# Context-Aware Authentication Framework

Diwakar Goel, Eisha Kher, Shriya Joag, Veda Mujumdar,
Martin Griss, and Anind K. Dey

Carnegie Mellon Silicon Valley,
NASA Research Park, Bldg. 23, Moffet Field, CA 94035
{diwakarg,ekher,sjoag,vmujumda}@sv.cmu.edu,
martin.griss@sv.cmu.edu, anind@cs.cmu.edu

**Abstract.** We present an extensible context-aware authentication framework which can adapt to the contextual information available in a smart environment. Having confidence in a user's identity and other contextual information is critical to the successful adoption of future mobile, context-aware services. This authentication framework can provide a standard base for the development of context-aware services, particularly while the user is mobile. Our implementation of the framework enhances usability during authentication by replacing the need for users to remember and enter their password with the act of a simple gesture. We discuss our architecture, implementation and policies and illustrate how they support usable authentication, using lightweight tagging and simple context from a smart environment.

**General Terms:** Barcode Management System, relational database, access policies.

**Keywords:** Barcode, context-aware computing, Wi-Fi signatures, location-based authentication, mobile computing, soft sensors, authentication.

## 1 Introduction

Context is the set of facts or circumstances that surround a situation or an event. [Dey 2001] The relative increase in the use of mobile systems as compared to desktop systems presents the opportunity to retrieve the dynamic context of mobile users and have it be ubiquitously available to various services. For example, in an increasing number of mobile applications, it is important to know the current location of a user. As context-aware systems become more pervasive, they have started incorporating a wide array of contextual cues pertaining to a user. Contextual cues help the computer system to better understand the state of the user and make informed decisions about any services requested by the user. The benefit of using context-aware systems is that a user does not have to explicitly provide information about her state, but this information can be sensed and used by the system nonetheless. This increase in the use of context-aware systems and smart environments suggests the potential value of a framework that would allow such systems to make a correct decision about a user's identity. Our implementation aims to address this need and presents an extensible framework for context-based authentication systems. The approach presented in this

S. Ystad et al. (Eds.): MobiCASE 2009, LNICST 35, pp. 26–41, 2010.

paper aims at circumventing the tedious task of remembering complicated passwords or carrying any additional tags at the user's end for frequent accesses to a resource; rather it enables the user to act naturally in an environment, which is unique to him, and have that unique, natural interaction authenticate him.

Through this work we aimed at striking a balance between relying completely on automated actions taken by the system on one hand, and those performed explicitly by the user for authentication on the other. We selected a lightweight mechanism for capturing user actions, QR codes (a two-dimensional barcode), owing to the several advantages in deployment they have over other access control systems. They are easier to generate and can be read by nearly all camera-equipped phones (which are reaching ubiquity). Moreover, this technique is robust against any kind of sniffing attacks which other radio based tags like RFID are susceptible to. We have further combined role- and location-based access control to leverage context, thus allowing the creation of rules that are based both on roles and location, *e.g.*, a *student* can access the environmental control system in the *room* where he is attending his class or attending a meeting in a conference room. The set of policies written for this system facilitates its decision-making capability, based on the user's role and the location she attempts to access.

The remainder of the paper proceeds as follows: Section 2 acknowledges the related work performed in this field and elucidates how our proposed implementation extends this work. Section 3 introduces the basic context-based authentication scenario used to explain the working of the proposed framework. Sections 4 and 5 explain the implementation of the architecture and the design for our context-aware based authentication framework. Section 6 describes the various experimental scenarios which help us demonstrate the strength of our framework under different circumstances. It also introduces the policies used for defining the access control mechanism in our implementation. Section 7 discusses the attacks and threats against the system and how they can be handled. In the final Sections 8 and 9, we discuss the future work which can be performed to improve this framework and make it more scalable, and provide a conclusion.

## 2 Related Work

We leverage the concept of context-aware computing from the many papers related to Context Models and frameworks, Context and Devices, and Context and Mobile Professionals [Dey & Abowd, 2000]. In one such paper, 'Seeing is Believing' [McCune et al., 2009], the authors propose to use the camera on a mobile phone as a new visual channel to achieve security properties formerly attainable with techniques as passwords, but now in a more intuitive manner. This collection of approaches is termed: Seeing is Believing (SiB). The paper discusses the several possible configurations considering the presence of a camera, a display or both. We have used this as a basis for our approach here. SiB can be used to establish a mutual security context between the devices, without a trusted authority. The protocol discussed in the paper involves a pre-authentication phase where the user captures the digest of the other user's public key (with whom they want to authenticate). The paper also

describes Unidirectional Authentication using SiB in which one device is display-less and the other one does possess a display.

In a similar fashion, our project focuses on unidirectional authentication of the user with the help of a smart phone equipped with a camera. The uniqueness of our project stems from the fact that we aggregate contextual cues with the QR codes to authenticate the user. In our protocol we present three levels of authentication depending upon the risk associated with the location the user wants to access and the role of the user. In our scenario, access to the foyer of our facility requires the lowest level of authentication, while the conference room and server room have the intermediate and highest levels of authentication, respectively.

We propose to combine both passive and active authentication in order to achieve the aforementioned authentication levels. Cerberus [Al-Muhtadi et al., 2003] is a system which supports multilevel authentication, where principals are associated with confidence values. Its context infrastructure captures rapidly changing context information and incorporates it into the knowledge base. Rather than relying solely on expensive cryptographic measures like public key exchanges and certificates, we too use contextual information from the user's device such as calendar hints, GPS location, and Wi-Fi location to authenticate after the QR code has been scanned at the location where the user intends to gain access. In Cerberus, context-aware security policies are described in an expressive language that supports binary operators, quantification and complex inferring while our access policies are based on the XACML standard (eXtensible Access Control Markup Language). Unlike Cerberus, we use a combination of light weight tagging using QR codes and contextual information for authentication.

In the paper, Using camera-equipped Mobile phones for interacting with real world [Rohs & Gfeller, 2004], the authors demonstrate the feasibility of recognizing 2-dimensional visual codes with resource-constrained mobile phone devices. This has been achieved using low-quality images obtained from integrated CCD cameras and even QR codes. A light-weight recognition algorithm has been designed such that it minimizes the use of floating point operations in order to achieve reasonable recognition rates of the code. Since the codes can only encode a limited amount of information, they normally serve as a key that is resolved to the actual data of interest, which is the basic design concept of our framework.

Our solution uses both role- and location-based access control to leverage context, and thus allows creation of rules that are based both on roles and location. Covington et al. [2002] present a Context-Aware Security Architecture (CASA) for providing authorization services in context-aware environments and applications. This framework supports the collection of contextual information from resources, the environment and the users who interact in that environment. In addition, they have explored, through the context-aware security architecture, an implementation of the Generalized Role-Based Access Control mode. Like our work, XML is used to specify access policies, role definitions and relationships and is also used as a common representation to share data between the various services in the architecture. The architecture is based on the context toolkit and a complex algorithm is used to authenticate users on the basis of the context and environmental data. In our work, the architecture contains an access management module which derives data from the store/database and the user is authenticated on the basis of the context and roles they belong to. In future

work, we will propose an algorithm which identifies the user through his unique profile along with the environmental cues.

## 3 Understanding Context-Aware Authentication Scenario

To better explain the implementation of our context-aware authentication framework, we present a general scenario of a student attending school on a regular day.

The student's personal information is registered with the server managed by the school. His personal information includes name, email ID, username, academic track as well as context such as his default location (study room), meeting times and usual arrival time on campus. He is also assigned the role "student" and his access permissions are determined accordingly.

The QR codes are displayed at screens at various locations in the school. On scanning a QR code with his smart phone, the user receives a URL on his phone. When he accesses this URL using Wi-Fi or some other network, information such as the id of the application and history about his previous access (encrypted on the phone) and current location is posted in the response. This information is verified against his personal data and location coordinates on a relational database on the server. Upon success, her presence is registered and she is permitted access to the interior premises. These stages can be seen in Fig. 1.

**Fig. 1.a.** QR code displayed on a screen

**Fig. 1.b.** User receives URL on his phone after scanning the QR code


**Fig. 1.c.** User prompted to enter password for high level authentication


**Fig. 1.d.** User granted access after successful authentication

## 4 Architecture

Our context-aware authentication system has the following major components, as seen in Fig. 2.

### 1 Core Access Management Module(CAMM)

This is the access management module which decides what challenges are presented to the user at the authentication sites and maintains usage patterns for every user. This module has two main entities:

### a. QR Code Generation and Management

This system maintains specific information for each QR code being displayed across all authentication sites. The characteristics that are stored are:

- The unique code being displayed
- Expiry time
- Message (optional)
- Owner
- Display location

### b. Usage Patterns

The system keeps logs of all authentication attempts, their times, their results and other context information that was provided or sensed in relation to every attempt. This provides valuable data to interpret system usage, learning user behavior, checking malicious usage and providing feedback on efficiency of deployment sites by giving information on which sites are popularly used and if a better distribution of sites could be achieved.

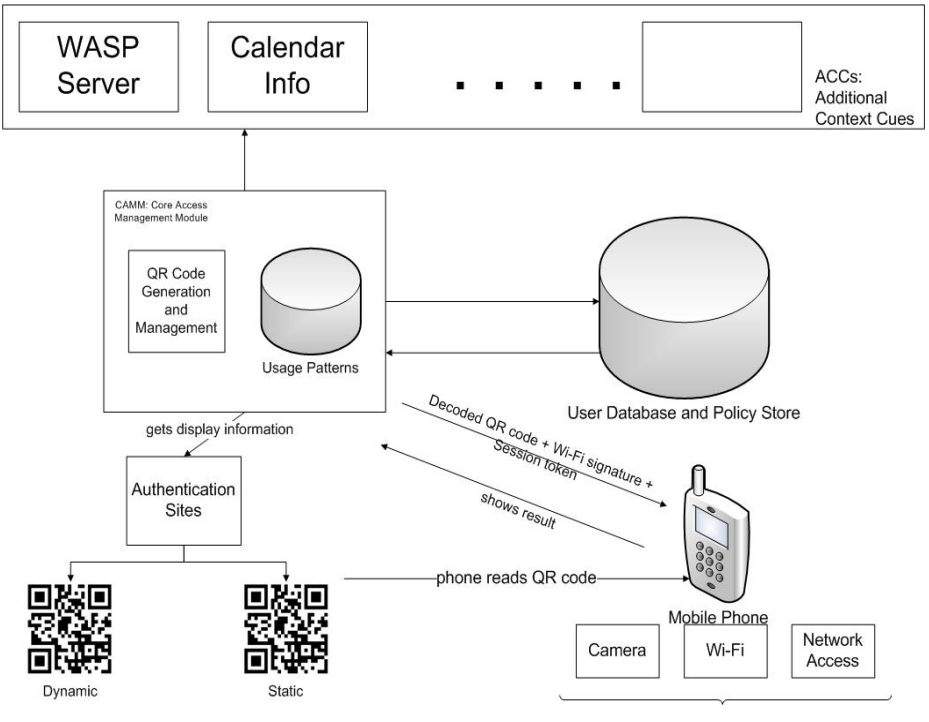


**Fig. 2.** Architecture of Context Aware Authentication Framework

## 2 User Database and Policy Store

This database stores user-specific information and policies to make decisions on whether to authenticate a request or not. Two essential pieces of information that are stored here are:

- Session token: The server maintains a list of active session tokens for each user, which is checked on each request. This token is freshly generated upon the completion of every successful authentication attempt and stored on the user's device to be used with the next attempt.
- Calendar id: To access the user's calendar information retrieving the relevant details to be used in access policy rules.

## 3 Client Mobile Device[1]

Our implementation requires the following capabilities on the mobile device:

a. Camera equipped: To be able to scan the QR code displayed.
b. Wi-Fi enabled: To utilize the WASP [Lin et al., 2009] interior positioning system's implementation on our campus.
c. Network Access: To make authentication requests to the server.

## 4 Authentication Site

The system can have different kinds of authentication sites based on security and cost requirements.

a. QR code dynamic display: These are linked to the server and display a dynamically generated code. These may be displayed on kiosks, tablet phones or computer screens.
b. bQR code static display: These are static codes which may be printed on paper or any other material and displayed at necessary locations which may require authentication.

## 5 Additional Context Cues (ACCs)

These are the other pieces of contextual information that may be available in a smart environment. Policy rules can be easily added to accommodate such ACCs. These are very important for learning a user's behavior and to eventually make our system more intuitive. Also, the availability of multiple ACCs provides a fallback mechanism in cases where some context information is not available. Some of the ACCs that were considered in our implementation were:

a. Weighted Access Point Similarity Positioning System. (WASP)[2] [Lin et al., 2009]: This system was implemented in our test environment; it provides a probabilistic location of a mobile handset in a building.

[1] The specific client application is not necessary and any bar code reader that can make a web request may be used. Although, using our client application enables a rich set of additional features which would be not available on other devices.

[2] The system can be used even in the absence of an implementation of the WASP system. Though, this would lead to the absence of the context cue available from indoor locationing.

b. Calendar information: User's time-schedule information from his online calendar. The attendees for an event are also considered to aid in the decision-making process.

These additional context cues can be used in situations where the system fails to authenticate a legitimate user in a location requiring a low or intermediate level of authentication. If, for example, a user is scheduled to attend an event to which he is unable to gain access, and the calendar information of a person authenticated for that event shows that the person stranded outside is also an attendee, the stranded person can be granted access.

The contextual cues used are Wi-Fi locationing, calendar, IMEI, user's role (such as student or faculty in a school environment). The cues can be extended to user alarm, to-do list, battery usage etc. and history can collectively be used to create a unique user profile. This unique profile can be used for authentication in absence of certain contextual cues.

We have defined a scalable system where the components CAMM, Client Mobile Device and ACCs can be easily expanded or replaced by other technologies. To illustrate this: the Access Management Techniques can be extended with NFC[3], Bluetooth security systems [4] or other biometric systems; the client mobile device can be extended with other capabilities through usage of NFC or Bluetooth; and the additional context cues can include the user's movement history or social context. We implemented this system using QR codes to first illustrate the basic concepts and secondly to create a system with low cost and certain definite advantages provided over other alternatives such as:

- QR codes support fast reading, using only the image recognition available across camera phones.
- Screens for displaying codes are everywhere.
- Codes can be printed and deployed anywhere.

## 5 Design

The user scans the QR code, **w**hich may be statically or dynamically generated depending on the security needs, at the reception area of the school building or any of the internal locations like the common study room, classrooms, library, and cafeteria or conference room. The QR code system provides a unique code to the user application, which along with other context from the phone, such as the session token and Wi-Fi signatures are sent back to the server as part of a request for further access. The relational database on the server maintains specific details about a user's roles, personal information, and alarms set in the user's calendar, and a pattern such as regular weekday routines, which may include the details of meetings, conferences or classes that the user is scheduled to attend. Information about the user's social context such as

[3] FeliCa-Contactless IC card technology by Sony, http://www.sony.net/Products/felica/abt/index.html

[4] BlueAccess BAL-100-BL, a wireless access control system. http://www.bluelon.com/index.php?id=248

whom else is in same location or neighborhood, gesture information determined from an accelerometer or from time history of location may also be recorded.

The access policies are based on XACML which is an OASIS standard that describes both a policy language and an access control decision request/response language. Using XACML simplifies the implementation and standardizes the policy set specification. Depending on the response collected from the user and the access policies defined, the user may be authenticated and granted access rights. The access rights can be role-based or location-based or a combination of both.

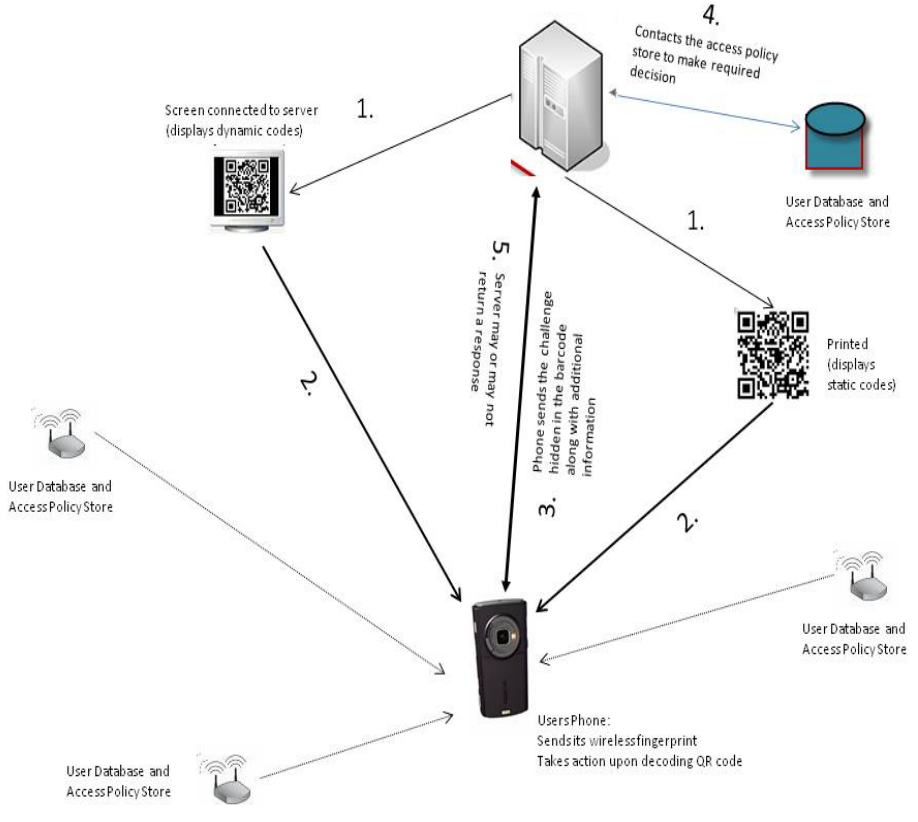


**Fig. 3.** Usage Scenario for the Framework

An example illustration of the working of our system is shown in Fig 3. It is comprised of the following steps

1. The user sees the deployed screen and touches it to see a QR code.
2. The user uses his phone to scan the code, and the phone automatically takes action sending the decoded code and its valid session token.
3. The server communicates with the user database and policy store to make a decision on whether to authenticate the user or not.

4. The server returns the response, which may be a success and a personalized message or a failure, or depending upon the access rules may prompt the user for further information.
5. The phone screen displays immediate feedback to the user about his action.

## 6 Experimental Scenarios

For the initial experimentation and testing purpose, we have developed several scenarios for location-based authentication on our campus. We establish role-based access control by using parameters such as Wi-Fi location and soft cues like meeting schedule from our campus' calendar and users' usual arrival time.

The users need not carry additional tags like RFID for authentication nor do they need to provide manual information about their schedule. The QR codes are displayed dynamically on screens as well as static printouts at any place for which access needs to be controlled. The QR codes represent the location as well as provide a link to access the server to request access. Also, by using dynamic screens we can generate new QR codes at different time intervals which are safer tokens owing to the randomness. Using static printouts, on the other hand brings predictibility and they again need to be changed manually and repeatedly to ensure that the same token is not used for a longer period of time. The interaction between the smart phone and the Barcode Management System is the crux of the application.

### 6.1 Scenario Based Roles

The users have been assigned different roles per their occupation. These roles determine what access permissions are to be given to the user for various locations. The four current roles include:

- Faculty
- Students
- Admin
- Visitor

Example: A user with role "Student" has access to the classroom but not to the computer server room. Only the user with role "Admin" has access to the server room. "Faculty" can always access their offices but "Student" needs prior permission to access the faculty offices, for example, for a scheduled faculty-student meeting.

The user ID consists of any of the parameters like phone IMEI number, user name and password. Wi-Fi locationing helps verify the presence of the user at a particular location using Wi-Fi co-ordinates. The calendar schedules are extracted from our campus' calendar.

### 6.2 Scenario 1

Goal: To recognize the presence of the user for registration/monitoring purpose and provide him with a session key

- User enters the school building
- User receives a QR code on a screen at the foyer (A new QR code is generated at regular intervals)
- User scans the QR code and gets a URL
- After confirming his User ID (with the personal information stored in the database), arrival time and Wi-Fi locationing, the data are appended to the URL
- The user now accesses that URL and thus sends context to the server using HTTPS
- The server checks the request parameters and ascertains authenticity of the user depending on the entries in the relational database and access polices. Depending on this, the server makes decisions on the access rights that should be made available to the user
- Moreover, further services can be made accessible via the person's phone.

### 6.3 Scenario 2

Goal: Access to specific locations like classroom, conference rooms, faculty offices per calendar hints

- There are QR code screens in front of classrooms, faculty offices, and conference rooms
- User receives a QR code on a screen at the entrance of the desired room
- User scans the QR code and gets a URL
- After confirming his IMEI number, calendar settings and Wi-Fi locationing, the data are appended to the URL
- The user now accesses that URL and thus sends context to the server
- Depending on the request parameters, the server ascertains authenticity of the user for access to specific locations by checking the meeting schedule (in the campus calendar) and permissions. The server makes decisions about the access permission to be given to the particular user. In case of access to the server room an additional authentication is required by the user by entering a password valid only for 'Admin'. On validating the password, the server permits access to the 'Admin'. On an unsuccessful attempt (invalid password), the permission is denied.

### 6.4 Access Rules and Policies

The access rules and policies defined for this project are role-based as well as location-based. These access policies work as the guidelines for the server to allow or deny access to a particular location such as classroom, conference room, faculty room or computer server room for the user. Again, per the role, the user is granted access to the specific location.

We have three levels of authentication associated with different locations depending upon the need for access control and associated risk at that location. The risk is minimal at places like classrooms because the access to location and data is not highly confidential, whereas server rooms are most critical in terms of security of data. The following table maps levels of authentication and risk levels with their associated locations in our scenario:

**Table 1.** Authentication vs. Location Map

| Level of Authentication | Location associated with it |
|---|---|
| Low | Classrooms |
| Intermediate | Conference , Faculty room |
| Highest | Server room |

These levels help in identifying the different levels of authentication required for various areas in the building. For example: A student gets access to his usual study area, but a classroom would be accessible only when the class on his schedule is conducted. A student is never allowed to get access to the server room which is associated with the highest level authentication in our scenario. For any other location access, the dynamically fetched values from the student's calendar are matched against the time at which the code is scanned and the other details such as whether there is a scheduled meeting at that time, and finally, according to the policies he/she is granted access for a certain period of time. In the future, we will explore the use of multiple levels of authentication for services within a particular location.

An example of the policy corresponding to the low level of authentication, where a user gets an access to the classroom is:

**Rule 1**

```
If ((Room type = Classroom) && (User ID belongs
to Role = Student)) – According to Table 1
{
        Permission = allowed
}
```

An example of the policy corresponding to the intermediate level of authentication, where a user gets an access to the conference room is:

**Rule 2**

```
If ((Room type = conference room && User ID = belongs
 to the list of User ID allowed))
   {
        if ( the start time of the event from the
        CMUwest_calendar > = [ within limit of +-
        15 mins] scan time of the code )
        {
            Permission = allowed
        }
    }
else
        Permission = Deny.
```

An example of the policy corresponding to the highest level of authentication, where a user is denied of the access to the server room is:

**Rule 3**

```
If ((Room type = Server Room)
&& (User ID belongs to Role = Student))
 {
       Permission = Deny
}
```

For visitors wanting to authenticate, a smart phone with the appropriate application must be given to them (likely in the foyer or reception area), if they do not already have one. Policies would have already been defined for the role Visitor, providing access to different locations in the building. In the future, to keep this project more scalable and the policies easy enough to be used in a distributed environment, we will design policies based on the concept of XACML.

## 7 Threats and Attacks

In this section we discuss the possible threats and attacks possible specific to our implementation considering the QR Code Generation and Management System to be the CAMM (Core Access Management Module). Here, we present several scenarios that could arise with such a system in place and evaluate its robustness in light of the same.

### 7.1 Replication of a Displayed Code for Reuse

Each QR code displayed at any authentication site has an expiry time and a binding with the location it has been deployed at. The validity periods can be calibrated to the security needs at the site, but are usually set to short intervals of a few minutes. This prevents a user from replicating a displayed QR code and using it later. Also, in the event a user tries to use a QR code from a location different from the authentication site, the request would be legitimate if made within the validity period because it would mean that the user was actually able to read the code from the correct location. In case of static codes, this is not considered a issue as they are aimed at a lower level of security and for services that do not need physical proximity for usage, for example: printers or Wi-Fi access.

### 7.2 Cloning or Theft of a User's Device

In the event that a user's phone is covertly cloned the framework is able to significantly limit the attack window. The attack window is limited to the time from which the phone was cloned to the next time the legitimate user makes an authentication attempt. If the legitimate user is able to successfully authenticate himself on his attempt after the cloning took place, it automatically makes the cloned copy defunct because now a new token sent by the server is stored on the phone which must be used for any subsequent attempts. While, in the other case where the legitimate user is not able to authenticate himself in his next attempt, he knows that there has been a breach and can take actions to deactivate his account.

On the other hand, the event of device theft is more difficult to contain but several properties of our implementation offer interesting solutions. Firstly, the system relies on physical presence of the device in front of the authentication site. It may not always be possible for the thief to appear publicly in front of a secured area risking suspicious activity and identification by others in the vicinity. Moreover the system can easily adapt to sense of insecurity or possible breach as discussed in 7.4.

### 7.3 Brute Force or Guessing Attacks

Each authentication attempt requires a user-requested short-lived code (through interaction with the touch-screen) and also a secure token received by the user from his past successful request. The generation of code on a site is highly random owing to the fact that they are generated when a user is at the site. This lends a unique randomness and limits the possibilities of finding a pattern in code generation. Moreover, codes expire in a couple of minutes which make it extremely difficult for an attacker to be able to brute force a 200 character value (as of the current implementation).

Moreover, to check brute force attempts, the CAMM can automatically disable a user account on receiving more than $n$ invalid requests.

### 7.4 Sense of Insecurity or of Possible Breach

If a sense of insecurity or of a possible breach is felt in the deployment space, the system can be easily adapted to guard against both. If a general sense of insecurity is felt at a site, the level of security can be elevated temporarily to add another level, which would request the user to enter his password upon a valid initial request.

Also, if specific devices or user accounts are suspected of being breached, those accounts can be de-activated or security levels could be elevated for just that user.

### 7.5 Attempts at Faking and Manipulation of Context Information

Attempts could be made by malicious users to try and manipulate context information to trick the system into granting extra privileges. For example, a malicious user may fake calendar entries in order to access a specific resource. To thwart such attacks, we plan to assign weights to various context cues considering their susceptibility to such attacks and also try and achieve peer verification where possible. For example, calendar entries would have a lower weight as compared to past history information or group membership. Also, an attempt would be made to see if there were other participants involved and if they have actually confirmed this calendar edit.

### 7.6 Other Attacks

The system relies on the strength of random number generation algorithms and security of over the air transmissions (on WLAN or available telecommunication networks) and weaknesses in these could directly affect this framework. But, securing over the air transmissions is outside the scope of this paper, and for all purposes it is assumed secure, though HTTPS is used for web requests.

Moreover, there arises an opportunity for an attacker to disrupt the system if he is able to gain access to any of the user's accounts like email or calendar due to the

user's carelessness or system vulnerabilities. Prevention against such attacks is outside the scope of this paper.

## 8 Challenges

Our suggested system is able to overcome critical shortcomings in currently used access control systems like physical keys or RFID tags like reducing the extent of breach on loss, doing away with the inconvenience of carrying an additional object just for access privileges alone and other attacks like passive sniffing in the case of RFID.

But, there are other challenges that crop up with our system like, support across the wide range of handsets, lighting conditions, ensuring network availability on the phone. The current prototype is built using the android platform. In case of poor lighting conditions or unavailability of network, the system will have to revert to password mechanism for authentication. Moreover, as with any new system, convincing people of the reliability and trustworthiness of this mechanism is a challenge with access control to physical resources.

## 9 Conclusion

In this paper we have described a new model for authentication in context-aware environments. We have used a combination of a user's context, authentication policies and light weight tagging to support role-based and location-based access control. This framework can be extended to support other contextual information from available resources, the environment and the users who interact with that environment.

QR codes have been presented as a means of authentication owing to their unique advantages like simple and inexpensive deployment, increased centralized control over authentication sites, rising presence of such capabilities in today's mobile phones and the novel 'line-of-sight' property they provide which is non-existent in other radio-based options.

## 10 Future Work

We will continue to iterate on our design and implementation of the Mobile Context Management System and demonstrate that we can apply it to assist in authenticating users and controlling access to data and services. We will prototype and evaluate several multi-user mobile applications incorporating this framework, such as a mobile transportation advisor and arranger, mobile eldercare concierge and monitor, mobile people finder, mobile environment controller, mobile meeting planner/scheduler and mobile advertising/shopping [Griss et al., 2002].

Furthermore, we will explore if this framework actually improves a user's experience while mobile, by providing appropriate services and information in appropriate situations, all in a secure manner. We would also like to extend our system to provide a robust platform for significant groups of users, adding multi-user context sharing,

additional context for critical levels of authentication and management to support mobile social applications.

Our future focus will also be on *fingerprinting* context attributes and using cues from user's history which would help the system to learn and define unique user profiles to authenticate users. The system will hence learn to give access to a user depending on the user profile rather than the role assigned to the user. Example: If a student has been attending meeting at a particular time every Monday for a month or more, he should be allowed access in the future depending on his previous registrations at the location.

## Acknowledgements

This research was supported by grants from Nokia Research Center and by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARO, CMU, or the U.S. Government or any of its agencies. We also acknowledge the efforts of Patricia Collins who reviewed multiple versions of the paper.

## References

1. Al-Muhtadi, J., Ranganathan, A., Campbell, R., Mickunas, M.D.: Cerberus: A Context-Aware Security Scheme for Smart Spaces. In: IEEE PerCom 2003 (2003)
2. Covington, M.J., Fogla, P., Zhan, Z., Ahamad, M.: A context-aware security architecture for emerging applications. In: Proceedings of 18th Annual Computer Security Applications Conference (2002)
3. Rohs, M., Gfeller, B.: Using Camera-Equipped Mobile phones for interacting with real World. In: Published in Advances in Pervasive Computing, Austrian Computing Soc (OCG), pp. 265–271 (2004)
4. Dey, A.K., Abowd, G.D.: The Context Toolkit: Aiding the Development of Context-Enabled Applications. Presented in the Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, June 6 (2000)
5. Dey, A.K., Abowd, G.D., Salber, D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Published in Human-Computer Interaction Journal 16(2-4), 97–166 (2001)
6. Griss, M., Letsinger, R., Cowan, D., Sayers, C., VanHilst, M., Kessle, R.: CoolAgent: Intelligent Digital Assistants for Mobile Professionals - Phase 1, Retrospective HP Laboratories report HPL-2002-55(R1) (July 2002)
7. Lin, T., Zhang, J., Griss, M.: Enhancement of Wi-Fi Indoor Locationing, Carnegie Mellon Silicon Valley, CyLab Mobility Research Center technical report MRC-TR-2009-04 (March 2009)
8. McCune, J.M., Perrig, A., Reiter, M.K.: Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentiction. International Journal of Security and Networks Special Issue on Secure Spontaneous Interaction 4(1-2), 43–56 (2009)
9. Meng, J., Yang, Y.: Application of Mobile 2D Barcode in China, Published in Wireless Communications, Networking and Mobile Computing. In: Presented in WiCOM 2008, 4th International Conference, October 12-14, pp. 1–4 (2008)