evaluate a representative sample of proposed password replacement schemes. Table I visually summarizes these and others we explored. Due to space constraints, we only explain in detail our ratings for at most one representative scheme per category (e.g. federated login schemes, graphical passwords, hardware tokens, etc.). Evaluation details for all other schemes in the table are provided in a companion technical report [1].

We introduce categories to highlight general trends, but stress that any scheme must be rated individually. Contrary to what the table layout suggests, schemes are not uniquely partitioned by the categories; several schemes belong to multiple categories, and different groupings of the schemes are possible with these same categories. For example, GrIDsure is both cognitive and graphical; and, though several of the schemes we examine use some form of underlying "one-time-passwords", we did not group them into a common category and indeed have no formal category of that name.

We emphasize that, in selecting a particular scheme for inclusion in the table or for discussion as a category representative, we do not necessarily endorse it as better than

alternatives—merely that it is reasonably representative, or illuminates in some way what the category can achieve.

A. Encrypted password managers: Mozilla Firefox

The Firefox web browser [22] automatically offers to remember passwords entered into web pages, optionally encrypting them with a master password. (Our rating assumes that this option is used; use without the password has different properties.) It then pre-fills the username and password fields when the user revisits the same site. With its Sync facility the passwords can be stored, encrypted, in the cloud. After a once-per-machine authentication ritual, they are updated automatically on all designated machines.

This scheme is *Quasi-Memorywise-Effortless* (because of the master password) and *Scalable-for-Users*: it can remember arbitrarily many passwords. Without Sync, the solution would have required carrying a specific computer; with Sync, the passwords can be accessed from any of the user's computers. However it's not more than *Quasi-Nothing-to-Carry* because a travelling user will have to carry at least a smartphone: it would be quite insecure to sync one's passwords with a browser found in a cybercafé. It is *Quasi-Physically-Effortless*, as no typing is required during authentication except for the master password once per session, and *Easy-to-Learn*. It is *Efficient-to-Use* (much more so than what it replaces) and has *Infrequent-Errors* (hardly any, except when entering the master password). It does not have *Easy-Recovery-from-Loss*: losing the master password is catastrophic.

The scheme is backwards-compatible by design and thus scores quite highly on deployability: it fully provides all the deployability benefits except for *Browser-Compatible*, unavoidably because it requires a specific browser.

It is *Quasi-Resilient-to-Physical-Observation* and *Quasi-Resilient-to-Targeted-Impersonation* because an attacker could still target the infrequently-typed master password (but would also need access to the browser). It is not *Resilient-to-Throttled-Guessing* nor *Resilient-to-Unthrottled-Guessing*: even if the master password is safe from such attacks, the original web passwords remain as vulnerable as before.[4] It is not *Resilient-to-Internal-Observation* because, even if TLS is used, it's replayable static passwords that flow in the tunnel and malware could also capture the master password. It's not *Resilient-to-Leaks-from-Other-Verifiers*, because what happens at the back-end is the same as with passwords. It's *Resilient-to-Phishing* because we assume that sites follow best practice, which includes using TLS for the login page. It is *Resilient-to-Theft*, at least under our assumption that a master password is being used. It offers *No-Trusted-Third-Party* because the Sync data is pre-encrypted locally before being stored on Mozilla's servers. It offers *Requiring-Explicit-Consent* because it pre-fills the username and password fields but the user still has to press enter to submit. Finally, it is as *Unlinkable* as passwords.

[4]Security-conscious users might adopt truly random unguessable passwords, as they need no longer remember them, but most users won't. If the scheme pre-generated random passwords it would score more highly here, disregarding pre-existing passwords. Similarly, for *Resilient-to-Leaks-from-Other-Verifiers* below, this scheme makes it easier for careful users to use a different password for every site; if it forced this behaviour (vs. just allowing it), it would get a higher score on this particular benefit.

B. Proxy-based: URRSA

Proxy-based schemes place a man-in-the-middle between the user's machine and the server. One reason for doing so, employed by Impostor [23] and URRSA [5] is to enable secure logins despite malware-infected clients.

URRSA has users authenticate to the end server using one-time codes carried on a sheet of paper. At registration the user enters the password, P_j, for each account, j, to be visited; this is encrypted at the proxy with thirty different keys, K_i, giving $C_i = E_{K_i}(P_j)$. The C_i act as one-time codes which the user prints and carries. The codes are generally 8-10 characters long; thirty codes for each of six accounts fit on a two-sided sheet. The keys, but not the passwords, are stored at the proxy. At login the user visits the proxy, indicates which site is desired, and is asked for the next unused code. When he enters the code it is decrypted and passed to the end login server: $E^{-1}_{K_i}(C_i) = P_j$. The proxy never authenticates the user, it merely decrypts with an agreed-upon key, the code delivered by the user.

Since it requires carrying one-time codes URRSA is *Memorywise-Effortless*, but not *Scalable-for-Users* or *Nothing-to-Carry*. It is not *Physically-Effortless* but is *Easy-to-Learn*. In common with all of the schemes that involve transcribing codes from a device or sheet it is not *Efficient-to-Use*. However, we do consider it to have *Quasi-Infrequent-Errors*, since the codes are generally 8-10 characters. It does not have *Easy-Recovery-from-Loss*: a revocation procedure is required if the code sheet is lost or stolen. Since no passwords are stored at the proxy the entire registration must be repeated if this happens.

In common with other paper token schemes it is not *Accessible*. URRSA has *Negligible-Cost-per-User*. Rather than have a user change browser settings, URRSA relies on a link-translating proxy that intermediates traffic between the user and the server; this translation is not flawless and some functionality may fail on complex sites, thus we consider it only *Quasi-Server-Compatible*. It is, however, *Browser-Compatible*. It is neither *Mature* nor *Non-Proprietary*.

In common with other one-time code schemes it is not *Resilient-to-Physical-Observation*, since a camera might capture all of the codes on the sheet. Since it merely inserts a proxy it inherits many security weaknesses from the legacy password system it serves: it is *Quasi-Resilient-to-Targeted-Impersonation* and is not *Resilient-to-Throttled-Guessing* or *Resilient-to-Unthrottled-Guessing*. It is *Quasi-Resilient-to-Internal-Observation* as observing the client during authentication does not allow passwords to be captured, but breaking

the proxy-to-server TLS connection does. It inherits from passwords the fact that it is not *Resilient-to-Leaks-from-Other-Verifiers*, but the fact that it is *Resilient-to-Phishing* from other one-time schemes. It is not *Resilient-to-Theft* nor *No-Trusted-Third-Party*: the proxy must be trusted. It offers *Requiring-Explicit-Consent* and is *Unlinkable*.

C. Federated Single Sign-On: OpenID

Federated single sign-on enables web sites to authenticate a user by redirecting them to a trusted identity server which attests the users' identity. This has been considered a "holy grail" as it could eliminate the problem of remembering different passwords for different sites. The concept of federated authentication dates at least to the 1978 Needham-Schroeder key agreement protocol [24] which formed the basis for Kerberos [25]. Kerberos has inspired dozens of proposals for federated authentication on the Internet; Pashalidis and Mitchell provided a complete survey [26]. A well-known representative is OpenID,[5] a protocol which allows any web server to act as an "identity provider" [27] to any server desiring authentication (a "relying party"). OpenID has an enthusiastic group of followers both in and out of academia, but it has seen only patchy adoption with many sites willing to act as identity providers but few willing to accept it as relying parties [28].

In evaluating OpenID, we note that in practice identity providers will continue to use text passwords to authenticate users in the forseeable future, although the protocol itself allows passwords to be replaced by a stronger mechanism. Thus, we rate the scheme *Quasi-Memorywise-Effortless* in that most users will still have to remember one master password, but *Scalable-for-Users* as this password can work for multiple sites. OpenID is *Nothing-to-Carry* like passwords and *Quasi-Physically-Effortless* because passwords only need to be typed at the identity provider. Similarly, we rate it *Efficient-to-Use* and *Infrequent-Errors* in that it is either a password authentication or can occur automatically in a browser with cached login cookies for the identity provider. However, OpenID has found that selecting an opaque "identity URL" can be a significant usability challenge without a good interface at the relying party, making the scheme only *Quasi-Easy-to-Learn*. OpenID is *Easy-Recovery-from-Loss*, equivalent to a password reset.

OpenID is favorable from a deployment standpoint, providing all benefits except for *Server-Compatible*, including *Mature* as it has detailed standards and many open-source implementations. We do note however that it requires identity providers yield some control over trust decisions and possibly weaken their own brand [28], a deployment drawback not currently captured in our criteria.

[5]OpenID is often confused with OAuth, a technically unrelated protocol for delegating access to one's accounts to third parties. The recent OpenID Connect proposal merges the two. We consider the OpenID 2.0 standard here, though all current versions score identically in our framework.

Security-wise, OpenID reduces most attacks to only the password authentication between a user and his or her identity provider. This makes it somewhat difficult to rate; we consider it *Quasi-Resilient-to-Throttled-Guessing*, *Quasi-Resilient-to-Unthrottled-Guessing*, *Quasi-Resilient-to-Targeted-Impersonation*, *Quasi-Resilient-to-Physical-Observation* as these attacks are possible but only against the single identity provider (typically cached in a cookie) and not for each login to all verifiers. However, it is not *Resilient-to-Internal-Observation* as malware can either steal persistent login cookies or record the master password. OpenID is also believed to be badly non-*Resilient-to-Phishing* since it involves re-direction to an identity provider from a relying party [29]. OpenID is *Resilient-to-Leaks-from-Other-Verifiers*, as relying parties don't store users passwords. Federated schemes have been criticized on privacy grounds and, while OpenID does enable technically savvy users to operate their own identity provider, we rate OpenID as non-*Unlinkable* and non-*No-Trusted-Third-Party* as the vast majority of users aren't capable of doing so.

D. Graphical passwords: Persuasive Cued Clickpoints (PCCP)

Graphical passwords schemes attempt to leverage natural human ability to remember images, which is believed to exceed memory for text. We consider as a representative PCCP [7] (Persuasive Cued Click-Points), a cued-recall scheme. Users are sequentially presented with five images on each of which they select one point, determining the next image displayed. To log in, all selected points must be correctly re-entered within a defined tolerance. To flatten the password distribution, during password creation a randomly-positioned portal covers a portion of each image; users must select their point from therein (the rest of each image is shaded slightly). Users may hit a "shuffle" button to randomly reposition the portal to a different region—but doing so consumes time, thus persuading otherwise. The portal is absent on regular login. Published security analysis and testing report reasonable usability and improved security over earlier schemes, specifically in terms of resistance to both *hotspots* and *pattern-based* attacks [11].

While not *Memorywise-Effortless*, nor *Scalable-for-Users* due to extra cognitive load for each account password, PCCP offers advantages over text passwords (and other uncued schemes) due to per-account image cues reducing password interference. It is *Easy-to-Learn* (usage and mental models match web passwords, but interface details differ), but only *Quasi-Efficient-to-Use* (login times on the order of 5s to 20s exceed text passwords) and at best *Quasi-Infrequent-Errors*.

PCCP is not *Accessible* (consider blind users) and has *Negligible-Cost-per-User*. It is not *Server-Compatible*; though it might be made so by having a proxy act as intermediary (much as URRSA does). It is *Browser-Compatible*. It is not *Mature*, but apparently *Non-Proprietary*.

PCCP is not *Resilient-to-Physical-Observation* (due to video-camera shoulder surfing), but is *Resilient-to-Targeted-Impersonation* (personal knowledge of a target user does not help attacks). We rate it *Quasi-Resilient-to-Throttled-Guessing* due to portal persuasion increasing password randomness, but note individual users may repeatedly bypass portal recommendations. Although the persuasion is also intended to mitigate offline attacks, we rate it not *Resilient-to-Unthrottled-Guessing* as studies to date have been limited to full password spaces of 2^{43} (which are within reach of offline dictionary attack, especially for users choosing more predictable passwords, assuming verifier-stored hashes are available). It is not *Resilient-to-Internal-Observation* (static passwords are replayable). It is *Resilient-to-Leaks-from-Other-Verifiers* (distinct sites can insist on distinct image sets). PCCP is *Resilient-to-Phishing* per our strict definition of that benefit; to obtain the proper per-user images, a phishing site must interact (e.g., by MITM) with a legitimate server. PCCP matches text passwords on being *Unlinkable*.

E. Cognitive authentication: GrIDsure

Challenge-Response schemes attempt to address the replay attack on passwords by having the user deliver proof that he knows the secret without divulging the secret itself. If memorization and computation were no barrier then the server might challenge the user to return a cryptographic hash of the user's secret combined with a server-selected nonce. However, it is unclear if a scheme within the means of human memory and calculating ability is achievable. We examine the commercial offering GrIDsure (a variant of which is described in a paper [30] by other authors) as representative of the class.

At registration the user is presented with a grid (e.g., 5×5) and selects a pattern, or sequence of cells. There are 25^4 possible length-4 patterns, for example. At login the user is again presented with the grid, but now populated with digits. To authenticate he transcribes the digits in the cells corresponding to his pattern. Since the association of digits to cells is randomized the string typed by the user is different from login to login. Thus he reveals knowledge of his secret without typing the secret itself.

This scheme is similar to passwords in terms of usability and we (perhaps generously) rate it identically in terms of many usability benefits. An exception is that it's only *Quasi-Efficient-to-Use*: unlike passwords, which can often be typed from muscle memory, transcribing digits from the grid cells requires effort and attention and is likely to be slower.

We consider the scheme as not *Accessible* as the two-dimensional layout seems unusable for blind users. The scheme has *Negligible-Cost-per-User*, in terms of technology. It is not *Server-Compatible* but is *Browser-Compatible*. It is not *Mature*. We rate it not *Non-Proprietary*, as the intellectual property status is unknown.

The security properties are, again, similar to passwords in many respects. It is not *Resilient-to-Physical-Observation*, as a camera that captures both the grid and user input quickly learns the secret. It is an improvement on passwords in that it is *Resilient-to-Targeted-Impersonation*: we assume that an attacker is more likely to guess secret strings than secret patterns based on knowledge of the user. However, its small space of choices prevents it from being *Resilient-to-Throttled-Guessing* or *Resilient-to-Unthrottled-Guessing*. In spite of the one-time nature of what the user types the scheme is not *Resilient-to-Internal-Observation*: too many possible patterns are eliminated at each login for the secret to withstand more than three or four observations. It shares the remaining security benefits with passwords.

F. Paper tokens: OTPW

Using paper to store long secrets is the cheapest form of a physical login token. The concept is related to military codebooks used throughout history, but interest in using possession of paper tokens to authenticate humans was spurred in the early 1980's by Lamport's hash-chaining scheme [31], later developed into S/KEY [32]. OTPW is a later refinement, developed by Kuhn in 1998 [33], in which the server stores a larger set of independent hash values, consisting of about 4 kB per user. The user carries the hash pre-images, printed as 8-character values like `IZdB bqyH`. Logging in requires typing a "prefix password" as well as one randomly-queried hash-preimage.

OTPW rates poorly for usability: the prefix password means the scheme isn't *Memorywise-Effortless* or *Scalable-for-Users*; it also isn't *Nothing-to-Carry* because of the paper token. The typing of random passwords means the scheme also isn't *Physically-Effortless*, *Efficient-to-Use* or *Infrequent-Errors*. We do expect that the scheme is *Easy-to-Learn*, as typing in a numbered password upon request is only marginally more difficult than using text passwords. It is also *Easy-Recovery-from-Loss* as we expect most users can easily print a new sheet if needed.

Paper-based tokens are cheap and easy to deploy. We rate OTPW as non-*Accessible* because plain printing may be insufficient for visually-impaired users, though alternatives (e.g. braille) may be available. We consider the price of printing to be *Negligible-Cost-per-User*. While not *Server-Compatible*, the scheme is *Browser-Compatible*. Finally, OTPW has a mature open-source implementation, making it *Mature* and *Non-Proprietary*.

Though OTPW is designed to resist human observation compared to S/KEY, it isn't *Resilient-to-Physical-Observation* because the printed sheet of one-time codes can be completely captured by a camera. Otherwise, OTPW achieves all other security benefits. Because login codes are used only once and randomly generated, the scheme is *Resilient-to-Throttled-Guessing*, *Resilient-to-Unthrottled-Guessing* and *Resilient-to-Internal-Observation*.

It is *Resilient-to-Phishing* as it is impractical for a user to enter all of their secrets into a phishing website even if asked, and *Resilient-to-Theft* thanks to the prefix password. As a one-to-one scheme with different secrets for each server, it is *Resilient-to-Leaks-from-Other-Verifiers*, *No-Trusted-Third-Party* and *Unlinkable*. Finally, the typing required makes it *Requiring-Explicit-Consent*.

G. Hardware tokens: RSA SecurID

Hardware tokens store secrets in a dedicated tamper-resistant module carried by the user; the RSA SecurID [34] family of tokens is the long-established market leader. Here we refer to the simplest dedicated-hardware version, which has only a display and no buttons or I/O ports. Each instance of the device holds a secret "seed" known to the back-end. A cryptographically strong transform generates a new 6-digit code from this secret every 60 seconds. The current code is shown on the device's display. On enrollment, the user connects to the administrative back-end through a web interface, where he selects a PIN and where the pairing between username and token is confirmed. From then on, for authenticating, instead of username and password the user shall type username and "passcode" (concatenation of a static 4-digit PIN and the dynamic 6-digit code). RSA offers an SSO facility to grant access to several corporate resources with the same token; but we rate this scheme assuming there won't be a single SSO spanning all verifiers.

In March 2011 attackers compromised RSA's back-end database of seeds [35], which allowed them to predict the codes issued by any token. This reduced the security of each account to that of its PIN until the corresponding token was recalled and reissued.

The scheme is not *Memorywise-Effortless* nor *Scalable-for-Users* (it needs a new token and PIN per verifier). It's not *Physically-Effortless*, because the user must transcribe the passcode. It's simple enough to be *Easy-to-Learn*, but *Quasi-Efficient-to-Use* because of the transcription. We rate it as having *Quasi-Infrequent-Errors*, like passwords, though it might be slightly worse. It is not *Easy-Recovery-from-Loss*: the token must be revoked and a new one reissued.

The scheme is not *Accessible*: blind users cannot read the code off the token. No token-based scheme can offer *Negligible-Cost-per-User*. The scheme is not *Server-Compatible* (a new back-end is required) but it is *Browser-Compatible*. It is definitely *Mature*, but not *Non-Proprietary*.

As for security, because the code changes every minute, SecurID is *Resilient-to-Physical-Observation*, *Resilient-to-Targeted-Impersonation*, *Resilient-to-Throttled-Guessing* and *Resilient-to-Unthrottled-Guessing* (unless we also assume that the attacker broke into the server and stole the seeds). It is *Resilient-to-Internal-Observation*: we assume that dedicated devices can resist malware infiltration. It's *Resilient-to-Leaks-from-Other-Verifiers*, as different verifiers would have their own seeds; *Resilient-to-Phishing*, because captured passcodes expire after one minute; and *Resilient-to-Theft*, because the PIN is checked at the verifier, so guesses could be rate-limited. It's not *No-Trusted-Third-Party*, as demonstrated by the March 2011 attack, since RSA keeps the seed of each token. It's *Requiring-Explicit-Consent*, as the user must transcribe the passcode, and *Unlinkable* if each verifier requires its own token.

H. Mobile-Phone-based: Phoolproof

Phoolproof Phishing Prevention [36] is another token-based design, but one in which the token is a mobile phone with special code and crypto keys. It uses public key cryptography and an SSL-like authentication protocol and was designed to be as compatible as possible with existing systems.

Phoolproof was conceived as a system to secure banking transactions against phishing, not as a password replacement. The user selects a desired site from the whitelist on the phone; the phone talks wirelessly to the browser, causing the site to be visited; an end-to-end TLS-based mutual authentication ensues between the phone and the bank's site; the user must still type the banking website password into the browser. Thus the scheme is not *Memorywise-Effortless*, nor *Scalable-for-Users*. It has *Quasi-Nothing-to-Carry* (the mobile phone). It's not *Physically-Effortless* as one must type a password. We rate it *Easy-to-Learn*, perhaps generously, and *Quasi-Efficient-to-Use* as it requires both typing a password and fiddling with a phone. It's no better than passwords on *Quasi-Infrequent-Errors*, since it still uses one. The only recovery mechanism is revocation and reissue, so it doesn't have *Easy-Recovery-from-Loss*.

On deployability: it's *Quasi-Accessible* insofar as most disabled users, including blind people, can use a mobile phone too (note the user doesn't need to transcribe codes from the phone). We assume most users will already have a phone, though perhaps not one of the right type (with Java, Bluetooth etc), hence it has *Quasi-Negligible-Cost-per-User*. The scheme requires changes, albeit minor, to both ends, so it's *Quasi-Server-Compatible* but, by our definitions, not *Browser-Compatible* because it uses a browser plugin. It's not really *Mature* (only a research prototype), but it is *Non-Proprietary*.

On security: it's *Resilient-to-Physical-Observation*, *Resilient-to-Targeted-Impersonation*, *Resilient-to-Throttled-Guessing*, *Resilient-to-Unthrottled-Guessing* because, even after observing or guessing the correct password, the attacker can't authenticate unless he also steals the user's phone, which holds the cryptographic keys. It's *Quasi-Resilient-to-Internal-Observation* because malware must compromise both the phone (to capture the private keys) and the computer (to keylog the password). It's *Resilient-to-Leaks-from-Other-Verifiers* because the phone has a key pair per verifier, so credentials are not recycled. It's definitely *Resilient-to-Phishing*, the main design requirement of the

scheme. It's *Resilient-to-Theft* because possession of the phone is insufficient: the user still needs to type user ID and password in the browser (for additional protection against theft, the authors envisage an additional PIN or biometric to authenticate the user to the device; we are not rating this). The scheme is *No-Trusted-Third-Party* if we disregard the CA that certifies the TLS certificate of the bank. It's *Requiring-Explicit-Consent* because the user must type user ID and password. Finally it's *Unlinkable* because the phone has a different key pair for each verifier.

I. Biometrics: Fingerprint recognition

Biometrics [37] are the "what you are" means of authentication, leveraging the uniqueness of physical or behavioral characteristics across individuals. We discuss in detail *fingerprint* biometrics [38]; our summary table also rates *iris recognition* [39] and voiceprint biometrics [40]. In rating for our remote authentication application, and *biometric verification* ("Is this individual asserted to be Jane Doe really Jane Doe?"), we assume unsupervised biometric hardware as might be built into client devices, vs. verifier-provided hardware, e.g., at an airport supervised by officials.

Fingerprint biometrics offer usability advantages *Memorywise-Effortless*, *Scalable-for-Users*, *Easy-to-Learn*, and *Nothing-to-Carry* (no secrets need be carried; we charge elsewhere for client-side fingerprint readers not being currently universal). Current products are at best *Quasi-Physically-Effortless* and *Quasi-Efficient-to-Use* due to user experience of not *Infrequent-Errors* (the latter two worse than web passwords) and fail to offer *Easy-Recovery-from-Loss* (here equated with requiring an alternate scheme in case of compromise, or users becoming unable to provide the biometric for physical reasons).

Deployability is poor—we rate it at best *Quasi-Accessible* due to common failure-to-register biometric issues; not *Negligible-Cost-per-User* (fingerprint reader has a cost); neither *Server-Compatible* nor *Browser-Compatible*, needing both client and server changes; at best *Quasi-Mature* for unsupervised remote authentication; and not *Non-Proprietary*, typically involving proprietary hardware and/or software.

We rate the fingerprint biometric *Resilient-to-Physical-Observation* but serious concerns include easily fooling COTS devices, e.g., by lifting fingerprints from glass surfaces with gelatin-like substances [41], which we charge by rating not *Resilient-to-Targeted-Impersonation*. It is *Resilient-to-Throttled-Guessing*, but not *Resilient-to-Unthrottled-Guessing* for typical precisions used; estimated "effective equivalent key spaces" [9, page 2032] for fingerprint, iris and voice are 13.3 bits, 19.9 bits and 11.7 bits respectively. It is not *Resilient-to-Internal-Observation* (captured samples of static physical biometrics are subject to replay in unsupervised environments), not *Resilient-to-Leaks-from-Other-Verifiers*, not *Resilient-to-Phishing* (a serious concern as biometrics are by design supposed to be hard to change), and not *Resilient-to-Theft* (see above re: targeted impersonation). As a plus, it needs *No-Trusted-Third-Party* and is *Requiring-Explicit-Consent*. Physical biometrics are also a canonical example of schemes that are not *Unlinkable*.

V. Discussion

A clear result of our exercise is that no scheme we examined is perfect—or even comes close to perfect scores. The incumbent (traditional passwords) achieves all benefits on deployability, and one scheme (the CAP reader, discussed in the tech report [1]) achieves all in security, but no scheme achieves all usability benefits. Not a single scheme is dominant over passwords, i.e., does better on one or more benefits and does at least as well on all others. Almost all schemes do better than passwords in some criteria, but all are worse in others: as Table I shows, no row is free of red (horizontal) stripes.

Thus, the current state of the world is a Pareto equilibrium. Replacing passwords with any of the schemes examined is not a question of giving up an inferior technology for something unarguably better, but of giving up one set of compromises and trade-offs in exchange for another. For example, arguing that a hardware token like RSA SecurID is better than passwords implicitly assumes that the security criteria where it does better outweigh the usability and deployability criteria where it does worse. For accounts that require high assurance, security benefits may indeed outweigh the fact that the scheme doesn't offer *Nothing-to-Carry* nor *Negligible-Cost-per-User*, but this argument is less compelling for lower value accounts.

The usability benefits where passwords excel—namely, *Nothing-to-Carry*, *Efficient-to-Use*, *Easy-Recovery-from-Loss*—are where essentially all of the stronger security schemes need improvement. None of the paper token or hardware token schemes achieves even two of these three. In expressing frustration with the continuing dominance of passwords, many security experts presumably view these two classes of schemes to be sufficiently usable to justify a switch from passwords. The web sites that crave user traffic apparently disagree.

Some sets of benefits appear almost incompatible, e.g., the pair (*Memorywise-Effortless*, *Nothing-to-Carry*) is achieved only by biometric schemes. No schemes studied achieve (*Memorywise-Effortless*, *Resilient-to-Theft*) fully, nor (*Server-Compatible*, *Resilient-to-Internal-Observation*) or (*Server-Compatible*, *Resilient-to-Leaks-from-Other-Verifiers*), though several almost do. Note that since compatibility with existing servers almost assures a static replayable secret, to avoid its security implications, many proposals abandon being *Server-Compatible*.

A. Rating categories of schemes

Password managers offer advantages over legacy passwords in selected usability and security aspects without

Category	Scheme	Described in section	Reference	Usability								Deployability						Security									
				Memorywise-Effortless	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	*Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Throttled-Guessing*	*Resilient-to-Unthrottled-Guessing*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*No-Trusted-Third-Party*	*Requiring-Explicit-Consent*	*Unlinkable*
(Incumbent)	Web passwords	III	[13]	●		●		●	●	○	●	●	●	●	●	●	●		○						●	●	●	●
Password managers	Firefox	IV-A	[22]	○	●	○	○	●	●	●		●	●	●		●	●	○	○					●	●	●	●	●
	LastPass		[42]	○	●	○	○	●	●	●	○	●	○	●		●		○	○	○	○		○	●	●		●	●
Proxy	URRSA	IV-B	[5]	●				●		○			●	○	●				○			○		●			●	●
	Impostor		[23]	○	●	●		●				●		●	○		●	●	○			○		●	●		●	●
Federated	OpenID	IV-C	[27]	○	●	●	○	○	●	●	●	●	●		●	●	●	○	○	○	○		●		●		●	
	Microsoft Passport		[43]	○	●	●	○	●	●	●	●	●			●	●		○	○	○	○		●		●		●	
	Facebook Connect		[44]	○	●	●	○	●	●	●	●	●	●		●	●		○	○	○	○		●		●			
	BrowserID		[45]	○	●	●	○	●	●	●	●	●	●		○	○		○	○	○	○		●		●		●	
	OTP over email		[46]	○	●	●		●		●	●	●	●		●		●	○	○	○	○		●	●	●		●	
Graphical	PCCP	IV-D	[7]			●		●	○	○	●		●		●		●		●	○			●	●	●	●	●	●
	PassGo		[47]			●		●	○	○	●		●		●	○	●		●						●	●	●	●
Cognitive	GrIDsure (original)	IV-E	[30]			●		●	○	○	●		●		●				●						●	●	●	●
	Weinshall		[48]			●							●		●		●	○	●				●	●	●	●	●	●
	Hopper Blum		[49]			●						●	●		●		●	○	●				●	●	●	●	●	●
	Word Association		[50]			●		●	●	○	○	●	●		●		●								●	●	●	●
Paper tokens	OTPW	IV-F	[33]					●			●		●		●	●	●		●	●	●	●	●	●	●	●	●	●
	S/KEY		[32]	●				●		○	●		●		●	●	●		●	●	●	●	●	○		●	●	●
	PIN+TAN		[51]					●		○	○		○		●	●	●		●	●	●	●	●	●	○	●	●	●
Visual crypto	PassWindow		[52]	●									○		●	●	●	○	●	●	●	●	●	●		●	●	●
Hardware tokens	RSA SecurID	IV-G	[34]					●	○	○					●	●	●	●	●	●	●	●	●	●		●	●	●
	Yubikey		[53]					●	○	○		●			●	●	●	●	●	●	●	●	●	●		●	●	●
	Ironkey		[54]	○	●		○	○	○	○		●		●	●	●	●	○			○		●	●	●	●	●	●
	CAP reader		[55]					●	○	○					●	●	●	●	●	●	●	●	●	●		●	●	●
	Pico		[8]	●	●		●		○	○							●	●	●	●	●	●	●	●	○	●	●	●
Phone-based	Phoolproof	IV-H	[36]			○		●	○	○		○	○	○			●	●	●	●	○	●	●	●		●	●	●
	Cronto		[56]			○		●	○	○			○		●	●	●	●	●	●	○	●	●	●		●	●	●
	MP-Auth		[6]			○		●	○		○	○	○					○					●	●		●	●	●
	OTP over SMS			●	●	○		●		○	○	○			●	●	●	●	●	●	○	●	●	○		●	●	●
	Google 2-Step		[57]			○		●	○	○	○	○			●	●		○	○	●	●	●	●	●		●	●	●
Biometric	Fingerprint	IV-I	[38]	●	●	●	○	●	○			○					○	●		●						●	●	
	Iris		[39]	●	●	●	○	●	○			○					○	●		●						●	○	
	Voice		[40]	●	●	●	○	●	○			○	○				○	●		○							●	
Recovery	Personal knowledge		[58]	○		●		●	●	○	●	●	●		●	●	●								●	●	●	●
	Preference-based		[59]	○		●		●	○	●	○	●	●		●				○					●	●	●	●	●
	Social re-auth.		[60]			●		●		●		●	●		●	○		○		●	●	○	●	●			●	○

●= offers the benefit; ○= almost offers the benefit; *no circle* = does not offer the benefit.

▮= better than passwords; ▮= worse than passwords; *no background pattern* = no change.

We group related schemes into categories. For space reasons, in the present paper we describe at most one representative scheme per category; the companion technical report [1] discusses all schemes listed.

Table I
COMPARATIVE EVALUATION OF THE VARIOUS SCHEMES WE EXAMINED

losing much. They could become a staple of users' coping strategies if passwords remain widespread, enabling as a major advantage the management of an ever-increasing number of accounts (*Scalable-for-Users*). However, the underlying technology remains replayable, static (mainly user-chosen) passwords.

Federated schemes are particularly hard to grade. Proponents note that security is good if authentication to the identity provider (IP) is done with a strong scheme (e.g., one-time passwords or tokens). However in this case usability is inherited from that scheme and is generally poor, per Table I. This also reduces federated schemes to be a placeholder for a solution rather than a solution itself. If authentication to the IP relies on passwords, then the resulting security is only a little better than that of passwords themselves (with fewer password entry instances exposed to attack).

Graphical passwords can approach text passwords on usability criteria, offering some security gain, but static secrets are replayable and not *Resilient-to-Internal-Observation*. Despite adoption for device access-control on some touch-screen mobile devices, for remote web authentication the advantages appear insufficient to generally displace a firmly-entrenched incumbent.

Cognitive schemes show slender improvement on the security of passwords, in return for worse usability. While several schemes attempt to achieve *Resilient-to-Internal-Observation*, to date none succeed: the secret may withstand one observation or two [61], but seldom more than a handful [62]. The apparently inherent limitations [63], [64] of cognitive schemes to date lead one to question if the category can rise above one of purely academic interest.

The hardware token, paper token and phone-based categories of schemes fare very well in security, e.g., most in Table I are *Resilient-to-Internal-Observation*, easily beating other classes. However, that S/KEY and SecurID have been around for decades and have failed to slow down the inexorable rise of passwords suggests that their drawbacks in usability (e.g., not *Scalable-for-Users*, nor *Nothing-to-Carry*, nor *Efficient-to-Use*) and deployability (e.g., hardware tokens are not *Negligible-Cost-per-User*) should not be over-looked. Less usable schemes can always be mandated, but this is more common in situations where a site has a de facto monopoly (e.g., employee accounts or government sites) than where user acceptance matters. Experience shows that the large web-sites that compete for both traffic and users are reluctant to risk bad usability [16]. Schemes that are less usable than passwords face an uphill battle in such environments.

Biometric schemes have mixed scores on our usability metrics, and do poorly in deployability and security. As a major issue, physical biometrics being inherently non-*Resilient-to-Internal-Observation* is seriously compounded by biometrics missing *Easy-Recovery-from-Loss* as well, with re-issuance impossible [9]. Thus, e.g., if malware captures the digital representation of a user's iris, possible replay makes the biometric no longer suitable in unsupervised environments. Hence despite security features appropriate to control access to physical locations under the supervision of suitable personnel, biometrics aren't well suited for unsupervised web authentication where client devices lack a trusted input path and means to verify that samples are live.

B. Extending the benefits list

Our list of benefits is not complete, and indeed, any such list could always be expanded. We did not include resistance to active-man-in-the-middle, which a few examined schemes may provide, or to relay attacks, which probably none of them do. However, tracking all security goals, whether met or not, is important and considering benefits that indicate resistance to these (and additional) attacks is worthwhile.

Continuous authentication (with ongoing assurances rather than just at session start, thereby addressing session hijacking) is a benefit worth considering, although a goal of few current schemes. Positive user affectation (how pleasant users perceive use of a scheme to be) is a standard usability metric we omitted; unfortunately, the literature currently lacks this information for most schemes. The burden on the end-user in migrating from passwords (distinct from the deployability costs of modifying browser and server infrastructure) is another important cost—both the one-time initial setup and per-account transition costs. While ease of resetting and revoking credentials falls within *Easy-Recovery-from-Loss*, the benefit does not include user and system aspects related to ease of renewing credentials that expire within normal operations (excluding loss). Other missing cost-related benefits are low cost for initial setup (including infrastructure changes by all stakeholders); low cost for ongoing administration, support and maintenance; and low overall complexity (how many inter-related "moving parts" a system has). We don't capture continued availability under denial-of-service attack, ease of use on mobile devices, nor the broad category of economic and business effects—e.g., the lack of incentive to be a relying party is cited as a main reason for OpenID's lack of adoption [28].

We have not attempted to capture these and other benefits in the present paper, though all fit into the framework and could be chosen by others using this methodology. Alas, many of these raise a difficulty: assigning ratings might be even more subjective than for existing benefits.

C. Additional nuanced ratings

We considered, but did not use, a "fatal" rating to indicate that a scheme's performance on a benefit is so poor that the scheme should be eliminated from serious consideration. For example, the 2–3 minutes required for authentication using the Weinshall or Hopper-Blum schemes may make them "fatally-non-*Efficient-to-Use*", likely preventing widespread adoption even if virtually all other benefits were provided.

We decided against this because for many properties, it isn't clear what level of failure to declare as fatal.

We also considered a "power" rating to indicate that a scheme optionally enables a benefit for power users—e.g., OpenID could be rated "amenable-to-*No-Trusted-Third-Party*" as users can run their own identity servers, in contrast to Facebook Connect or Microsoft Passport. The popularity of webmail-based password reset indicates most users accede to a heavily-trusted third party for their online identities already, so "amenable-to" may suffice for adoption. OpenID is arguably amenable to every security benefit for power users, but doesn't provide them for common users who use text passwords to authenticate to their identity provider. However, as one could argue for an amenable-to rating for many properties of many schemes, we maintained focus on properties provided by default to all users.

D. Weights and finer-grained scoring

We reiterate a caution sounded at the end of Section II: the benefits chosen as metrics are not all of equal weight. The importance of any particular benefit depends on target use and threat environment. While one could assign weights to each column to compute numerical scores for each scheme, providing exact weights is problematic and no fixed values would suit all scenarios; nonetheless, our framework allows such an endeavour. For finer-grained evaluation, table cell scores like *partially* could also be allowed beyond our very coarse {*no*, *almost*, *yes*} quantization, to further delineate similar schemes. This has merit but brings the danger of being "precisely wrong", and too fine a granularity adds to the difficulty of scoring schemes consistently. There will be the temptation to be unrealistically precise ("If scheme X gets 0.9 for this benefit, then scheme Y should get at most 0.6"), but this demands the ability to maintain a constant level of precision *repeatably* across all cells.

We have resisted the temptation to produce an aggregate score for each scheme (e.g., by counting the number of benefits achieved), or to rank the schemes. As discussed above, fatal failure of a single benefit or combined failure of a pair of benefits (e.g., not being *Resilient-to-Internal-Observation* and fatally failing *Easy-Recovery-from-Loss* for biometrics) may eliminate a scheme from consideration. Thus, seeking schemes purely based on high numbers of benefits could well prove but a distraction.

Beyond divergences of judgement, there will no doubt be errors in judgement in scoring. The table scoring methodology must include redundancy and cross-checks sufficient to catch most such errors. (Our exercise involved one author initially scoring a scheme row, co-authors verifying the scores, and independently, cross-checks within columns to calibrate individual benefit ratings across schemes; useful clarifications of benefit definitions often resulted.) Another danger in being "too precise" arises from scoring on second-hand data inferred from papers. Coarsely-quantized but self-consistent scores are likely better than inconsistent ones.

On one hand, it could be argued that different application domains (e.g., banking vs. gaming) have different requirements and that therefore they ought to assign different weights to the benefits, resulting in a different choice of optimal scheme for each domain. However on the other hand, to users, a proliferation of schemes is in itself a failure: the meta-scheme of "use the best scheme for each application" will score rather poorly on *Scalable-for-Users*, *Easy-to-Learn* and perhaps a few other usability benefits.

E. Combining schemes

Pairs of schemes that complement each other well in a two-factor arrangement might be those where *both* achieve good scores in usability and deployability and *at least one* does so in security—so a combined scheme might be viewed as having the AND of the usability-deployability scores (i.e., the combination does not have a particular usability or deployability benefit unless both of the schemes do) and the OR of the security scores (i.e., the combination has the security benefit if either of the schemes do). An exception would appear to be the usability benefit *Scalable-for-Users* which a combination might inherit from either component.

However, this is necessarily just a starting point for the analysis: it is optimistic to assume that two-component schemes always inherit benefits in this way. Wimberly and Liebrock [65] observed that the presence of a second factor caused users to pick much weaker passwords than if passwords alone were used to protect an account—as predicted by Adams's "risk thermostat" model [66]. Thus, especially where user choice is involved, there can be an erosion of the efficacy of one protection when a second factor is known to be in place. Equally, defeating one security mechanism may also make it materially easier to defeat another. We rated, e.g., Phoolproof *Quasi-Resilient-to-Internal-Observation* because it requires an attacker to compromise both a PC and a mobile device. However, malware has already been observed in the wild which leverages a compromised PC to download further malware onto mobile devices plugged into the PC for a software update [67].

See O'Gorman [9] for suggested two-factor combinations of biometrics, passwords, and tokens, for various applications (e.g., combining a hardware token with a biometric). Another common suggestion is pairing a federated scheme with a higher-security scheme, e.g., a hardware token.

VI. Concluding remarks

The concise overview offered by Table I allows us to see high level patterns that might otherwise be missed. We could at this stage draw a variety of conclusions and note, for example, that graphical and cognitive schemes offer only minor improvements over passwords and thus have little hope of displacing them. Or we could note that most of

the schemes with substantial improvements in both usability and security can be seen as incarnations of Single-Sign-On (including in this broad definition not only federated schemes but also "local SSO" systems [26] such as password managers or Pico). Having said that, we expect the long-term scientific value of our contribution will lie not as much in the raw data distilled herein, as in the methodology by which it was assembled. A carefully crafted benefits list and coherent methodology for scoring table entries, despite inevitable (albeit instructive) disagreements over fine points of specific scores, allows principled discussions about high level conclusions.

That a Table I scheme (the CAP reader) scored full marks in security does not at all suggest that its real-world security is perfect—indeed, major issues have been found [55]. This is a loud warning that it would be unwise to read absolute verdicts into these scores. Our ratings are useful and we stand by them, but they are not a substitute for independent critical analysis or for considering aspects we didn't rate, such as vulnerability to active man-in-the-middle attacks.

We note that the ratings implied by scheme authors in original publications are often not only optimistic, but also incomplete. Proponents, perhaps subconsciously, often have a biased and narrow view of what benefits are relevant. Our framework allows a more objective assessment.

In closing we observe that, looking at the green (vertical) and red (horizontal) patterns in Table I, most schemes do better than passwords on security—as expected, given that inventors of alternatives to passwords tend to come from the security community. Some schemes do better and some worse on usability—suggesting that the community needs to work harder there. But *every* scheme does worse than passwords on deployability. This was to be expected given that the first four deployability benefits are defined with explicit reference to what passwords achieve and the remaining two are natural benefits of a long-term incumbent, but this uneven playing field reflects the reality of a decentralized system like the Internet. Marginal gains are often not sufficient to reach the activation energy necessary to overcome significant transition costs, which may provide the best explanation of why we are likely to live considerably longer before seeing the funeral procession for passwords arrive at the cemetery.

Acknowledgments

The authors thank the anonymous reviewers whose comments helped improve the paper greatly. Joseph Bonneau is supported by the Gates Cambridge Trust. Paul C. van Oorschot is Canada Research Chair in Authentication and Computer Security, and acknowledges NSERC for funding the chair and a Discovery Grant; partial funding from NSERC ISSNet is also acknowledged. This work grew out of the Related Work section of Pico [8].

References

[1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," University of Cambridge Computer Laboratory, Tech Report 817, 2012, www.cl.cam.ac.uk/techreports/UCAM-CL-TR-817.html.

[2] R. Morris and K. Thompson, "Password security: a case history," *Commun. ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[3] A. Adams and M. Sasse, "Users Are Not The Enemy," *Commun. ACM*, vol. 42, no. 12, pp. 41–46, 1999.

[4] C. Herley and P. C. van Oorschot, "A research agenda acknowledging the persistence of passwords," *IEEE Security & Privacy*, vol. 10, no. 1, pp. 28–36, 2012.

[5] D. Florêncio and C. Herley, "One-Time Password Access to Any Server Without Changing the Server," *ISC 2008, Taipei.*

[6] M. Mannan and P. C. van Oorschot, "Leveraging personal devices for stronger password authentication from untrusted computers," *Journal of Computer Security*, vol. 19, no. 4, pp. 703–750, 2011.

[7] S. Chiasson, E. Stobert, A. Forget, R. Biddle, and P. C. van Oorschot, "Persuasive cued click-points: Design, implementation, and evaluation of a knowledge-based authentication mechanism," *IEEE Trans. on Dependable and Secure Computing*, vol. 9, no. 2, pp. 222–235, 2012.

[8] F. Stajano, "Pico: No more passwords!" in *Proc. Sec. Protocols Workshop 2011*, ser. LNCS, vol. 7114. Springer.

[9] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2019–2040, December 2003.

[10] K. Renaud, "Quantification of authentication mechanisms: a usability perspective," *J. Web Eng.*, vol. 3, no. 2, pp. 95–123, 2004.

[11] R. Biddle, S. Chiasson, and P. C. van Oorschot, "Graphical Passwords: Learning from the First Twelve Years," *ACM Computing Surveys*, vol. 44, no. 4, 2012.

[12] J. Nielsen and R. Mack, *Usability Inspection Methods*. John Wiley & Sons, Inc, 1994.

[13] J. Bonneau and S. Preibusch, "The password thicket: technical and market failures in human authentication on the web," in *Proc. WEIS 2010*, 2010.

[14] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," *IEEE Symp. Security and Privacy*, May 2012.

[15] K. Fu, E. Sit, K. Smith, and N. Feamster, "Dos and don'ts of client authentication on the web," in *Proc. USENIX Security Symposium*, 2001.

[16] D. Florêncio and C. Herley, "Where Do Security Policies Come From?" in *ACM SOUPS 2010: Proc. 6th Symp. on Usable Privacy and Security*.

[17] L. Falk, A. Prakash, and K. Borders, "Analyzing websites for user-visible security design flaws," in *ACM SOUPS 2008*, pp. 117–126.

[18] S. Gaw and E. W. Felten, "Password Management Strategies for Online Accounts," in *ACM SOUPS 2006: Proc. 2nd Symp. on Usable Privacy and Security*, pp. 44–55.

[19] D. Florêncio and C. Herley, "A large-scale study of web password habits," in *WWW '07: Proc. 16th International Conf. on the World Wide Web*. ACM, 2007, pp. 657–666.

[20] D. Balzarotti, M. Cova, and G. Vigna, "ClearShot: Eavesdropping on Keyboard Input from Video," in *IEEE Symp. Security and Privacy*, 2008, pp. 170–183.

[21] B. Kaliski, *RFC 2898: PKCS #5: Password-Based Cryptography Specification Version 2.0*, IETF, September 2000.

[22] Mozilla Firefox, ver. 10.0.2, www.mozilla.org/.

[23] A. Pashalidis and C. J. Mitchell, "Impostor: A single sign-on system for use from untrusted devices," *Proc. IEEE Globecom*, 2004.

[24] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, pp. 993–999, December 1978.

[25] J. Kohl and C. Neuman, "The Kerberos Network Authentication Service (V5)," United States, 1993.

[26] A. Pashalidis and C. J. Mitchell, "A Taxonomy of Single Sign-On Systems," in *Proc. ACISP 2003, Information Security and Privacy, 8th Australasian Conference*. Springer LNCS 2727, 2003, pp. 249–264.

[27] D. Recordon and D. Reed, "OpenID 2.0: a platform for user-centric identity management," in *DIM '06: Proc. 2nd ACM Workshop on Digital Identity Management*, 2006, pp. 11–16.

[28] S.-T. Sun, Y. Boshmaf, K. Hawkey, and K. Beznosov, "A billion keys, but few locks: the crisis of web single sign-on," *Proc. NSPW 2010*, pp. 61–72.

[29] B. Laurie, "OpenID: Phishing Heaven," January 2007, www.links.org/?p=187.

[30] R. Jhawar, P. Inglesant, N. Courtois, and M. A. Sasse, "Make mine a quadruple: Strengthening the security of graphical one-time pin authentication," in *Proc. NSS 2011*, pp. 81–88.

[31] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[32] N. Haller and C. Metz, "RFC 1938: A One-Time Password System," 1998.

[33] M. Kuhn, "OTPW — a one-time password login package," 1998, www.cl.cam.ac.uk/~mgk25/otpw.html.

[34] RSA, "RSA SecurID Two-factor Authentication," 2011, www.rsa.com/products/securid/sb/10695_SIDTFA_SB_0210.pdf.

[35] P. Bright, "RSA finally comes clean: SecurID is compromised," Jun. 2011, arstechnica.com/security/news/2011/06/rsa-finally-comes-clean-securid-is-compromised.ars.

[36] B. Parno, C. Kuo, and A. Perrig, "Phoolproof Phishing Prevention," in *Proc. Fin. Crypt. 2006*, pp. 1–19.

[37] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: a tool for information security," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 125–143, 2006.

[38] A. Ross, J. Shah, and A. K. Jain, "From Template to Image: Reconstructing Fingerprints from Minutiae Points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 544–560, 2007.

[39] J. Daugman, "How iris recognition works," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 14, no. 1, pp. 21–30, 2004.

[40] P. S. Aleksic and A. K. Katsaggelos, "Audio-Visual Biometrics," *Proc. of the IEEE*, vol. 94, no. 11, pp. 2025–2044, 2006.

[41] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial "gummy" fingers on fingerprint systems," in *SPIE Conf. Series*, vol. 4677, Apr. 2002, pp. 275–289.

[42] LastPass, www.lastpass.com/.

[43] D. P. Kormann and A. D. Rubin, "Risks of the Passport single signon protocol," *Computer Networks*, vol. 33, no. 1–6, 2000.

[44] "Facebook Connect," 2011, www.facebook.com/advertising/?connect.

[45] M. Hanson, D. Mills, and B. Adida, "Federated Browser-Based Identity using Email Addresses," *W3C Workshop on Identity in the Browser*, May 2011.

[46] T. W. van der Horst and K. E. Seamons, "Simple Authentication for the Web," in *Intl. Conf. on Security and Privacy in Communications Networks*, 2007, pp. 473–482.

[47] H. Tao, "Pass-Go, a New Graphical Password Scheme," Master's thesis, School of Information Technology and Engineering, University of Ottawa, June 2006.

[48] D. Weinshall, "Cognitive Authentication Schemes Safe Against Spyware (Short Paper)," in *IEEE Symposium on Security and Privacy*, May 2006.

[49] N. Hopper and M. Blum, "Secure human identification protocols," *ASIACRYPT 2001*, pp. 52–66, 2001.

[50] S. Smith, "Authenticating users by word association," *Computers & Security*, vol. 6, no. 6, pp. 464–470, 1987.

[51] A. Wiesmaier, M. Fischer, E. G. Karatsiolis, and M. Lippert, "Outflanking and securely using the PIN/TAN-System," *CoRR*, vol. cs.CR/0410025, 2004.

[52] "PassWindow," 2011, www.passwindow.com.

[53] Yubico, "The YubiKey Manual, v. 2.0," 2009, static.yubico.com/var/uploads/YubiKey_manual-2.0.pdf.

[54] Ironkey, www.ironkey.com/internet-authentication.

[55] S. Drimer, S. J. Murdoch, and R. Anderson, "Optimised to Fail: Card Readers for Online Banking," in *Financial Cryptography and Data Security*, 2009, pp. 184–200.

[56] Cronto, www.cronto.com/.

[57] Google Inc., "2-step verification: how it works," 2012, www.google.com/accounts.

[58] S. Schechter, A. J. B. Brush, and S. Egelman, "It's no secret: Measuring the security and reliability of authentication via 'secret' questions," in *IEEE Symp. Security and Privacy*, 2009, pp. 375–390.

[59] M. Jakobsson, L. Yang, and S. Wetzel, "Quantifying the Security of Preference-based Authentication," in *ACM DIM 2008: 4th Workshop on Digital Identity Management*.

[60] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: somebody you know," in *ACM CCS 2006*, pp. 168–178.

[61] D. Weinshall, "Cognitive Authentication Schemes Safe Against Spyware," *IEEE Symp. Security and Privacy*, 2006.

[62] P. Golle and D. Wagner, "Cryptanalysis of a Cognitive Authentication Scheme," *IEEE Symp. Security and Privacy*, 2007.

[63] B. Coskun and C. Herley, "Can "Something You Know" be Saved?" *ISC 2008, Taipei.*

[64] Q. Yan, J. Han, Y. Li, and H. Deng, "On limitations of designing usable leakage-resilient password systems: Attacks, principles and usability." Proc. NDSS, 2012.

[65] H. Wimberly and L. M. Liebrock, "Using Fingerprint Authentication to Reduce System Security: An Empirical Study," in *IEEE Symp. Security and Privacy*, 2011, pp. 32–46.

[66] J. Adams, "Risk and morality: three framing devices," in *Risk and Morality*, R. Ericson and A. Doyle, Eds. University of Toronto Press, 2003.

[67] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *ACM SPSM 2011: 1st Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 3–14.