How to Wear Your Password

Markus Jakobsson
Qualcomm Technologies, Inc.

Abstract

We describe a new authentication paradigm that seeks to achieve both a desirable user experience and a high level of security. We describe a potential implementation of an identity manager in the guise of a *smart bracelet*. This bracelet would be equipped with a low-power processor, a Bluetooth LE transmitter, an accelerometer, and a clasp that is constructed so that opening and closing it would break and close a circuit, thereby allowing an automatic detection of when the bracelet is put on and taken off. For reasons of cost, design and error avoidance, the bracelet could be designed to *not* have any user interface, nor any biometric sensors: All user interaction could be assisted by third-party devices, such as user phones and point of sale terminals.

Our approach is based on the principle of *physical and logical tethering* of an identity manager to a user (e.g., by closing the clasp), where an identity manager represents its user's interests after an initial user authentication phase, and until the user causes a disassociation by untethering the device (e.g., by opening the clasp). The authentication phase can be based on any type of authentication, and – to allow for the greatest possible simplicity of design – can be aided by a third-party device, such as the user's cell phone.

We describe the physical design, including aspects to protect against violent attacks on users. We also describe the lightweight security protocols needed for pairing, determination of user intent, and credential management, and give examples of usage scenarios – including automated login; simplified online and point-of-sale purchases; assisted appliance personalization; and automated event logging. We then overview the protocols associated with the example usage scenarios, and discuss the security implications of our proposed design.

Keywords: authentication, identity manager, password manager, physical tethering, smart bracelet, smart clasp.

1 Introduction

Although security technologies largely have kept an even pace with the development of both advances in computing and threats against the infrastructure, authentication mechanisms have largely remained the same in spite of increasing pressure on them [5]. This has led to a degradation of *both* usability and security. For example, along with the shift toward mobile computing, which has made entry of password more error-prone [10], advances in passwords cracking (e.g., [16]) and increased processing power has, in effect, made passwords *weaker*. At the same time, the increase in the number of accounts per user has caused people to have to choose between greater inconvenience; engaging in potentially dangerous reuse of passwords across sites [4]; and having to place trust in password managers [13]. As a result, there have been many calls to end the era of the password – e.g., [9, 14].

Biometrics has emerged as one possible solution to the problem of improving *both* security and user experience – especially in the context of mobile computing [12]. For many use scenarios, biometrics offer users a practical approach to authenticate to a trusted device. It is not clear, though, that biometrics are the right solution for user authentication directly to potentially *un*trusted devices – such as point of sale terminals and vending machines. Similarly, while implicit authentication proposals (e.g., [7]) offer the hope of combining ease of use with improved security by authenticating users based on their continuous and recognizable behavior, this is not a good fit for all use cases – and from a more pragmatic point of view, a practical solution has yet to be developed.

In this paper, we address challenges relating to authentication, with a focus on security and usability – and with attention to potential special cases. We propose an authentication approach that relies on an identity manager. This could be integrated in a smart watch, a smart bracelet, or other wearable technology. To avoid the drawbacks associated with verifying the identity of the user at the time of an authentication request – as is the approach of several related commercial approaches – we "lock in" an identity by determining when the identity manager is associated or disassociated with its user. We refer to this notion as "physical tethering" of the identity to the user.

Our proposal is based on a collection of techniques. Some of these are well-known concepts, such as Stajano and Anderson's *secure transient association* concept [15], where a device is provided with a configuration for a limited duration of time. We also make use of Stajano and Anderson's *resurrecting duckling* security policy, where a newly awoken device – the identity manager of the smart bracelet – accepts an identity from an "imprinting" device. There are two significant differences between our proposal and those of Stajano and Anderson, though. First, in our approach, the user identity that is sent by the imprinting device (such as a phone) is not simply asserted by this device, but rather, is first *verified* by it. More concretely, this means that the imprinting device first verifies the user identity, e.g., using biometrics or a password. Second, whereas Stajano and Anderson assume the presence of only *one* possible imprinting device, we avoid the problems that would come with an improper identity association by requiring that an identity manager receives an accelerometer trace from the imprinting device, and compares this to an accelerometer trace generated by the identity manager. If these agree, then the verified identity is imprinted for the duration of the session. This is along the same lines as what was used in the "bump" payment developed by the founders of PayPal, before they pivoted towards the technology that is now associated with PayPal.

One of the new techniques introduced in this paper is the principle that an identity imprint is made after a user puts on the identity manager (e.g., places the bracelet on his wrist and closes the clasp) and that the imprint is *removed* as the identity manager detects an event associated with removal (e.g., the opening of the clasp). This simple principle allows the identity manager to represent the user for the proper duration, without any need for ongoing identity verifications. This, in turn, translates to a simpler user experience without compromising security.

A closely related technique is used to attribute actions and determine user intent. This problem can best be seen in the context of many users (and their identity managers) standing in line to pay at a checkout register. If the point of sale terminal communicates the amount to be paid, and multiple identity managers receives the transmission, which one should pay? This is a problem that developers of wireless payment systems have long been aware of. A typical solution to this problem is what is used by the payment system Square, where a clerk matches a photo of a user (sent by his device to the point of sale terminal) to the proper customer, attributing the payment to the person at the front of the line. Instead, we automate the process by letting the identity manager receive an accelerometer trace – for example, from the stylus used to sign on the point of sale terminal – and compare this to a locally generated trace. If these two match, then the identity manager approves the payment. This type of attribution both ascertains intent and matches a transaction to the proper user. This approach, of course, can be used to determine intent in contexts other than payment transactions.

Another new technique relates to the physical design of the bracelet carrying the identity manager. One part of this has already been mentioned: the smart clasp, which reports to the processor when it is being opened and closed. This addresses theft of identity managers that are not worn: A burglar or misguided family member intending to use another person's identity manager to perform actions in his name would simply fail to do this, as the identity manager would stop representing its user as the clasp is opened, and not resume this operation until it has gone through a proper matching with a device that has verified the user's identity. However, this does not by necessity address the related problem of theft of identity managers as they are *worn*. By making sure that the tearing of a bracelet causes the breaking of the circuit (and therefore erasure of the identity it carries), "bag-snatching style" theft is also addressed. This is easily achieved by designing the bracelet so that it incorporates a thin wire going from the processor to the clasp and back — see figure 1 for an illustration of this design. A third related problem is that of aggravated assault aimed at the theft of an active identity manager. It would clearly be undesirable if the design

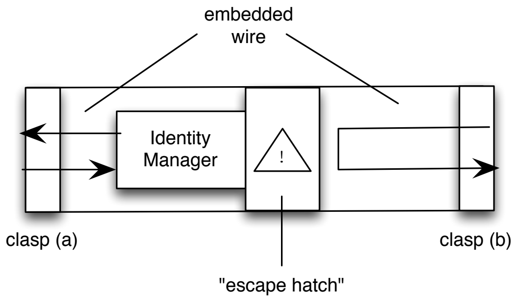

Figure 1: *A schematic of the bracelet design. The bracelet contains an identity manager and a smart clasp (parts a and b, separated in the figure). The arrows indicate a circuit that is closed when the clasp is closed, and broken when the clasp is opened or the bracelet torn. Breaking the circuit causes the identity to be erased, while closing it places the identity manager into "identity acquisition" mode. The bracelet can be physically removed by opening the "escape hatch", e.g., by removing a screw; this does not break the circuit.*

described above would lead brutal criminals to cut off the hand of a victim in order to remove the identity manager without deactivating it. Therefore, the identity manager bracelets should allow for physical disassembly of a kind that could be attempted by an average user, without causing deactivation. At the same time, it is beneficial if it is not practically possible to *reassemble* the bracelets: the disassembly, simply speaking, should *only* be an escape hatch to protect users against violent crime[1].

The use of an identity manager like ours has the potential of dramatically improving security. Users can no longer succumb to phishing attacks if they no longer use memorized credentials, and the devices representing users can tell good sites from bad (see, e.g., [8]) – a much easier problem than expecting the end user to be in charge of this assessment. Moreover, as with standard password managers, the credentials can be created to be longer and more secure than the passwords typical users choose. As the temptation of reusing credentials for multiple sites [4] vanishes, the security against breaches improves. Moreover, malware attacks are made less likely by the fact that the end user cannot install software alongside the identity manager, and the fact that the interface between the identity manager and the surrounding world is very constrained. However, we note that the exposure to malware does not entirely vanish, though, as a corrupted device interacting with an identity manager could potentially carry out bait-and-switch attacks in which a user's intent is misrepresented or associated with the wrong terms or amounts. Another potential contribution to improved

[1] It could be argued that the user could instead simply take off the bracelet, give it to his attacker, and then authenticate to the bracelet in the same way as when he puts it on himself. However, if the authentication is memory-based, this may not work, as many people's memory fail under duress.

security would come from the painless user experience associated with authentication: users will, for example, no longer be tempted to leave their computers unlocked as they go to pick up a printout. Moreover, an obvious benefit of the proposed approach is resistance to shoulder surfing; whereas this type of attack can be defended against using technology soon to appear in the marketplace (see, e.g., [3]), it is not clear what the tradeoffs between security and usability will be for such solutions.

The security benefits we identify can be obtained using other technologies – such as authentication tokens (avoiding malware corruption), biometric sensors (simplifying authentication) and password managers (taking over the task of credential management from users). However, it is the *simultaneous* achievement of the benefits that represents the security advances of our approach, along with the fact that shortcomings associated with these other approaches are avoided.

Outline: We begin in section 2 by describing different security levels, and by outlining a collection of principal usage scenarios and example applications. We then detail the basic structure of our proposal in section 3, describing the processes and protocols. We end with a discussion of security features (section 4) and a brief overview of potentially interesting directions for future work (section 5.)

2 Usage Scenarios and Application Examples

The technology we describe can be used for a variety of reasons, the most obvious of which relates directly to authentication. However, other potential uses relate more closely to personalization, tracking and attribution. As a motivation to the solution (which we describe in the next section,) we will provide some examples of uses. We begin by describing three security levels of relevance:

Levels of Security. For all of our usage examples, we consider three levels of security: *proximity verification*, *implicit assurance*, and *explicit confirmation.*

The lowest level of security – proximity verification – simply relies on verifying that an identity manager associated with an identity with access privileges is in the proximity of the object that the user interacts with – this object may be a phone or a mouse, for example.

The intermediate level of security – implicit assurance – relies on determining plausible user intent for a user associated with an identity manager that passes the proximity verification. This can be achieved by making sure that the observed user actions are matching movement data associated with the identity manager. For example, the tap on a screen to perform an action should match accelerometer data, thereby supporting that the tap was performed by the wearer of the identity manager. Similarly, the typing on a laptop keyboard or the turning of a door knob can be matched with accelerometer data to attribute the user intent to the user associated with the identity manager. The naming of this security level – implicit assurance – is due to the fact that the observed action is one that the user performs without being requested to.

The highest level of security is obtained from explicit user confirmation of an action, where the user's identity manager also has passed the proximity verification. Two examples of explicit confirmation are for a user to shake a phone (with the hand used to wear the smart bracelet), and signing on the screen of a point of sale terminal, where either the detected screen movements or the stylus accelerometer data is compared to the accelerometer data generated by the identity manager. In these settings, a user is requested to perform an action in order for this action to be matched to movement data associated with the motion of the identity manager.

Application Example 1: Logging in. Our approach can replace the traditional log-in process. On a mobile device, such as a phone, the normal use of the device may initiate the log-in process simply by waking the device up – whether this amounts to picking it up, touching its screen, or pressing a button. The process may also be initiated by the user starting an application or attempting to access a resource referenced by an application. Example resources are user address books; emails; usage log files (such as the list of most recently placed phone calls); photos or directories of photos; and the ability to place toll calls. A desktop or laptop may be accessed in a similar way, where a mouse, mousepad, or keyboard takes on the role of collecting data to be used to obtain assurance or confirmation. Different resources may be associated with different security levels – for example, a user may require an intermediate security level to unlock a phone; a low security level to access its email reader (once the phone is woken up); but a high security level to gain access to usage logs.

A special case of the log-in case is where either the identity manager or an associated proxy acts as a password manager for sites that are not compatible with the identity manager technology, and wherein log-in session is moderated by one of these devices. This can be done without the exposure of session secrets to the facilitating device [8].

Application Example 2: Paying. The technology we propose can be used to facilitate payments, whether online or point of sale payments. To perform an online payment, the user may initiate the payment process by clicking a checkout button. For low and medium risk purchases, this may be sufficient, as it provides an intermediate level of security. However, for high-risk purchases, an explicit confirmation may be required instead. Here, many factors influence the risk level of the transaction, such as the value; the type of merchandise; the user's history of purchases; and the location where the purchase is initiated. For example, a user may assign an intermediate level of security to performing payments from his phone, but his financial service provider may escalate this to a high level of security, based on the type of transactions the user performs. For a point of sale transaction, an explicit confirmation can be obtained as the user signs his signature on a point of sale terminal and the movement is correlated with the movement of his identity manager. This not only provides guarantees of user intent (which may be useful for disputed transactions) but also helps identify which one out of several possible users is to be associated with the transaction. Other types of payments – such as payments of subway fares – may not not require more security than a proximity verification.

For practical purposes, the communication between the wireless device manager and the Point of Sale terminal should be done using a communication technology that does not require device pairing. A practical choice is the *just works* mode of Bluetooth LE, with security added against impersonation attacks using a cryptographic commitment on movement data.

Application Example 3: Attribution. By comparing accelerometer traces associated with a touch screen with accelerometer traces of bracelets, one can attribute user interactions on the screen even when multiple users touch the screen simultaneously. This may give rise to new type of gaming environments, for example; this may also one of the few usage scenarios where it makes sense for users to wear *two* bracelets at the same time. In the context of gaming, it may not be necessary to attribute actions to a user identity, but some form of pseudonymity may be more practical.

Private credentials [2] is suitable in a second type of attribution related authentication, where it may not be desirable from a privacy perspective for users to have actions attributed to themselves, but where it is perfectly reasonable to attribute actions to a *group* of users. In example uses of this type, the technology we describe provides an alternative to CAPTCHAs and help defend against some forms of advertisement fraud.

3 Smart Bracelets with Identity Managers

One of the principles behind our proposal is the notion that the identity manager becomes both physically and logically associated to a person at the same time, and that both of these associations are also *broken* at the same time. More accurately, the logical association – in which a bracelet acquires an identity – is initiated after a user puts on the bracelet, and is terminated as he takes it off. This can be achieved using a smart clasp that conveys to the identity manager when it is opened and closed. Therefore, much of the functionality relates to the physical tethering of the device to its user – a bracelet that can be taken off without opening its clasp, for example, would not satisfy the requirements we place on it.

A user can, in principle, use any form of authentication during the logical association of the identity manager. For example, the user can use a voice-based credential [6, 1] to authenticate directly to the identity manager. However, in many situations it may be more practical to use a third party computational device (such as a phone) as a proxy, whether to simply collect identity information, later to convey this to the identity manager where it is validated; or to both collect and verify the identity information. In the latter case, there is a trust relationship between the identity manager and the proxy; moreover, there is also a need for the identity manger to verify that it is correctly paired before it accepts a new identity. This is done to determine user intent, and is most conveniently performed using an explicit confirmation. (For example, the explicit confirmation can be achieved by requesting that the user hold the phone in the hand associated with the bracelet and make a shaking motion, letting the identity manager compare the accelerometer traces of the two devices with each other.)

Turning now to the verification methods associated with our previously described three levels of security, we can describe the protocols between the station and the identity manager as follows:

1. **Proximity verification.** The station – whether phone, mouse, doorknob, point of sale terminal, etc – transmits a wake-up signal to the identity manager. The identity manager responds with an identity assertion, which may consist of a static unique identifier, a pseudonym, the output from a rolling code, or a cryptographic token. A schematic is shown in figure 2.

   Note that it is not possible to infer the identity simply from a plain acknowledgment – in spite of the dedicated channel – since the station does not know whether the identify manager has been disassociated from a previous identity or not.

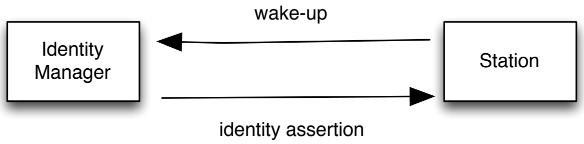

Figure 2: *In the lowest security level, a proximity verification is sufficient to proceed with a transaction. The station transmits a wake-up signal that is received by the identity manager, causing it to respond with an identity assertion. The wake-up signal can contain an indicator of the station's identity, which is compared to a whitelist kept by the identity manager. If there is a match, then an identity assertion is encrypted using a key associated with the station and the ciphertext transmitted to the station.*

2. **Implicit assurance.** Implicit assurance requires the comparison of two signals – such as two accelerometer traces, one accelerometer trace and an associated click timing signal, or similar. The comparison is preferably carried out by the identity manager – which, in a sense, is responsible for correctly representing its user – and the result conveyed to the station, along with a representation of the identity associated with the identity manager, as described above. An example protocol is detailed in figure 3.

3. **Explicit confirmation.** Explicit confirmation requires the user to perform an additional task that affects the sensors of both the station and the identity manager. This allows sensor signals to be compared in a similar

way as is done for implicit assurance, but allows for a higher precision provided a user task that generates sufficient entropy. The security level can be adjusted to a desired level by selecting the type of task or the duration of it. The same protocol as for the implicit assurance can be used – see figure 3 – except that additional requirements are placed on the movements.

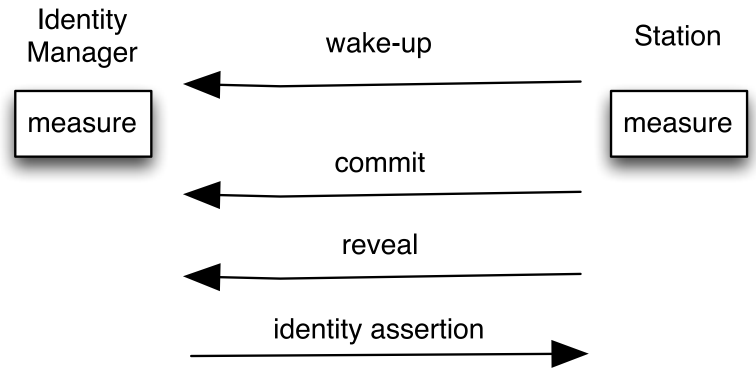

Figure 3: *The figure describes the general approach for implicit assurance and explicit confirmation. The station starts by sending a wake-up signal. As soon as the identity manager receives the wake-up signal, it measures movements m_1. At the same time, the station measures movements m_2. Within a time period T of the wake-up signal, the station transmits a commitment to the measured movements m_2 and its public key P. The commitment can be computed by applying a cryptographic hash function to m_2, P and a number r that the station chooses at random. After the time T has elapsed from the wake-up signal, the station reveals the values (m_2, P, r), similar in spirit to the techniques underlying TESLA [11]. The identity manager verifies three facts: (1) that the commitment was received within time T of the wake-up signal; (2) that the commitment corresponds to the revealed values; and (3) that the movements m_1 and $m2$ match each other sufficiently well. For explicit confirmation, it is also verified that the movements satisfy additional requirements. If all of these conditions are satisfied, then the identity manager prepares an identity assertion, encrypted using the public key P, and transmits the resulting ciphertext to the station.*

4 Security Discussion

There are two types of security assertions that can be made – relating to physical ownership, and to the information that is processed. Starting with the physical security aspect, a collection of different abuses are considered:

- A first type of abuse we consider is unauthorized use, where an identity manager is "borrowed" by a user other than the user whose identity it is associated with. This is obviously not possible, based on the design of the smart clasp, since this "resets" the identity manager as it is opened. To acquire an identity after the clasp is closed again, the user needs to authenticate – whether using a phone as an "authentication intermediary" that verifies a fingerprint or a traditional credential, or by authenticating directly to the identity manager, e.g., using voice. Therefore, a person who cannot authenticate as the owner also cannot instantiate the identity manager.

- As noted above, opening the smart clasp terminates the session during which the identity manager represents an authenticated user. Similarly, by integrating conductive wiring throughout the bracelet, it is possible to defend against abuse in which a thief tears a bracelet from the wrist of a user: Like opening the clasp, tearing the bracelet will also break the circuit.

- At the same time, we believe that a physical design of the bracelet that permits "silent" (and non-violent) removal of an identity manager from a user under duress is beneficial, to avoid aggravated assault aimed at acquiring enabled identity managers. This can be achieved by letting the conductive wiring loop from the identity manager – through the clasp – to a point close to the other side of the identity manager, where the wiring"loops back" through the clasp to a point close to where it started. The part of the bracelet right after where the wiring loops back can be attached to the identity manager with a screw that – when removed – opens the bracelet without breaking the circuit.

Turning to more traditional computer security aspects, we claim that the described technology comes with an array of benefits:

- Users are protected against phishing attacks, since users will not be in the habit of using any credentials (other than possibly to authenticate to their identity managers as these are associated with them). The identity manager will not release credentials to sites that the credentials do not belong with.

- The quality of user credentials becomes less of an issue, as these are only used in the association stage. This limits the potential use of weak user credentials to people with physical access to a targeted identity manager.

- Users are protected – to some extent – against malware. This is because the very constrained interface between the identity manager and its surroundings makes it harder for the identity manager to be corrupted. In particular, since users are unlikely to install any software on their identity managers, an entire class of vulnerabilities is avoided.

- If either the identity manager or an associated proxy acts as a password manager, this also relieves the user from the task of managing passwords with sites that are not directly compatible with the identity manager. The exposure of the associated solution to potential malware attacks depends first and foremost on what entity acts as a password manager, and if it is a phone, whether the password manager runs in a secure execution environment.

- The approach offers a degree of privacy against tracking, as the identity manager will only send identity assertions to to whitelisted stations (as described in figure 2), and to stations that are physically paired with the identity manager (as described in figure 3.)

We have not considered the question of how to respond to "authorized borrowing" of identity managers, where a user instantiates an identity manager with his identity, although he is not wearing it. This type if abuse can easily be performed by a person willing to authenticate to an identity manager in his proximity – and then let the person wearing the identity manager perform the explicit confirmation. Another version of this unwanted use is when a user closes the clasp of a bracelet without *anybody* wearing it, followed by an authentication session and an explicit confirmation (while holding the bracelet in his hand to make sure the accelerometer traces will match.) This undesirable use is analogous to password sharing, and it can be hoped that the identity risks of doing it would be understood by users.

5 Conclusions and Future Work

We have discussed a new authentication paradigm, based on physical and logical tethering of identity managers to users, with a streamlined user experience aimed at providing simplicity and robustness against failures. We have argued how our approach can be used to change how authentication is done – whether in the context of traditional access control or payments.

Our focus has been on authentication, and while we have suggested that our approach can also be used for personalization, tracking and attribution, we have not spent much effort examining these areas; we feel that this may be an interesting area of future work. Moreover, we have focused on security aspects, and have not discussed privacy issues in depth. We believe that a careful study of privacy implications and enhancements would be valuable.

Acknowledgments

Thanks to Joel Linsky, Brian Redding and Hossein Siadati for inspiring discussions and insightful feedback, and to Brent Sammons for helpful feedback on the presentation.

References

[1] Dick Tracy Logs in. `http://www.sesame-security.com`. [Accessed 6/27/2014].

[2] J. Camenisch, A. Lehmann, and G. Neven. Electronic identities need private credentials. *IEEE Security & Privacy*, 10(1):80–83, 2012.

[3] A. De Luca, E. von Zezschwitz, N. D. H. Nguyen, M.-E. Maurer, E. Rubegni, M. P. Scipioni, and M. Langheinrich. Back-of-device authentication on smartphones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2389–2398, New York, NY, USA, 2013. ACM.

[4] D. Florencio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 657–666, New York, NY, USA, 2007. ACM.

[5] C. Herley, P. C. van Oorschot, and A. S. Patrick. Passwords: If we're so smart, why are we still using them? In *Financial Cryptography*, pages 230–237, 2009.

[6] M. Jakobsson and R. Akavipat. Rethinking passwords to adapt to constrained keyboards. In *In Proceedings of Mobile Security Technologies, 2012 (2012*.

[7] M. Jakobsson, E. Shi, P. Golle, and R. Chow. Implicit authentication for mobile devices. In *Proceedings of the 4th USENIX Conference on Hot Topics in Security*, HotSec'09, pages 9–9, Berkeley, CA, USA, 2009. USENIX Association.

[8] M. Jakobsson and S. Taveau. The Case for Replacing Passwords with Biometrics. In *In Mobile Security Technologies (MoST)*, 2012.

[9] M. Kotadia. Gates predicts death of the password. `http:news.cnet.com2100-1029-5164733.html`. [Accessed 6/18/2014].

[10] S. Lee and S. Zhai. The performance of touch screen soft buttons. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 309–318, New York, NY, USA, 2009. ACM.

[11] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(Summer), 2002.

[12] M. Rothman and B. Wilson. Authentication death match: Mobility vs. passwords (and why passwords will lose), Webcast, May 17, 2011.

[13] D. Silver, S. Jana, D. Boneh, E. Chen, and C. Jackson. Password managers: Attacks and defenses. In *23rd USENIX Security Symposium (USENIX Security 14)*, San Diego, CA, Aug. 2014. USENIX Association.

[14] F. Stajano. Pico: No more passwords! In *Proceedings of the 19th International Conference on Security Protocols*, SP'11, pages 49–81, Berlin, Heidelberg, 2011. Springer-Verlag.

[15] F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Security Protocols Workshop*, pages 172–194. Springer-Verlag, 1999.

[16] R. Veras, C. Collins, and J. Thorpe. On the Semantic Patterns of Passwords and their Security Impact. In *Proceedings of NDSS '14*, 2014.