Tangible Security for Mobile Devices

YuQun Chen
Microsoft Research
One Microsoft Way
Redmond, WA USA
yuqunc@microsoft.com

Michael Sinclair
Microsoft Research
One Microsoft Way
Redmond, WA USA
sinclair@microsoft.com

ABSTRACT
Existing measures to secure the internal data on mobile devices can generally fall into two categories: passwords and biometrics. Neither is satisfactory, for different reasons. We propose a new scheme, analogous to using physical keys to unlock doors or a plug-in security dongle to unlock software. The user wears one or more security tokens whose presence in the close proximity of the user's cell phone allows the secrets on the phone to be unlocked. As long as the thief is unable to steal both the token and the cell phone, security is mostly guaranteed. We call this approach tangible security. This preliminary paper describes how such a system could be built and used.

Categories and Subject Descriptors
C.3 [**Special-purpose and Application-based Systems**]: Smartcards

General Terms
Algorithms, Design, Security, Human Factors.

Keywords
Security, tokens, RFID, NFC, encryption, AES, mobile devices.

1. INTRODUCTION
The mobile phone is fast becoming an all-encompassing, personal accessory for communication, entertainment and personal data, with Apple's iPhone an example of the latest successes. As the cell phone becomes the ultimate wallet as well as the keeper of sensitive personal information, securing it against theft of data or hacking will be of paramount importance. Among these problems, one of the biggest concerns for most users is theft of the device: A cell phone is usually easier to lose than a wallet, as the phone is normally used more often. Many software and hardware mechanisms have been proposed or implemented to make it difficult for thieves or hackers to obtain sensitive information from a stolen phone. So far, few, if any of these schemes seem promising against hardcore thieves or hackers.

The main drawback of existing protection schemes is the inconvenient requirement that the user authenticate herself to the cell phone, by typing a password or PIN code, for example. Consequently, most users opt to disable security measures completely or employ obvious passwords or PIN codes. There are proposals to use biometrics such as fingerprints to replace passwords. These methods have been proven compromisable and are still in the development stage. There are several problems with biometrics. Firstly, fingerprint technology has shown that fingerprints are fairly easy to capture and the readers can be spoofed with a surrogate print [1]. Secondly, unlike passwords, existing biometrics all rely on static and permanent data about a person, i.e., fingerprints and iris patterns; they cannot be renewed, nor replaced. There is a growing concern that once a person's biometrics data are stolen, the game is over [2].

I mimicking the real-world locks and keys, we propose a new approach to mobile phone security. The central idea is to *physically* separate the security management functionality from the mobile device itself: The user *wears* a device, a *security token*, which contains the cryptographic key(s) for unlocking sensitive data on the cell phone. Good examples of wearable tokens are a finger ring, a wristwatch or a smartcard. When unlocking a critical functionality on the cell phone, i.e., the payment module, the user simply brings the phone close to the token. In the case of the ring token, holding the phone in the hand does the trick.

When the phone and the token are in close proximity, messages are exchanged for the phone to obtain the necessary rights (from the token) to access protected functionalities or data. With a carefully devised security protocol, the thief has to steal both the cell phone and the token, a much more difficult task. Likewise, a user has to lose both items at the same time before she has to worry about her cell phone's data possibly being compromised.

We call our approach *Tangible Security* (or **TS**) for mobile phones, as it gives the user a physical embodiment of the concept of security. Our concept resembles Zero-Interaction Authentication [3], but differs from ZIA in several key aspects. Firstly, ZIA tries to secure laptop file systems using a wearable, wireless token; consequently, their protocol is heavy duty. The application scenarios for mobile phone require light-weight, fast protocols that our paper addresses. Secondly, ZIA assumes a token with its own battery that can perform sophisticated computation; whereas we want to make our tokens truly ubiquitous, robust and inexpensive; our cryptographic protocol is designed specifically for inexpensive, batteryless and RFID-like tokens. This is where the major contribution of our paper lies. And lastly, we generalize the use of wearable, near-field wireless token to a broad range of applications and make Tangible Security a new paradigm of user authentication.

2. SYSTEM DESCRIPTION
There are two main entities in a TS system: a mobile device and one or more *security tokens*. A token is akin to a key; the mobile device, a car. This paper focuses on the simplest case wherein the user carries only one security token. The multi-token protocol, in

MobiQuitous 2008, July 21-25, 2008, Dublin, Ireland.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3936
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3936

which a user wears multiple tokens and a subset of them can activate the cell phone, is beyond the scope of this paper.

In the simplest embodiment, the token contains the *root key* from which all keys used by the mobile device are derived. The more advanced embodiments can have multiple keys and secrets stored in the token. The problem of maintaining security in the system is the same, regardless of the number of secrets stored in the token. We therefore focus on the single-key case in this paper and further assume that the root key is the encryption key for all secrets on the mobile device. We denote this secret by **S**.

There is optionally a physical *kill* switch on the token. By default, the token is enabled and permits accesses by the phone when the authentication procedure is followed. In the case of the loss of the cell phone, the user can simply disable the token so that the thief cannot acquire the secret by being very close to the victim's token. Exact implementation of the switch may depend on the form factor of the token. For example, a ring token could have a sliding, circular band; the user enables/disables the token by twisting the band.

2.1 Requirements and Implications

In addition to security, some key design decisions for the token are convenience and durability. Therefore we require the normal token-mobile device communication to be contactless but near-field in nature. This way the user only has to put the cell phone in the vicinity of the token to activate protected functionalities. It is vital that this communication be localized: That the mobile device is able to talk to the token implies *close proximity* and henceforth is highly likely to be worn or held by the user.

The token itself is batteryless so that the user need not worry about recharging or battery replacement. This also makes the token more robust in most environments. Preferably the user ought to be able to wear the token all the time – sleeping, in the shower, during exercise or while swimming.

An obvious way to satisfy the above requirements is to use an RFID-like technology for the near field such as Near-Field Communication (NFC), which is chosen in our current prototyping effort. Note that one can also use personal area networks such as using the human skin for data transmission [4], though in that case the token may need its own power source.

The advantages of using NFC are twofold: Eavesdropping is very difficult and man-in-the-middle attacks are nearly impossible [5]; power consumption on the cell phone is much less than in the case of RFID, because the token receives a significant fraction of the radio wave emitted by the cell phone.

Despite these advantages, the fact that the token is powered, during normal operations, by the cell phone whose power reserve is relatively small compared with that of a typical handheld RFID reader, places severe constraints on its computational power. Under such low-power constraint, even a simple cryptographic operation may take several seconds to complete. The security protocol thus designed must take this limitation into account.

2.2 Security Assumptions

We assume that a passive attacker can record all conversations between the cell phone and the token. He can also replay past conversations to both the token and the cell phone. Although some communication mechanisms such as NFC make eavesdropping very difficult [4], we want to design our system such that other communication protocols can also be used. Furthermore, as will be seen shortly, defense against eavesdropping incurs little additional complexity.

2.3 A Light-Weight Secret-Retrieval Protocol

The basic idea is to store a one-time pad in the token. The pad contains encrypted values of S, using different keys. Every time the mobile phone wants to access the secret S, it requests an encrypted value from the token and discards it afterwards.

More specifically, we assume **S** to be a 128-bit AES [5] key that we use to encrypt everything important on the cell phone. In addition, we use AES to encrypt/decrypt **S**. Denote the encryption and decryption function $\mathbf{E}(x, \boldsymbol{k})$ and $\mathbf{D}(x', \boldsymbol{k})$, respectively, where x is the plain text, x' the cipher text and $\boldsymbol{k}$ the symmetric key.

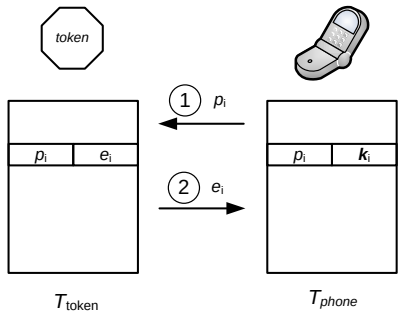

Figure 1: Basic protocol between the cell phone and the token.

The symmetric keys used to encrypt **S** are denoted by $\boldsymbol{k}_1$ ... $\boldsymbol{k}_N$, where N is a constant chosen based on the memory size of the token. We denote the encrypted value of **S** from key $\boldsymbol{k}_i$ by e_i, i.e.

$$e_i = \mathrm{E}(\mathbf{S}, \boldsymbol{k}_i) \text{ and } \mathrm{S} = \mathrm{D}(e_i, \boldsymbol{k}_i)$$

We further associate a 128-bit pseudo-random number, the *validation number* p_i with each $(e_i, \boldsymbol{k}_i)$. Its use will become clear shortly.

On the token, we maintain a hash table T_{token} of N entries; each holds a pair (p_i, e_i)[1]. The index is computed from p_i. Similarly on the cell phone, we maintain a hash table T_{phone} of N entries; each holds a pair $(p_i, \boldsymbol{k}_i)$. Both tables have the same hash function H: $\{0,1\}^{128} \rightarrow \{1..N\}$.

2.3.1 The Preparation Step

In the first step, the user randomly generates N AES keys $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_N$ such that there are no collisions among $H(\boldsymbol{k}_1), \ldots, H(\boldsymbol{k}_N)$. The user then computes $e_i = \mathrm{E}(\mathbf{S}, \boldsymbol{k}_i)$, i=1..N.

Next, the user writes each (p_i, e_i) to table T_{token} at entry $H(\boldsymbol{k}_i)$ and similarly, each $(p_i, \boldsymbol{k}_i)$ to table T_{phone}.

The computation in the preparation step can be done on the user's home PC which also downloads the entries into the token and the cell phone. Alternatively, the cell phone can perform the computational and program the token. To do so, we add an electrical contact to both the token and the mobile device. This

[1] A simple-indexed table also works. However, doing so would require the cellphone to remember which entry on the token contains what. This becomes problematic when we start to deal with multiple tokens and multiple devices.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3936
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3936

programming interface lets the cell phone directly power the token and update its NVRAM. It also allows the cell phone to *refill* the token's table *on demand* (Section 2.3.3).

When using the cell phone to program the token, one can generate the initial secret using the phone. But the plain-text secret must be removed from the phone immediately after the preparation step.

2.3.2 The Secret-Retrieval Step

Whenever the cell phone wants to retrieve the secret from the token, it first selects an unused entry in T_{phone}. Call it (p, $\boldsymbol{k}$). It sends a retrieval command along with validation number p to the token.

Upon receiving p, the token checks the entry (p', e) at index $H(p)$. If the two validation numbers match, i.e., $p'=p$, it sends back e.

After retrieving **S** by decrypting e, the cell phone immediately wipes out the entry (p, $\boldsymbol{k}$). This action ensures that in the event of phone theft from this point on, the thief will be unable to recover S using recorded conversations.

Note that the use of validation numbers offers an inexpensive way for the token to authenticate the cell phone.

2.3.3 Table Refills

The size of the one-time pad T_{token} can be chosen so that the user only needs to reprogram it infrequently. For example, assume a retrieval rate of 50 times per day, a table of 1500 entries requires reprogramming only once a month. It would be more convenient, however, that the cell phone *replenishes* T_{token} on the token as entries are being consumed or when the number of unused entries falls below certain threshold. We call such an action online update (as opposed to offline reprogramming that the user does at home).

The key aspect of a secure online update is that the new entry to be installed in the token, the *refill,* must not be easily deciphered by an eavesdropper. The only time he can obtain the content of the refill is when the entry is being consumed.

However, the relatively simple circuitry on the token, due to its low-power constraint, means that a refill cannot be sent in the air using a strong cryptographic function. Although one can use pseudo-random numbers to scramble the updates, such a scheme either has limited number of refills per entry or is insecure if unlimited refills (per entry) are allowed.

A further complication arises because the mobile device's radio wave is generally insufficient to power the token to update its NVRAM. We therefore use the programming interface for table refills.

To do so, we simply use the plain-text secret from a successful retrieval and the preparation algorithm to generate more entries and download them to the token.

2.4 Security Analysis

We assume the eavesdropper has full knowledge of all (p_i, e_i) pairs that have been consumed by the cell phone. In order to recover the secret **S** from them, he must know the corresponding keys $\boldsymbol{k}_i$. Since the cell phone immediately deletes each $\boldsymbol{k}_i$ after retrieval, the attacker has an extremely small window of time (around a second) to steal the cell phone without disrupting the on-going communication. Note that the use of NFC makes this task nearly impossible. The above analysis shows that the theft of the cell phone alone does not lead to a successful attack.

In light of this, an attacker may want to trick a token to reveal at least one unused e_i before stealing the cell phone. For example, he may stand very close to the victim on a subway and use his smart device to query the victim's token. Unfortunately for the attacker, our use of 128-bit validation numbers renders this attack impractical.

It is also clear that the thief gains nothing by stealing the token alone, because there is no relationship between p_i and $\boldsymbol{k}_i$.

Since the token and cell phone use a shared-secret scheme, man-in-the-middle attacks are infeasible against our system. Such attacks (and eavesdropping) are further rendered impossible if NFC is used as the communication mechanism.

An attacker can still break the scheme by using the stolen cell phone (token) on the token (cell phone) that the user still possesses. This presumes that the victim is unaware of the theft and allows the attacker to get very close to her. If the victim discovers that she has lost either the cell phone or the token, simple actions (described below) are sufficient to forestall a future successful attack.

2.4.1 Recovery from Theft or Losses

If the lost item is the cell phone, the victim can simply flip the kill switch on the token to temporarily disable it. Similarly if the token is lost, she can disable the tangible security feature on the cell phone and resort to the traditional methods of authentication. The fallback mechanism could be a strong (but inconvenient) password. In the event that the user forgets her password, possibly due to inactive use, she can also call the mobile operator to ask them to unlock **S**. In that event, usual security checks such as answering a few personal questions apply.

These fallback measures are temporary. The user can and should rebuild a new secure system by purchasing a new token or getting a replacement cell phone and re-programming the pair. It should be obvious that the new cell phone-token pair is incompatible with the lost phone/token, simply because the way T_{token} and T_{phone} are constructed and used. Note that a new secret **S** will be generated during reprogramming.

2.5 Other Applications

Here we sketch two additional applications for the Tangible Security paradigm.

2.5.1 Online Authentication

The first one is for the token to store one-time passwords for an online account [6] so that the user can securely log in, either via the cell phone or via an untrusted PC, by using her token. To do so, we modify the protocol as follows. Instead of a single secret **S**, we use multiple secrets $\boldsymbol{s}_i$, each being the one-time password. To log onto a secure site X, the user taps the cell phone on the token and retrieves the encrypted secret e_i. The cell phone then decrypts e_i to obtain $\boldsymbol{s}_i$ and uses it to authenticate itself to X.

The preparation (hence refill) step is modified accordingly such that the secrets, i.e., new one-time passwords, are now retrieved from the site X, encrypted and written to the token.

2.5.2 Presence Indicator

A TS token can also be used as a secure presence indicator. For example, the user's PC automatically unlocks when she returns to her desk. Since the PC can sport a more powerful antenna than the cell phone, thereby providing more power to the token, the latter can do more sophisticated computation. In particular, one can use

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3936
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3936

a simple challenge-response protocol here: The PC sends a random number to the token; the latter hashes the message using SHA-1 and a secret key shared with the PC; the validity (or the lack) of the hash convinces the PC whether the legitimate user is present.

3. RELATED WORKS

The closest projects to ours are Zero-Interaction Authentication (ZIA) [3] and Proximity Beacons from NIST [8]. ZIA introduced the concept of a wearable, wireless token to authenticate the user. However, their system assumed a relatively powerful token device, one that contains its own battery and can perform sophisticated computation. The Proximity Beacon project does something very similar to ours, namely using an RFID-like device to authenticate the user to her mobile device. Their system also relies on a relatively powerful beacon and uses Bluetooth as the communication substrate; their protocol relies on PKI. The goal of our work is to make the security token ubiquitous and robust. Therefore we have chosen to use batteryless, RFID-like tokens. Our lightweight protocol is specifically designed for this type of devices which are so inexpensive that they can be thrown away and yet robust enough to be worn in all kinds of circumstances.

One can view Tangible Security as a generalization of smartcards and as a novel application to mobile devices. However, the difference between the Tangible Security and present use of smartcard goes beyond just the form factor. A smartcard is typically certified by a public authority to authenticate the user to a building or website. The TS token and mobile device act as a *private* pair and the user is in full control.

Our system bears some semblance to a biometrics system in that both rely on the presence (and proximity) of a physical trait to authenticate the user. In biometrics, the trait is thought to be unique and unforgeable. The latter assumption is questionable for fingerprint-based systems [1]. A Tangible Security system employs cryptographic algorithms to prevent forgery and therefore provides stronger security guarantees. Furthermore, one need not worry about her finger being cut off or an eyeball taken out by robbers [7]. Unlike biometrics data, our tokens are essentially anonymous entities. Privacy concerns that are associated with biometrics are irrelevant for Tangible Security.

Lastly, iButton[10] is a product that embeds a cryptographic chip in a metal case that has found numerous applications. One can definitely uses it for Tangible Security. The main drawback of an iButton is that it requires electrical contact with the mobile device. In many cases, tokens using contactless, near-field communication is more convenient to the users.

4. CONCLUSIONS

In this paper, we described a new paradigm for maintaining secrets on the cell phone against thefts, by embedding secrets in a separate wearable, physical item which call security tokens. Our system is resilient against casual theft. Furthermore, by allowing the user to associate security with something akin to a physical key and completely doing away with passwords, Tangible Security has the potential to drastically simplify the concept of security for average users. As a result, more people may be inclined to use the strong security measures offered by TS, as opposed to current practice of easy-to-break or no passwords.

5. ACKNOWLEDGMENTS

We thank our colleagues Paul England and Josh Benaloh for interesting ideas on the applications of Tangible Security and on the cryptographic protocol, and Marcus Peinado for his help with the paper.

6. REFERENCES

1. *Spoofing and Anti-Spoofing Measures.* **Schuckers, S.A.C.** 4, 2002, Information Security Technical Report, Vol. 7, pp. 56-62.

2. **Electronic Frontier Foundation (EFF).** Biometrics: Who's Watching You? [Online] September 2003. http://www.eff.org/wp/biometrics-whos-watching-you.

3. *Zero-Interaction Authentication.* **Corner, Mark D. and Noble, Brian D.** Atlanta, Georgia, USA : ACM Press, 2002. MobiCom '02: Proceedings of the 8th Annual International Coference on Mobile Computing and Networking. pp. 1-11.

4. *Personal Area Networks: Near-field Intrabody communication.* **Zimmerman, T. G.** 3&4, 1996, IBM Systems Journal, Vol. 35, pp. 606-617.

5. *Security in Near Field Communication (NFC), Strengths and Weaknesses,".* **Haselsteiner, E. and Breitfuß, K.** Graz : s.n., July 12-14, 2006. Workshop on RFID Security 2006.

6. **NIST.** Announcing the Advanced Encryption Standard (AES). *Federal Information Processing Standards Publication 197.* s.l. : National Institute of Standards and Technology, November 26, 2001.

7. **Florêncio, D. and Herley, C.** *One-Time Password Access without Changing the Server.* Microsoft Research. s.l. : Microsoft Research, 2007. Technical Report. 65.

8. **Jansen, Wayne, Gavrrila, Serban and Korolev, Vlad.** *Proximity Beacons and Mobile Device Authentication: An Overview and Implementation.* NIST. 2005.

9. **Kent, J.** *Malaysia car thieves steal finger.* Kuala Lumpur, Malaysia : BBC News, March 31, 2005.

10. iButton Overview. [Online] Maxim Integrated Products. http://www.maxim-ic.com/products/ibutton/ibuttons/.

11. **Karygiannis, T et al.** *Guidelines for Securing Radio Frequency Identification (RFID) Systems.* NIST. Gaithersburg, MD 20899-8930 : s.n., 2007. 800-98.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2008.3936
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2008.3936