UbiPhone: The Phone You Can't Forget, An Authentication Investigation

Taurin Tan-atichat
University of California, San Diego
Department of Computer Science and Engineering

Abstract

Many individuals have become dependent upon their cell phones for communication but do not have the means to recover important information if data were to be lost from their phones. A proposed solution is the UbiPhone system, a ubiquitous personal phone book that can be accessed from any cell phone. However, a secure, reliable, and easy to use authentication seem is needed in order to provide such a service. This paper examines 3 different authentication schemes: username/password, ShotCodes (circular bar codes), and face recognition.

Introduction

Cell phones have become the killer application of real time communication. People constantly have their phones with them, turned on, and with service no matter where they go. It is the single most effective way to contact a person immediately. Huge inconveniences can result from the disconnection from the phone network. Physical environmental obstacles and network congestion are temporary detachments, easy to recover from, and difficult to entirely prevent. The loss of phone numbers is a more permanent problem, can be difficult to recover from, and is relatively easy to prevent with the proper infrastructure.

The UbiPhone system is a proactive solution to the loss of data on a cell phone. Personal information from a cell phone is stored on a central server before a possible loss. This information must be made easily available to a user on the occasion when the user needs to retrieve the information. However, this information may be sensitive so it should be made difficult for others to access. So there is a need for a form of authentication that leverages between these two objectives. This paper describes some biometric methods considered and 3 different methods implemented.

Related Work

The problem of cell phone loss is not new and is in fact growing [1]. Some carriers, such as Verizon, do provide backup services but require a fee which is unattractive and poses as a roadblock to the majority of customers [2]. Additionally, these services do not allow easy retrieval, as you must call in to report your phone lost, stolen, or otherwise defunct. Retrieval should be made easy and only require a user to authenticate himself on another phone.

There have been some authentication systems that incorporate the use of cell phones such as StrikeForce's system which lets the cell phone act as an authentication token when logging into the main system [3]. A phone call is placed to a person's phone to verify his identify. However, very few systems use a cell phone directly to authenticate a user.

Background

Username/password systems are the de facto standard authentication scheme used on computers. Everything from login, to website surfing, to installing new software requires a username/password. However, it is not ideal on a cell phone due to the cumbersomeness of typing alphabetic letters on a numeric keypad. Additionally, simple passwords are easy to be guessed and compromised while complex passwords are easy to forget.

Another omnipresent authentication scheme is the identification card. You are asked to see it when you go through security at the airport and when you pay for something at the store with a credit card. Unfortunately, a cell phone can't easily verify an identification card but it can verify a ShotCode, *Figure 1*. The ShotCode is essentially a circular bar code that a camera phone can easily interpret and verify. Of course loss of a ShotCode can be a hindrance.

Lastly, face recognition is just a couple levels higher than ShotCodes. The pure simplicity of it is a wonderful characteristic since there is no need to carry anything or remember anything. People's faces do not usually change very often throughout the years which ensures long term identification. However, algorithms to accurately recognition a person's face are not very mature.

These three authentication methods will be further discussed in detail but some other authentication schemes that were considered but declined for implementation included other biometrics such as voice, fingerprint, and iris mostly due to the lack of peripherals on cell phones that allow for quality capture.

Figure 1. A ShotCode

Implementation

The UbiPhone system is run as a web service for cell phones. It transmits data over HTTP and uses the Wireless Markup Language (WML) to display content inside Wireless Application Protocol (WAP) browser. It is similar to HTML so integrating a username/password was quite easy by using a standard form.

ShotCodes are a third party service that run as a Java 2 Micro Edition (J2ME) application in which you simply use the phone's camera to take a picture of the ShotCode. Once the phone recognizes the pattern, it opens up the phone's WAP browser and directs the browser to fetch a preprogrammed URL. Once again, this easily integrated into the UbiPhone System by assigning each user one ShotCode that directly places a user into the UbiPhone System when used.

Face recognition was the most difficult authentication scheme to implement. Difficulties included having to send the face image over the slow network, being limited to just one training set for each individual, not having control of the lighting in scenes, not having control of the framing of faces, being limited to low resolution images due to memory constraints and suboptimal algorithms, and server administration issues.

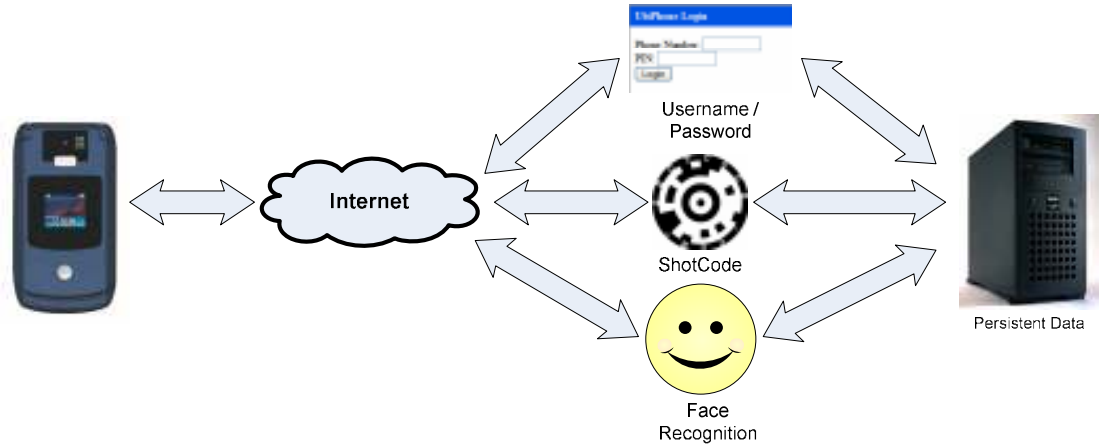

Figure 2. UbiPhone system with authentication methods

Some highlights of the problems include the inability to do an HTTP POST request from a cell phone with an embedded binary file. The binary file must be converted into ASCII-printable characters, which increases the file size that is already taking several seconds to be transmitted over a dial-up speed wireless connection. After getting the image across, it must be decoded back into binary form, resized, read into the Java environment, converted from color to grayscale, and converted into Matrix form.

The Eigenfaces algorithm was used as the primary face recognition algorithm [4]. This was chosen because it is one of the most well known algorithms, with adequate learning resources, and low complexity. The only significant difficulty with it was the computation of the eigenvectors of the covariance matrix of the adjusted data since the size of the matrix was $m*n$ X $m*n$ where m was the width of the image and n was the height in pixels. Without resizing the image, the memory consumption was outrageously high, and even with moderate resizing CPU time was excessively long.

Analysis

The username/password authentication scheme was not as bad as initially anticipated. The problems of typing alphabetic letters on a numeric keypad were overcome by using a numeric-only username and PIN. An easy choice was to use the user's phone number as identification. The amount of time required to type in all information was approximately 5 seconds. One pitfall however was that users would have to remember the URL of the UbiPhone system and type that into the WAP browser. This took approximately one minute.

The ShotCode authentication scheme also made an impressive showing. Benefits include not having to remember the URL, having fast (approximately 5 seconds) and reliable third party service and redirection to the UbiPhone system. The recognition accuracy rate was impressive and never encountered a false positive. The algorithm was tolerant to different lighting conditions, different rotation, but suffered from angled tilts and bad framing. The problem of having to always carry the ShotCode around was

reduced by making an identification card printable in the form of a business card. Such an item could be carried around in a wallet with ease.

Face recognition was clearly a preferred method of authentication however there were many difficulties involved in getting it to function correctly. Even after getting the system up and running, many problems bogged down the effectiveness of face recognition. Some examples include poor lighting conditions, inability to access the phone's internal camera to allow for better framing, the phone's security manager that inconveniently asked for permission to record images using the camera, the amount of time to receive a response from the server, the unreliable wireless network, and the ease of faking a face with a photo of a face. It's actually quite surprising that face recognition is easier to fake than the ShotCode since many images of faces are available to try to use, but obtaining a valid ShotCode is quite difficult.

Formal tests under real world conditions were not done but could be predicted to be quite poor. Also, registration time was approximately 1 minute and recognition time was approximately 15 seconds. However, a test with the Yale Face Database showed an accuracy rate of 14/15 individuals identified correctly. This shows that the algorithm itself was not at fault for the poor results but it was the combination of the whole chain of events that must occur for face recognition to function, with much emphasis placed on uncontrolled image lighting and background.

User Feedback

A survey was given out to determine comfort with using one's face for authentication versus other more traditional types of authentication. People preferred to use face recognition over ShotCodes. Although no follow-up questions were asked as to why they felt that way, one possibility is that people are often used to having their photograph taken while they are not used to photographing something else in order to be granted entry into a secure location.

Future Work

It was shown that face recognition has a long way to go before it can be feasible on a cell phone as an authentication scheme. Some improvements include preprocessing images, use face detection, reduce background noise, normalize lighting, allow video to be used for training new users to obtain multiple images without taking multiple pictures, use a more efficient base64 binary to ASCII encoding, and use Singular Value Decomposition (SVD) in Eigenfaces computation to improve speed and resolution.

Conclusion

More work is definitely needed in authentication of cell phones by themselves with the up and coming large number of applications and features being added into them in order to mitigate issues that might come up in the case of a phone being stolen. For the time being numeric username and PIN as well as ShotCodes will suffice.

Acknowledgements

This work would not have been possible without the dedication and support of Brian Robbins, Chung Yen, Ben Laxton, David Kriegman, and Bill Griswold.

References

[1]http://www.time.com/time/europe/magazine/article/0,13005,901020311-214207,00.html

[2]http://getitnow.vzwshop.com/search_going.aspx?id=search_going&appSearchParentCategoryId=246&appSearchText=backup&bhcp=1

[3]http://www.sftnj.com/news/pdf/ProtectID-8-22-05.pdf

[4]M. Turk and A. Pentland, “Face recognition using eigenfaces”, in Proc. IEEE Conf. on Comp. Vision and Patt. Recog., 1991, pp. 586-591.

[5]http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[6]Sing Li, and Jonathan Knudsen, "Beginning J2ME", Apress, Berkeley, CA, 2005.

[7]W. Jason Gilmore, "Beginning PHP and MySQL 5: From Novice to Professional", Apress, Berkeley, CA, 2006.