# A Gesture-based Authentication Scheme for Untrusted Public Terminals

*Shwetak N. Patel, Jeffrey S. Pierce, Gregory D. Abowd*
College of Computing & GVU Center
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332-0280
{shwetak, jpierce, abowd}@cc.gatech.edu

**ABSTRACT**
Powerful mobile devices with minimal I/O capabilities increase the likelihood that we will want to annex these devices to I/O resources we encounter in the local environment. This opportunistic annexing will require authentication. We present a sensor-based authentication mechanism for mobile devices that relies on physical possession instead of knowledge to setup the initial connection to a public terminal. Our solution provides a simple mechanism for shaking a device to authenticate with the public infrastructure, making few assumptions about the surrounding infrastructure while also maintaining a reasonable level of security.

**Categories and Subject Descriptors:** K.6.5 [**Security and Protection**]: Authentication; H.5.2 [**User Interfaces**]: Input Devices and strategies (GUI)

**Additional Keywords:** Mobile phone, interaction with gestures, sensors

**INTRODUCTION AND MOTIVATION**
Users are increasingly carrying mobile devices, such as cell phones and personal digital assistants (PDAs), with significant processing power, storage capacity, and network connectivity. The input and output (I/O) capabilities of those devices are, however, extremely limited. They typically feature small displays and offer very slow input methods. While the processing, storage, and network capabilities of those devices are likely to improve drastically, the I/O capabilities are likely to remain limited because the devices must remain small enough that users are willing to carry them. A number of researchers have begun to explore ways to overcome this limitation by leveraging existing I/O devices in the local environment [5, 7, 9]. This approach constitutes an emerging trend in research, and we expect it to continue as mobile devices proliferate. While promising, the approach presents some interesting security and authentication challenges. If users begin to rely on opportunistically annexing [7] local I/O resources, they will need a secure and seamless authentication scheme to establish a connection to a public terminal.


*UIST '04*, October 24–27, 2004, Santa Fe, New Mexico, USA.



**Figure 1:** Prototype of a Motorola i95cl instrumented with a 2-axis accelerometer.

We are particularly interested in exploring the authentication problem of verifying that the user is the one trying to access his device when he initiates annexing from a public terminal by "pulling" information and the desired interface from his mobile device. Supporting this annexation method is desirable because it does not require direct physical interaction with the mobile device. For instance, a user could employ a public terminal to send a lengthy SMS text message or pull up her schedule for colleagues from her cell phone without retrieving it from her pocket or purse. Annexing by pulling, while desirable, requires that users authenticate themselves to their devices to prove that they are the one initiating the connection. This requirement introduces a problem: passwords, the traditional authentication method, are vulnerable to replay attacks. An attacker can instrument the terminal to sniff keystrokes and intercept passwords. The attacker can then later replay that information to gain access to private information on the device.

We developed a simple, accurate technique that prevents replay attacks while requiring little or no direct interaction so that the mobile device can remain in the user's pocket,

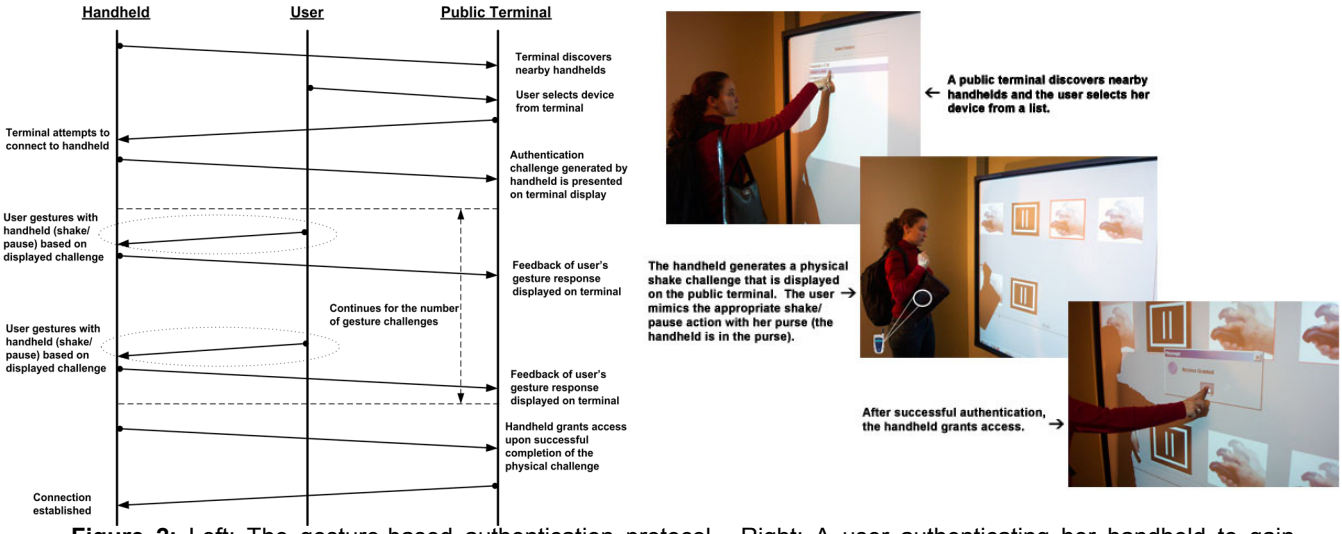


**Figure 2:** Left: The gesture-based authentication protocol. Right: A user authenticating her handheld to gain access to its contents through a public terminal.

purse, or backpack. A user's mobile device displays a gesture sequence on the public terminal, and the user authenticates to his device by shaking it in the required back-and-forth pattern. This scheme requires no secret knowledge for authentication. The user also does not have to retrieve the device because it can recognize the gesture despite an enclosing container such as a pocket or backpack. In addition, the only strict requirement of the public terminal is network connectivity (although short-range wireless facilitates discovery) because the challenge generator and response verifier reside on the mobile device. We believe that our approach is more desirable than relying on physical contact between devices, which would require additional hardware instrumentation of every public terminal, because it minimizes assumptions about the surrounding infrastructure.

Our gesture-based authentication approach provides two additional advantages. First, our approach, while employing a challenge-response structure, is not knowledge-based, relieving the user of the burden of recalling a password or PIN without compromising security. Second, we rely on physical possession of the device, rather than just proximity to it, without requiring that the user retrieve and directly interact with it (another advantage over device contact).

We describe a prototype implementation of this technique on a Motorola iDEN i95cl mobile phone (shown in Figure 1), including the hardware and software implementation, and discuss some of the design issues.

## RELATED WORK

Authentication schemes most commonly rely on possession of either secret knowledge or a physical token. Passwords and PINs, the most common authentication method, rely on demonstrating the possession of secret knowledge, but replay attacks make them unsuitable for authentication through a public terminal. Researchers have attempted to reduce the potential impact of replay attacks by increasing the amount of secret knowledge. Intel's Photographic Authentication [6], for example, relies on a person's ability to recognize personalized content like photographs. Each time a user authenticates, their device presents different personal photographs intermingled with photographs from other sources. The user must then correctly identify which photographs are their own. Though this scheme reasonably addresses replay attacks, it is highly prone to cognitive attacks, where the attacker knows or is acquainted with the person and can easily figure out the correct picture. Probabilistic attacks also pose a problem with this system, because only a few pictures are shown each round. Increasing the number of authentication rounds addresses those problems, but increases the required authentication time and the likelihood of mistakes.

Authenticating by demonstrating possession of a physical token in theory prevents replay attacks because the user must have the token. However, using these tokens can be cumbersome. The most common example, where the user keys a challenge into the token and must submit the generated response, tends to be time-consuming and error prone. Researchers have explored physical token schemes that rely on proximity [1], but some of these systems constantly emit a unique identifier that can be captured and replayed. In addition, just because the user is next to his device does not necessarily mean that he is the only one attempting to access it.

Researchers have also explored authentication by demonstrating physical possession of devices using synchronous physical actions. Rekimoto's Synctap [8] uses distributed synchronous events to create secure connections between devices. For instance, a user may connect to a printer by simultaneously pressing the print button on a PDA and the power button on a printer. Other work has also looked at using synchronous gesture to create logical connections between devices [2, 3, 4], though this work does not directly focus on authentication.

## THE AUTHENTICATION SCHEME

Our system uses a series of shakes and pauses called a gesture sequence as the authenticator. A user authenticates to his device when using some public terminal by asynchronously mimicking a gesture sequence that the mobile device randomly generates and displays on the terminal. Accessing the mobile device involves the following steps (diagrammed in Figure 2):

1. The user selects the handheld (either by entering its address manually or by selecting it from a list of devices detected by short range beaconing) using an interface on the public terminal. Although not implemented, a user could simply shake her mobile device to both cause it to announce its presence and to make the terminal sort the list by recently shaken handhelds.

2. The mobile device randomly generates a gesture challenge and displays it on the terminal. The authentication message asks the user to demonstrate possession of the device by shaking it in a specific sequence (presented as a series of shakes and pauses)

3. The user performs the gestures. As each symbol is correctly authenticated, the visualization grays out that symbol and highlights the next one.

4. Upon successful authentication, the device grants the user access.

Our scheme is a variation of the classic challenge-response protocol. The challenge in our case is for the user trying to access a device to prove that he physically controls it. Note that the user's mobile device generates the gesture sequences (i.e. the challenge), which guards against crafted challenges. The user responds by shaking the mobile device directly, thus guarding against replay attacks.

## THE SYSTEM IMPLEMENTATION

We built our system prototype using a Motorola i95cl cell phone with Nextel data service. While we focused on a cell phone, our approach is suitable for a variety of handheld platforms.

### The Hardware

The system consists of a Motorola iDEN i95cl mobile phone instrumented with a Memsic 2125GL thermal 2-axis accelerometer. Because we cannot rely on knowing the absolute orientation of the device (it might be in a pocket or bag), we need to be able to detect motion along at least two axes. The Memsic 2125 is a low-power high resolution accelerometer that provides both static and dynamic acceleration. It is sensitive enough to detect precise free form rotation and acceleration to about 1 mG (milli Gs) of resolution, providing the flexibility for a variety of gestures. The Memsic provides samples every 10 ms.

The J2ME VM on the mobile phone does not provide a sufficiently accurate high resolution timer needed to sample the accelerometer, so we use a simple Basic Stamp II micro-controller for the signal processing. The micro-controller decodes and filters the modulated signal and sends the acceleration values to the i95cl via the phone's RS-232 (serial) port.

A production version would have the Memsic directly connected to the mobile phone's microprocessor rather than relying on the phone's UART controller. In that case, the micro-controller would no longer be necessary. As seen in Figure 1, we mounted the accelerometer and micro-controller on the inside of the phone's back plate.

### The Software

The gesture recognizer running on the mobile phone is written using J2ME. The recognizer takes the gesture sequence and compares it with the performed gesture. The gesture consists of a sequence of shakes and pauses. Our algorithm quantizes the values coming from the accelerometer into 200 ms samples. Each sample consists of twenty X and twenty Y acceleration values (because of the 10 ms duty cycle). For each X and Y value, we construct a 2-D acceleration vector. The vector represents both magnitude and direction. We use these vectors to determine if it is a valid shake or a pause.

We detect a pause by looking at the lack of changes in direction over a period of time. A shake always produces an alternating g-force because the user moves the phone in opposing directions. Even if the shake motion is more of an alternating rotation motion, it still produces these opposing vectors. The lack of any direction change then indicates that the phone is not being shaken, which is what we consider a pause.

A shake consists of vectors with an alternating change in direction with nearly symmetric magnitudes. We detect a shake by looking at the frequency of each 200 ms sample. Any gesture that produces at least 3 Hz of motion with greater than a net .8 g of magnitude is considered a shake.

## ADDRESSING POTENTIAL ERRORS AND LIMITATIONS

False positives and negatives are the potential errors for our system. These two types of errors are inversely related, i.e. designing the system to reduce one type of error increases the other. We discuss this tradeoff below.

### False Positives

False positives represent a breach in system security and are extremely detrimental to the system. Addressing this issue is paramount. A false positive may occur when a particular motion, such as walking or fidgeting in a chair, matches a gesture sequence. To help determine our potential false positive rate, we conducted an experiment where researchers carried around a version of our prototype that continuously logged acceleration values for a total of a day. We also conducted an experiment where we logged various everyday motions like walking, carrying the phone in a purse, etc.

Figure 3 shows time domain acceleration graphs that represent shaking the phone, walking with the phone in a pocket, and walking with the phone in a purse. General human motion rarely produces the magnitudes needed for the gesture string. Even if they do, the frequencies are much

lower than what a shake would produce. Typical human motion does not produce the frequent and sharp motions that could potentially cause a false positive. We found that even a phone shifting around in a backpack or purse does not produce the high frequencies needed for a gesture. Our experiments resulted in no false positive incidents for the 4 bit gesture string of shake-pause-shake-shake. These results suggest that we can adopt a short gesture string length with little to no impact on the false positive rate, and we may be able to lower the magnitude and frequency thresholds.

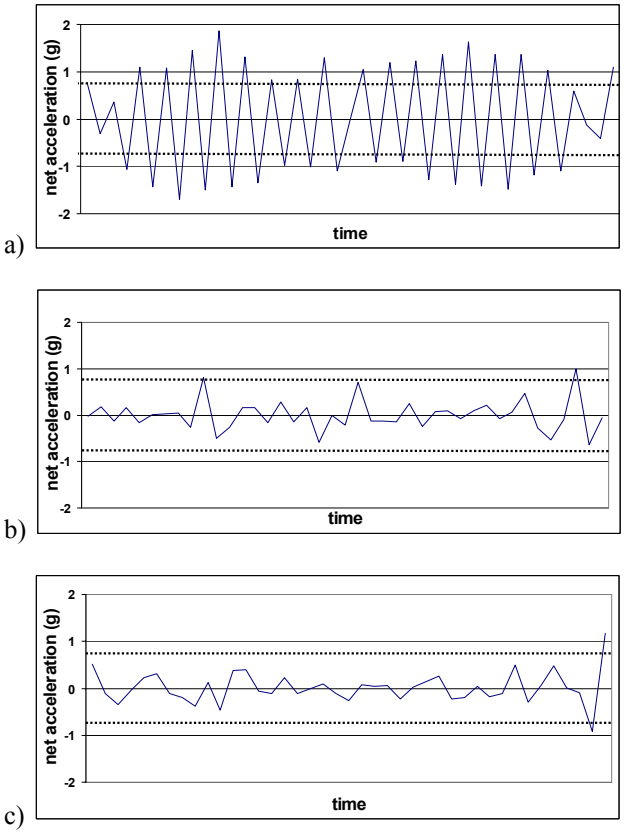


**Figure 3:** One second time domain acceleration graphs of (a) a person shaking the phone to authenticate (b) walking with the phone in a pocket (c) walking with the phone in a purse.

### False Negatives

False negatives represent a rejection for a legitimate user and are not detrimental to privacy and security. However, they frustrate the user and slow down interaction. We found informally that users had few to no false negative errors with a 4-bit gesture string after several minutes of practice. Reducing the magnitude and frequency thresholds, as our initial experiments suggest is possible, should reduce false negatives even further.

## CONCLUSION

Our implemented challenge-response authentication system has several interesting features. The augmentation of the mobile device is relatively straightforward. All of the computational requirements for the authentication protocol are managed by the mobile device. We do not make any assumptions about the public infrastructure beyond network connectivity (and short range wireless if device discovery is desired). From a user's perspective, the gesture-based authentication is simple to understand, yet it provides a reasonable level of security against attacks. We note that our approach concentrates on verifying that the possessor of a device is the one trying to access it; we do not, therefore, address the issue of theft. However, combining our approach with password protection addresses that issue.

## ACKNOWLEDGMENTS

The authors thank Motorola, and in particular Joe Dvorak of iDEN Advancing Technologies Group, for the donation of the iDEN handsets and Nextel service for this research.

## REFERENCES

1. Corner, M. and B. Noble. *Zero-interaction authentication*. MobiCom 2002. September 2002.

2. Fishkin K. P., Kurt Partridge, and Saurav Chatterjee.. *Wireless User Interface Components for Personal Area Networks*. IEEE Pervasive Computing Magazine, October-December 2002. p. 49-55.

3. Hinckley, K. *Synchronous Gestures for Multiple Users and Computers.* In Proceedings of ACM UIST 2003, p. 149-158. Vancouver, Canada, November 2003.

4. Holmquist, L. E., F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen. *SmartIts Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts*. In Proceeding of Ubicomp 2001, p. 116-122. Atlanta, USA, September 2001.

5. Myers, B., R. Miller, J. Nichols, et al. *Using Hand-Held Devices and PCs Together*. Communications of the ACM , v44 n11, November 2001, pages 34-41.

6. Pering T., Murali Sundar, John Light, Roy Want. *Photographic Authentication through Untrusted Terminals.* IEEE Pervasive Computing: Mobile and Ubiquitous Systems. October 2002. p. 30-36.

7. Pierce, J. S., and Heather E. Mahaney. *Opportunistic Annexing for Handheld Devices: Opportunities and Challenges*. In Proceedings of HCIC 2004, Fraser, CO.

8. Rekimoto, J., Yuji Ayatsuka and Michimune Kohno, *SyncTap: An Interaction Technique for Mobile Networking*. In Proceeding of MOBILE HCI 2003, Udine, Italy, September 2003.

9. Want, R., Trevor Pering, Gunner Danneels, Muthu Kumar, Murali Sundar, John Light: *The Personal Server: Changing the Way We Think about Ubiquitous Computing.* Ubicomp 2002. p. 194-209.