Gesture Authentication with Touch Input for Mobile Devices

Yuan Niu and Hao Chen

University of California at Davis,
Davis, California
{niu,hchen}@cs.ucdavis.edu

Abstract. The convergence of our increasing reliance on mobile devices to access online services and the increasing number of online services bring to light usability and security problems in password entry. We propose using gestures with taps to the screen as an alternative to passwords. We test the recall and forgery of gesture authentication and show, using dynamic time warping, that even simple gestures are repeatable by their creators yet hard to forge by attackers when taps are added.

Keywords: mobile authentication, gestures, android, security.

1 Introduction

There are two prevalent trends in the way we use the Internet today. One is the increasing reliance on mobile devices to access online services, and the other is the ever increasing number of online services. Such services, like online banking, email, social networking, etc, all require its users to create and track credentials. The most common form of credentials are passwords and personal identification numbers (PIN). With so many different services, the number of credentials to remember and manage can be overwhelming, prompting users to fallback on less secure measures such as reusing the same password across multiple sites, creating simpler passwords, or relying on password reset [24].

With smartphones, an additional problem is introduced by the limited screen real estate: *password entry.* Password guidelines suggest that "good" passwords contain a mixture of letters, numbers, and symbols. Memorizing and typing these passwords is already frustrating, but the task becomes even more cumbersome on mobile phones' virtual keyboards where letters, numbers, and symbols may appear on separate screens. A survey of 50 smartphone users revealed that password entry is considered more annoying than other limitations to mobile devices, such as a small screen or poor signal. In addition, 56% of theses users had typed a password incorrectly at least once every ten times [10].

Smartphones using virtual keyboards take input by reading taps to the screen. Often, the user sees a larger or highlighted version of the key being tapped. Password entry fields have also been modified to display the most recently typed letter for a short interval. Soft keyboards often place letters, numbers, and symbols

R. Prasad et al. (Eds.): MOBISEC 2011, LNICST 94, pp. 13–24, 2012.

into separate screens because of space limitations. When they display letters and symbols or numbers concurrently, only a subset of alternate characters are available by holding down the tapped letter for a longer interval to indicate choice of the alternate character. The extra feedback provided helps the user verify her intended keystrokes, but it also helps a nearby third party to observe the user's keystrokes. The extra taps needed to switch between letter, number, and symbol screens also discloses the order of letters, numbers, and symbols within the password.

We propose an alternative to passwords and PINs: *gestures* with touch input. We define gestures as a series of small movements involving motion wrists and forearms while the hand holds the phone. The sheer range of motion and subtleties of force and speed will provide variations between users, even though there some motions, e.g. circles, that may be popular components in gestures. For even more variation, the user can tap the screen with her thumb while she holds the phone to perform a gesture.

2 Gestures

Phishing and human behavior studies demonstrate that humans are creatures of habit [11]. At an even more basic level, muscle memory enables us to learn and perform motor skills quickly and repeatedly without much conscious effort.

Gestures capture a biometric quantity of muscle memory and physical characteristics of the specific user. For the purposes of authentication, we care only about gesture *recall* rather than gestures *identification*.

2.1 Usability Benefits

Gestures provide a faster and more convenient method of authentication compared to a complex password. Studies [2, 24] show that passwords that are hard to guess are (not surprisingly) also hard to remember. Gestures, on the other hand, rely on motor skills. Touch typing, brushing your teeth, or signing a signature are examples of fine motor skills into which we put little conscious thought, but are activities we can replicate accurately and precisely.

A few use cases for gesture authentication are described below.

Device Unlock and Second Factor. The most obvious application of gesture authentication is replacing existing phone unlock mechanisms involving pins or gestures drawn on the touchscreen. Gestures can also be used to supplement existing authentication mechanisms where additional proof is needed.

Password Management. Because gestures should be easy to remember, it is possible for users to record one gesture for every unique password. However, remembering which password corresponds to which gesture could get complicated as the number of unique passwords increase. One increasingly common method of dealing with multiple passwords is to rely on a password manager. In this scenario, the gesture will serve as the master password for unlocking all other passwords.

Alternatively, because storing passwords on the phone is risky in case the device is stolen, we could use gestures to access credentials stored remotely[1] or as alternate OpenID credentials. Another advantage of this approach is that it eases the processing burden of running gesture recognition on the phone.

2.2 Security Benefits

A gesture based authentication system would make it more difficult for a shoulder surfer to replay the password, even if he observes the entire gesture. Subtleties like force, speed, flexibility, pressure, and individual anatomical differences would prevent the casual observer from repeating the gesture well enough to authenticate successfully. Furthermore, there are taps to the screen which may be hidden as the user moves the phone. These types of gestures will not require users to look at the screen, so they can gain increased privacy by choosing gestures that can be performed by holding the phone away from the view of onlookers, such as under a table.

Entropy. A text-based password's entropy depends heavily on its length. The entropy per character is $log_2(N)$ bits where N is the size of the pool from which the characters are selected. When all 94 printable ASCII are in the pool, the theoretical entropy per character is 6.55 bits for a random password.

Intuitively, gestures contain more entropy than text passwords because users are not limited to printable characters. Quantitatively, the theoretical entropy provided by our gestures can be measured using the following factors:

It is possible to record with a sample rate between $100hz$ to $120hz$ on some smartphones. The accelerometer in the earliest Android phone can record each axis with $8-bit$ precision, and this precision will only improve as accelerometers improve. For instance, $13-bit$ precision recording is already possible on existing accelerometers. Considering orientation gives us 8.49, 8.49, and 7.49 bits of data per sample for the x, y, and z axis, respectively. Using a conservative sampling rate of $60hz$ for both acceleration and orientation, we get 2908.2 bits of entropy per second. If we consider just acceleration, one second of gesture data contains 1440 bits of entropy. Adding touches to the screen further complicate things, as there are any number of taps the user may choose to perform concurrently while performing the gesture.

Shoulder Surfing. Shoulder surfers observing typed passwords on smartphones may have an easier time when the device has a touchscreen, such as an iPhone or Nexus One by taking advantage of visual feedback to the user as they select letters or switch between numbers, symbols, and letters. Gestures are more robust against shoulder surfers, even those with video cameras. It is hard to estimate the force and timing of gestures correctly solely with brute force. Furthermore, the standard recording frame rate for HD video is $29fps$ or $30fps$ in devices like the iPhone, whereas we can sample gesture data at a conservative rate

[1] We assume the existence of secure storage for such credentials.

of $60fps$. Tapping lightly on the screen with the thumb while moving the phone will further frustrate the attacker, since the attacker would need to identify each tap and its timing.

3 Experimental Design

We conducted two sets of users studies to determine user recall and attacker success rates.

The stages to our experiments are: 1. a) A long term user study with only gestures and b) a follow up attack study and 2. a) an attack study on the two simplest gestures from stage 1 with touches to the screen added and then b) a long term user study on recall. We recruited users from the computer science department at UC Davis because we did not believe that technical expertise would affect gesture choice or performance. We relied on a 3rd party application, Contextlogger [12], which gave us only accelerometer data, for Stage 1, but found that it was insufficient for our needs in Stage 2, at which point we developed a custom Android application.

Stage 1A: Gesture Only. Each subject was asked to perform the gesture of his choice several times a day, at least 5 iterations at a time, over at least a week. Most subjects participated for at least one month. We had 5 participants who performed 7 gestures. The seven gestures are: 2 are *alphas*, *the Nike 'swoosh'*, *alpha followed by a circle*, *first letter of the Thai alphabet*, *the subject's initials*, and *the subject's signature.*

Stage 1B: Attacks. We assume an attacker with access to video. The attacker is limited to 5 attempts at gesture replication before the victim's account is locked out. Subjects from Stage 1A were asked if they would allow their gesture to be video recorded. We recorded a total of 6 from 4 participants from a frontal angle and a back angle, which we felt gave the most direct views of the gesture performance.

Participants from Stage 1A were asked to play the role of the attacker with the incentive of a small prize awarded to the most convincing attacker. 6 subjects volunteered. Subjects were allowed as much time as they needed to perform the attacks, and there was no set rules as to how attacks must proceed.

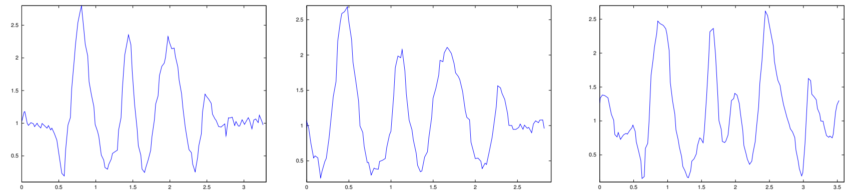
Stage 2A: Adding Taps. We asked two "victims" from Stage 1B with the simplest gestures, the alpha and the swoosh, to record new videos showing the same gesture they used in Stage 1A, this time with taps added. These gestures were recorded using an Android application that we developed. We recruited 9 subjects to act as attackers. They were asked to view the two victim videos, again with no set rules as to how the attacks must proceed. Attackers were told that in addition to the movement, taps had been added to each gesture. These participants were given the videos to study and when they were ready, asked to record at least ten attempts for each victim gesture.

We chose to perform the attack study before starting a second longterm study to test our hypothesis that adding taps would make very simple gestures harder to imitate.

Part 2B: Extended Study with Taps. We recruited subjects for a long term study, once again, and had 6 users. We asked these subjects to provide at least two weeks of data. The six gestures performed were: the *Nike swoosh* with 5 taps, a *parry-thrust* with 2 taps, *signature* with 4 taps, *two loops* with 3 taps, a *back and forth* motion with 3 taps, and *initials* with 2 taps.

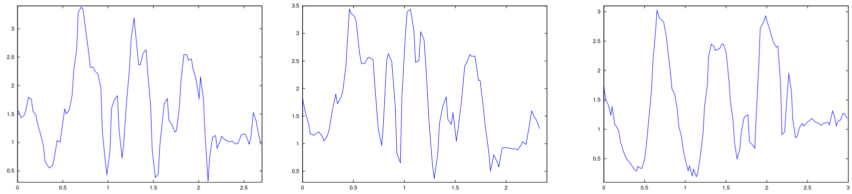
4 Analysis

We present two examples from an earlier feasibility study not discussed in this paper - a simple compound (Fig 1) consisting of an alpha connected to a circle, and a complex gesture (Fig 2) consisting of a Chinese character written in cursive to demonstrate what forgery and recall might look like.



(a) Successful Match Example (b) Forgery Attempt

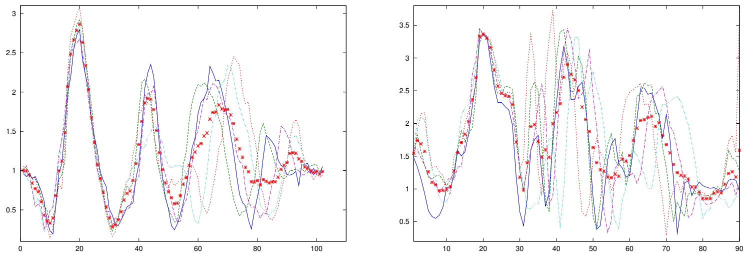
Fig. 1. Simple Gesture



(a) Successful Match Example (b) Forgery Attempt

Fig. 2. Complex Gesture

As seen in figure 3, multiple trials of the same gesture generally have the same shape, but the timing varies. We used DTW to compute the similarity between a victim's gestures, and the similarity between the victim's gestures and those of his attackers.



(a) Simple Gesture (b) Complex Gesture

Fig. 3. Overlays of Gesture Data, adjusted to match for the first feature

4.1 Dynamic Time Warping

Dynamic time warping (DTW) is a dynamic programming sequence alignment algorithm that matches one time series onto a reference time series. The result is a monotonically increasing path and a cumulative matching cost. If two sequences are identical, then the path is a perfect diagonal and the cumulative matching cost is 0. When two sequences differ slightly, the cumulative matching cost is the sum of the distances between the two points matched. DTW with Euclidean distance matching allows us to measure the similarity of two gestures.

4.2 Methodology

In our experiments, we use an existing implementation of DTW [8]. To computer the score for similarity between a gesture and the reference gesture, we do the following:

Let n be the number of repetitions we have for an instance $g \subset G$ of a gesture G, and $r \subset G$ is the set of repeated gestures currently used as reference set. The reference set can be considered the stored "password" and is generated by the user during the initial training period. After this, users only need to perform their gesture once to authenticate. The score, $\varTheta$, is then:

$$\varTheta = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{n-1} \sum_{j=1}^{n-1} dtw(r_i, g_j) \tag{1}$$

For victim and reference sets, let m be the number of attempts $f \subset F$ we have for a gesture G, and $r \subset G$ is the instance of the victim's gesture currently used as reference. The score, $\varTheta$, is then:

$$\varTheta = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} dtw(r_i, f_j) \tag{2}$$

We use Equation 1 to generate an average score, which we then used to compute two cutoff values: 1) mean of the repetition scores + standard deviation

2) mean of the repetition scores + half the standard deviation.

The cufoff value determines whether the gesture in question scores low enough (i.e. is similar enough) to the reference gestures.

5 Results

Our extended study showed promising results in that most gestures, except the very simplest, were repeatable by their creators yet difficult to forge by attackers.

5.1 Stage 1: Gesture with No Taps

Gesture Variation. For each user, we gathered at least one week of data (at least 35 repetitions of the same gestures). Our goal in this study was to find a scoring system that is lenient enough to give a legitimate user some leeway in gesture variation and shift over time, and at the same time makes it difficult for an attacker to succeed. We do not address gesture shift over time in our analysis, although we did notice that as time went on, the average DTW scores of later gestures were decreasing and stabilizing.

In these studies, the false negative rate is $1 - rate\ of\ success$ for repetitions, and the false positive rate is the rate of success for attacks. Lower scores indicate greater gesture similarity, because the score is a reflection of the total matching cost between two sequences.

Using Reference Set. Simply using a globally established average DTW value skews the average with initially inconsistent performance. Choosing just one gesture runs into the danger of selecting an outlier that will skew the false negative and false positive rates as well. We randomly sampled 5 files from all gestures to act as our initial, acceptable gesture set and generated a set of $\binom{5}{2}$ DTW scores for each possible pair of gestures. From this set, we tested two possible upper limits to the scores: 1) Average score + Stdev(set of reference scores) and 2) Average score + Stdev(set of reference scores)/2. We calculated average score using Equation 1 where $n = 5$. We adjusted the maximum acceptable score by taking the average of half of the remaining gestures to balance the outliers that were possibly in the initial gesture reference set. We use the same adjusted scores to evaluate the success of attacks. Results are shown in Table 1. This simulates the case where a user has been using gestures to authenticate for some time.

Using adjusted scores, we see that it improves the repetition success rates, but again, skew from outliers is significant enough to improve the attacker's success rates when the gesture is overly simple. A slightly more complicated compound gesture, `alpha+circle`, maintained relatively low false positive and low false negative rates.

Table 1. Repetition Success Rates with Reference Set (1) and Adjustments (2)

	Cutoff 1	Cutoff 2	Attack Cutoff 1	Attack Cutoff 2
alpha1	0.93	0.88	0.57	0.26
alpha2	1	1	0.61	0.26
nike	1	0.98	1.0	0.75
alpha+circle	0.94	0.9	0.06	0.01
initials	0.95	0.93	0.1	0
signature	0.97	0.96	0.15	0.02
Thai letter	0.98	0.93	NA	NA

`alpha and circle` was successfully performed by just one attacker. Again, the majority of successful attacks on `alpha2` came from the subject who performed `alpha1`.

5.2 Stage 2: Gesture with Taps

Attack Study. To judge success on gesture forgery with taps, we examined the taps data first, as we expected the gesture data to be similar to the attacks described previously. Attackers were often unsure of the number of taps and varied the number as they tried to perform the gestures. At first glance, it seems as though attackers are fairly successful (see Table 2) at guessing the number of taps. However, when we use DTW to score the similarity of timing between attackers and victims, none of the attackers' scores fall into the acceptable range[2].

Table 2. Tapping Count Success Rates for Attacks. The Alpha received 130 attempts, and the Swoosh received 118.

Attacker	Alpha (2 taps)	Swoosh (5 taps)
1	0.21	0.17
2	0.93	0.85
3	0.23	0
4	0	0.70
5	0	1
6	0.57	0.02
7	0.81	0
8	0.08	1
9	0.44	1

[2] Between 0 and (Victim's average score + stdev(victim's score)).

Extended Study. We used the same methodology from stage 1 to evaluate gesture performance. A gesture passes if both the timing and number of taps are correct, and then we evaluate the accelerometer data. We show rates of success in table 3. The results demonstrate that gesture movements are clearly replicable, even with the additional task of memorizing taps, by their respective owner. In some cases, the taps seem to be more difficult to remember than the gesture - for example, the *back-forth* movements were performed with high success rates, but the taps associated with the gesture had the lowest success rates of all gestures. The cutoff rates are adjustable, however, so for services that do not require high security, the more lenient rate could be used for authentication. Conversely, for services that require more security, an even stricter cutoff rate could be enforced.

Table 3. Success Rates of Tasks for Gesturing with Taps

	Touch Count	Touch Timing Cutoff 1	Touch Timing Cutoff 2	Accelerometer Cutoff 1	Accelerometer Cutoff 2
back-forth	0.94	0.87	0.82	0.96	0.89
initials	0.92	1.00	0.97	0.90	0.86
loops	0.97	1.00	0.84	0.92	0.74
parry thrust	0.97	0.98	0.97	1.00	0.79
signature	0.89	0.93	0.79	0.93	0.79
swoosh	0.96	0.94	0.90	0.95	0.88

6 Related Work

Traditionally, authentication factors for computer systems are classified as one of the following: something you know, something you have, or something you are. Biometrics, the measurement of physical characteristics or behavioral traits to identify an individual, are a way to provide evidence for that last factor. Factors such as fingerprints, retinal patterns, or voice patterns have been well studied and evaluated. Jain et al provide a survey of biometric methods [9].

Tapping a rhythm in place of password entry was proposed by Wobbrock [23]. The TapSong user studies showed differences in people's tapping of the same song, and that eavesdropping was difficult because the attacker had no sense of what song was being "played."

Many consumer electronics devices today contain 3-axis accelerometers to measure the positioning and motion of the device. When the device is manipulated by the user, the motions can be a biometric, capturing unique biomechanical traits of the user. Accelerometers have been used to capture gaits [6,16], arm-sweeps [7], and hand gestures [4,14,15]. The accelerometers provide a time-series of measurements in the 3-axes while the motion is performed.

Ravi et al [21] used accelerometer data to recognize 8 activities, such as brushing teeth, running, going up or down stairs, etc using a single accelerometer attached near the pelvic region. They used FFT to extract the energy associated with an activity and used the Weka toolkit [1] to classify the activities. They discussed and tried various classifiers, finally finding that plurality voting yielded the best results.

Existing signature and gesture recognition systems make use of dynamic time warping techniques to score the similarity of inputs [13, 14, 17]. Template adaptation changes the expected sequences used in matching based on age of the template and its similarity to the current input. If the current input is a match, the old template is discarded and the current input becomes the new template. We apply these techniques to address personally unique gesture recognition.

Pylvänäinen [20] used Hidden Markov Models (HMM) to build a recognizer for accelerometer recorded gestures without using feature extractors. However, the gestures described, a "circle or an upward line" are too basic to be used for authentication. They were able to determine that sampling at $30Hz$ was sufficient enough for maximum accuracy.

Similary, Schömer et al [22] worked on gesture recognition and training using the Wiimote. The Wiigee project [19] deals with gesture recognition using a left to right HMM.

Patel et al [18] use gestures to authenticate untrusted public terminals by displaying a pattern that must be replicated on a user's cellphone through gesturing.

Farella et al [5] tested four distinct gestures and found that it is possible to distinguish these gestures in small groups. Chong and Marsden [3] tested the usage of gestures as passwords by creating a limited "alphabet" from which all gesture passwords would be formed. Our system removes the restriction of any alphabet and allows users to choose whatever motions they want as their gesture. Moreover, we are the first, to the best of our knowledge, to combine screen taps with movements to enhance the security of gestures.

Czeskis et al [4] demonstrated that users are able replicate simple gestures accurately in order to activate RFIDs. Again, a key difference in their goals and ours is the reproducibility of gestures: We want only one person to successfully be able reproduce his own gesture.

7 Conclusions and Future Work

We proposed gesture-based authentication on mobile devices. We evaluated several scoring and decision methods using dynamic time warping. We conducted user studies to examine the consistency of repeating one's own gestures over time and the difficulty of emulating others' gestures.

We discovered that the more complicated gestures, unsurprisingly, have low false positive rates and low false negative rates. We also found that we can enhance the security of the simplest gestures by requiring the user to tap the screen during the gesture, because the attackers were unable to observe or replicate correctly the number of taps or the timing of the taps.

Our examination of the gesture with tap data revealed an unexpected benefit: by examining the taps data first, we can sometimes avoid the more computationally expensive task of comparing accelerometer data. Adding taps effectively makes the gesture a two part "password" for attackers, while it remains one integrated motion for the average user.

Gesturing to authenticate can protect users from shoulder surfers and malicious bystanders who may observe the process of password entry. To prevent attackers from emulating gestures, the user should avoid overtly simple gestures, or combine these simple gestures with tapping.

7.1 Future Work

Based on feedback from stage 2 of the user study, one additional step could be implemented before uploading the data: asking the participant whether they feel that gesture could pass the authentication process with a simple "Yes" or "No" popup. We also need to study the effect of feedback to the user in the form of "Pass" or "Fail" when they upload each gesture attempt.

References

1. Weka machine learning project, `http://www.cs.waikato.ac.nz/~ml/weka`
2. Adams, A., Sasse, M.A.: Users are not the enemy. Commun. ACM 42(12), 40–46 (1999)
3. Chong, M.K., Marsden, G.: Exploring the Use of Discrete Gestures for Authentication. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009. LNCS, vol. 5727, pp. 205–213. Springer, Heidelberg (2009)
4. Czeskis, A., Koscher, K., Smith, J., Kohno, T.: Rfids and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In: CCS 2008: Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 479–490. ACM, New York (2008)
5. Farella, E., O'Modhrain, S., Benini, L., Riccó, B.: Gesture Signature for Ambient Intelligence Applications: A Feasibility Study. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVASIVE 2006. LNCS, vol. 3968, pp. 288–304. Springer, Heidelberg (2006)
6. Gafurov, D., Helkala, K., Søndrol, T.: Biometric gait authentication using accelerometer sensor. Journal of Computers 1(7) (2006)
7. Gafurov, D., Snekkkenes, E.: Arm swing as a weak biometric for unobtrusive user authentication. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1080–1087 (2008)
8. Giorgino, T.: Computing and visualizing dynamic time warping alignments in r: The dtw package. Journal of Statistical Software 31(7), 1–24 (2009)
9. Jain, A., Bolle, R., Pankanti, S. (eds.): Biometrics: Personal Identification in Networked Society. Kluwer Academic Publishers (1999)
10. Jakobsson, M., Shi, E., Golle, P., Chow, R.: Implicit authentication for mobile devices. In: 4th USENIX Workshop on Hot Topics in Security, HotSec 2009 (2009)

11. Karlof, C., Goto, B., Wagner, D.: Conditioned-safe ceremonies and a user study of an application to web authentication. In: Sixteenth Annual Network and Distributed Systems Security Symposium (2009)
12. Kunze, K.: Context logger, `http://contextlogger.blogspot.com/`
13. Lei, H., Govindaraju, V.: A comparative study on the consistency of features in on-line signature verification. Pattern Recogn. Lett. 26(15), 2483–2489 (2005)
14. Liu, J., Wang, Z., Zhong, L., Wickramasuriya, J., Vasudevan, V.: uWave: Accelerometer-based personalized gesture recognition and its applications. In: IEEE Int. Conf. Pervasive Computing and Communication (PerCom) (March 2009)
15. Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V.: User evaluation of lightweight user authentication with a single tri-axis accelerometer. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2009, pp. 15:1–15:10. ACM, New York (2009)
16. Mäntyjärvi, J., Lindholm, M., Vildjiounaite, E., Mäkelä, S., Ailisto, H.: Identifying users of portable devices from gait pattern with accelerometers. In: Proceedings of IEEE Interational Conference on Acoustics, Speech, and Signal Processing, ICASSP 2005 (2005)
17. Nalwa, V.S.: Automatic On-line Signature Verification. In: Chin, R., Pong, T.-C. (eds.) ACCV 1998. LNCS, vol. 1351, pp. 10–15. Springer, Heidelberg (1997)
18. Patel, S., Pierce, J., Abowd, G.: A gesture-based authentication scheme for untrusted public terminals. In: ACM Symposium on User Interface Software and Technology, pp. 157–160. ACM Press (2004)
19. Poppinga, B., Schlömer, T.: wiigee: A Java based gesture recognition library for the wii remote, `http://wiigee.sourceforge.net/`
20. Pylvänäinen, T.: Accelerometer Based Gesture Recognition Using Continuous HMMs. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005, Part I. LNCS, vol. 3522, pp. 639–646. Springer, Heidelberg (2005)
21. Ravi, N., Dandekar, N., Mysore, P., Littman, M.: Activity recognition from accelerometer data. In: American Association for Artificial Intelligence (2005)
22. Schlömer, T., Poppinga, B., Henze, N., Boll, S.: Gesture recognition with a wii controller. In: TEI 2008: Proceedings of the 2nd International Conference on Tangible and Embedded Interaction, pp. 11–14. ACM, New York (2008)
23. Wobbrock, J.O.: Tapsongs: tapping rhythm-based passwords on a single binary sensor. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, UIST 2009, pp. 93–96. ACM, New York (2009)
24. Yan, J., Blackwell, A., Anderson, R., Grant, A.: Password memorability and security: Empirical results. IEEE Security and Privacy 2(5), 25–31 (2004)