QUARTIRS

QR-Based User Authentication Replacement for Traditional Identification Reference System

Jenny Astrachan
jennya@mit.edu

Erica Du
ericadu@mit.edu

Marcel Polanco
mpolanco@mit.edu

May 13, 2015

Abstract

Although MIT ID numbers are currently used as a suitable means of identification, they are easily and often shared in public settings, suggesting ease of impersonation. This paper describes the design, implementation, and security analysis of QUARTIRS, a certificate based identification module created to replace the current ID number system used within the MIT community. QUARTIRS consists of both an Android mobile application and a web interface implemented in Django, and takes advantage of the existing trust and familiarity with MIT certificates for a simple, mobile, and extensible alternative to the current system.

1 Introduction

Within the MIT community, ID cards and numbers have a dual public and private nature. On one hand, MIT ID numbers were designed to be treated as public pieces of information, not to be used for authentication purposes [1]. Indeed, ID numbers within the MIT community are frequently passed between systems, written down, and shared. The public nature of ID numbers suggests that they can be easily obtained.

However, in practice, MIT students can use their ID numbers as the sole method of authentication in various scenarios on campus, such as paying student groups via TechCash, gaining access into MIT dorms, and using meal swipes at MIT dining halls [2]. These use cases coupled with the ease of impersonation in the current identification system suggest a need for a more secure alternative for student authentication in the physical world.

Before 1994, Social Security Numbers were used as institute wide identifiers, before issues with privacy were raised [3]. With the onset of new technologies such as smartphones and the widespread availability of the internet, the current system again needs to be adapted to meet new security demands and to stay relevant in the lives of MIT students.

2 Background

In the physical world, the current method of identification relies primarily on the use MIT ID cards. Oftentimes, students are able to substitute the use of the physical ID card with some combination of their name, ID number, and perhaps answering a question about themselves.

However, in the online world, authentication methods include the use of MIT certificates or a Kerberos username and password pair. MIT specific services such as WebSIS, Atlas, eCAT, and Stellar all rely on MIT certificates as a means of authentication. To obtain an MIT certificate, one must be in possession of three key components: a valid MIT number, a MIT Kerberos username and password, and a MIT-supported browser on a device [4]. The certificate is designed such that the MIT ID number cannot be obtained directly from the certificate [5]. Because of these attributes, MIT certificates are strictly more secure than MIT ID numbers.

3 Assumptions

QUARTIRS make the following assumptions about MIT students.

- MIT Students are willing to, or already have, MIT certificates, the MIT Certification Authority and an MIT personal certificate, installed on their devices. This assumption is made because of the already prevalent use and dependence of MIT certificates in MIT specific services listed above.
- MIT Students use certificates as a means of online identification.

QUARTIRS make the following assumptions about the nature of MIT Certificates.

- The MIT Certification Authority (CA) is trusted.
- Installation of an MIT certificate requires the existence of a passcode on a mobile device.

QUARTIRS make the following assumptions about devices and hardware.

- Devices with certificates belong to the owner of the certificate.
- Certificate access implies device ownership. If another person gained access to my mobile device, he or she would not be able to access my certificate because the mobile device is password or pin protected.
- Hardware and the physical world have physical and functional integrity. In other words, a smart phone can and will accurately scan a QR code when presented with one.

4 Threat Model

A variety of threats exist with the current system that QUARTIRS seeks to mitigate. Among these are:

Stealing/Spoofing Identification: In the current system, MIT ID numbers are shared as a means of identifying and authenticating a user. In the use cases we have identified, these numbers are often shared under a trust based system, often written down or stated explicitly, leaving these numbers susceptible to theft. Should one of these numbers be heard or memorized, an adversary may later replay the identification step using the ID number as a spoof.

False ID Generation: The current system does well to prevent the creation of false ID numbers. All IDs are generated securely through an initial process of randomization [6]. Our system must also provide such guarantees.

MITM-Based Attacks: In the current system, it is easy for an external third party to retrieve identity information through audible or visible channels. Even with ID cards themselves, attacks have been demonstrated in which one could obtain identification information for current physical access systems that use RFID technology [7].

Phishing: Another issue that remains a concern is knowing whom the party is seeking identification of a user. Should a user provide their information to a malicious party, their identification is left open for misuse.

5 Security Policy

5.1 Objective

The main security goals of QUARTIRS is to prevent the theft and misuse of identification information that could potentially lead to invalid or unauthorized transactions.

5.2 Definitions and Policy Based Terms

MIT certificates provide an alternate means of authenticating into MIT services than providing one's Kerberos username and password or MIT ID number. Certificates are distributed by the MIT Information Services & Technology department to appropriate parties affiliated with the MIT organization. This is strictly limited to those who have an associated Kerberos account, which may include students, internal organizations, faculty, and staff.

5.3 User Groups and Relevant Permissions

We denote the services of *creating, reading, updating, and deleting* material within the system by the letters C, R, U, D. The permissions assigned to each user group involve performing one or more of these CRUD operations to site data, or directly using a service provided by the site; we denote this under the label USE.

The following user groups are identified as having a role in the QUARTIRS protocol. Their appropriate permissions are identified as follows:

5.3.1 People with MIT Certificates

Those in possession of an MIT certificate stored on a computing device should have the following capabilities:

Service usage permissions

USE request to verify another MIT identity

USE provide identity information to another MIT user

Data Usage Permissions

RD access list of verified users that they requested

- Cannot access list of verified users requested by another user

5.3.2 People without MIT Certificates

Individuals lacking an MIT certificate should not have any access privileges that standard users have. This means:

Service usage permissions

- Cannot request to verify another MIT identity
- Cannot provide identity information to another MIT user

Data Usage Permissions

- Cannot access any lists of verified users

5.3.3 QUARTIRS Admin

Admins are those users who maintain the QUARTIRS service and contribute to its development. They are allotted the following access

Service usage permissions: [USE] all services available on the site.

Data modification permissions:

CRUD everything authorized to `People With Certificates`

CRUD internal site/application mechanisms – `code, pages, templates, server-info, client-info`

CRUD site API, routes

CRUD data usage policy

CRUD site maintenance mechanisms and variables – `request-throttling, DB-backup, DB-migration, site-uptime, blacklists`

CRUD the above list of authorizations

5.3.4 MIT

The role of MIT is to promote users in the category of a person in 5.3.2 to that of a person in 5.3.1 through the provision of an MIT certificate:

CRUD MIT certificates and how they are alloted

5.4 Privacy Policy

The QUARTIRS system may collect any data from users that use the services provided by the application. Administrators can not act as users given this statement, rather they can only modify the data created by or relevant to users.

The QUARTIRS system may disclose system information to third parties that includes user data, only under the agreement that that party will not further disclose the released information. Reasons for information disclosure include: scientific research, institutional research, system maintenance, transactions, and certificate releases

This policy is open to change at any time for any reason. Continued use of the QUARTIRS system constitutes consent to this privacy policy.

5.5 Terms of Service

Terms of Service for the QUARTIRS system extends those of MIT's general institute policies.

The Terms of Service is open to change at any time for any reason. Continued use of the QUARTIRS system constitutes consent to these Terms of Service.

6 Design

In the following section, we will discuss the design of QUARTIRS. Its design consists of the protocol, which determines how the clients interact; the trusted server, which serves as a trusted party to relay information between the parties; the client application, which is used by the party wanting to authenticate another party; and the mobile application, which is used by the party to authenticate themselves to another party.

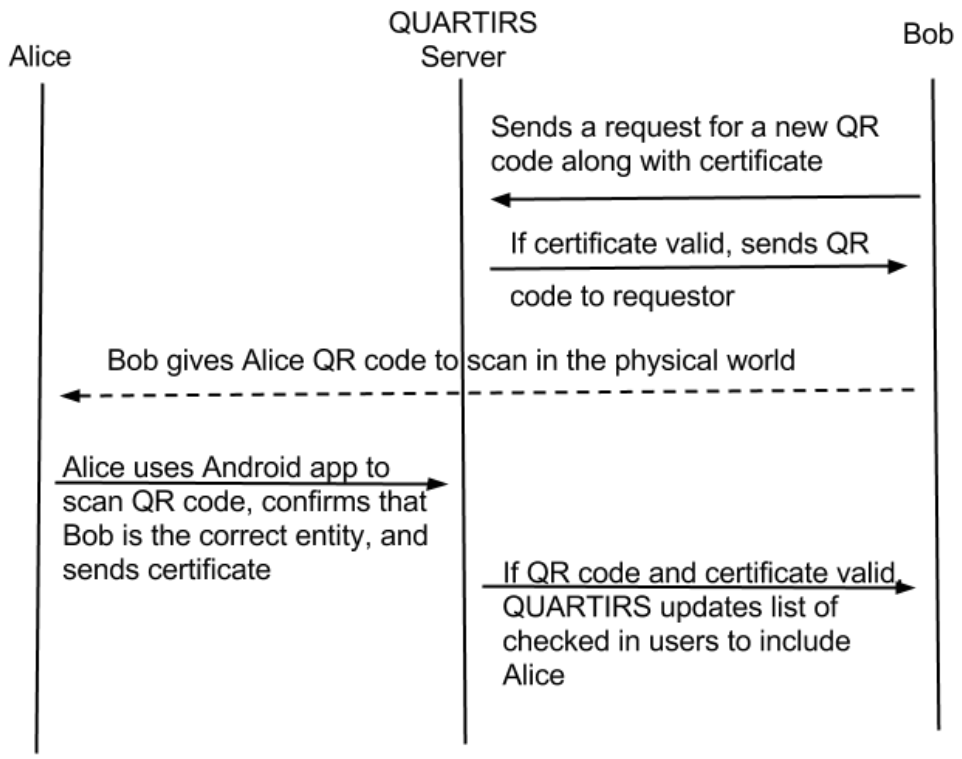


Figure 1: Protocol Diagram for QUARTIRS. Solid lines indicate interactions with the server, and dotted lines indicate interactions in the physical world.

6.1 Protocol

Before we could implement QUARTIRS, we needed to implement a secure protocol between the clients and the server. The protocol needs to protect against the types of attacks listed in our threat model (Section 4). The protocol, illustrated in Figure 1, has four phases: generating a QR code, scanning the QR code, checking in, and receiving the check in.

1. *Generating the QR code.* In this phase, Bob wants the server to send him a fresh QR code that can be used for someone to check in with him. He first sends a request to the server asking for a new QR code. In sending the request, he also sends his MIT certificate so that the server can verify that he is a trusted MIT entity. Once he is successfully authenticated, the server sends him a QR code which encodes a unique URL generated by the server to be used for a single check in.

2. *Scanning the QR code.* This phase of the protocol occurs in the physical world. Alice scans the code that Bob has just generated, which implies that she and Bob are interacting in person for this phase. Alice can therefore assess if she would like to check in with Bob depending on the situation.

3. *Checking in.* After Alice scans the QR code, she is taken to the unique URL that was generated by the server originally. The server asks her if she would like to send her identity information to Bob. If she does, she can approve and send her certificate to prove her identity.

4. *Receiving the check in.* Finally, the server then informs Bob that the code has been used and that Alice is the person who checked in with that code. At this point, Bob is free to generate a

new code and repeat the protocol.

These four phases of our protocol authenticate both Alice and Bob to the server and securely transfer identification information (in one direction, from Alice to Bob). Alice does not have to trust Bob and Bob does not have to trust Alice as long as both trust the server.

6.2 Trusted Server

The server is trusted by both the party authenticating and the party being authenticated. Because both parties trust the server, neither have to trust each other. The server relays communication between the two parties and coordinates the transfer of identity information by generating the QR code for the authenticating party and performing the check in for the party being authenticated. In order to perform any actions, the server must ensure that the party requesting the action is a trusted party. It trusts any party with an MIT certificate because MIT entrusted the party with a certificate, therefore making it a valid party. The server's API calls are designed so that information can only be retrieved by a party who is allowed to retrieve the information as defined by our security policy (Section 5). All network traffic with the server is encrypted through an SSL connection.

6.3 Web Client Application

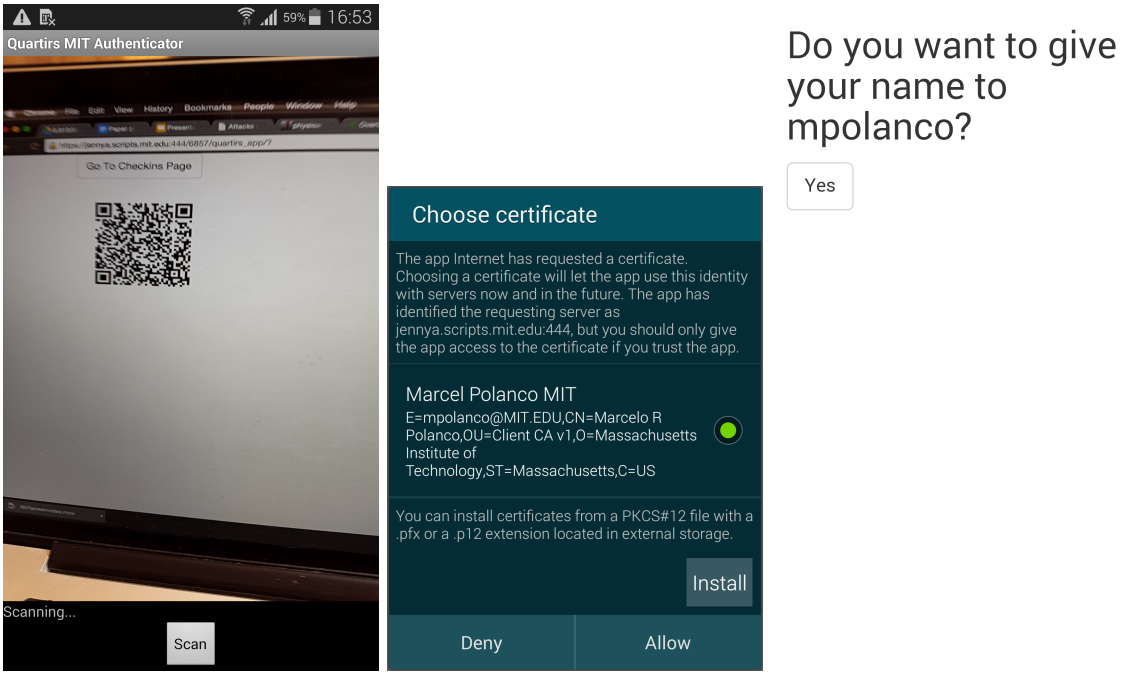
The web client is used by the party wishing to authenticate other parties. It is designed to have two different views, one for a QR code to be scanned and one to list the check ins that have occurred. With this design, the web client can be used on different devices allowing an authenticating party to present only the QR code to the client without also showing the parties who have been previously checked in. The web client must trust the server with it's MIT certificate, which is reasonable since any identifying information about a party cannot be retrieved from only a certificate [5].

6.4 Mobile Application

The mobile application is used by the party wishing to provide identification and authenticate to another party using the web client application. The application serves to scan the QR code presented through the web client application such that it can identify the server address to which it should present the client certificates. Once identity has been verified from the certificate, the application is used to verify the identity of the party requesting the user's identification. The application must also trust the central server with the user certificate.

7 Implementation

The following section describes our implementation of the QUARTIRS system, which includes an Android application and a web interface. In total we wrote on the order of 1000 lines of code, which can be found in our GitHub repository [8].



(a) Capture View (b) Certificate Selection (c) Authentication Verification

Figure 2: QUARTIRS mobile application interface as implemented on Android. (a) depicts to QR code capture view. (b) once scanned and directed to the central server, the user selects the certificate they wish to provide. In (c) the user verifies the user they are authenticating to.

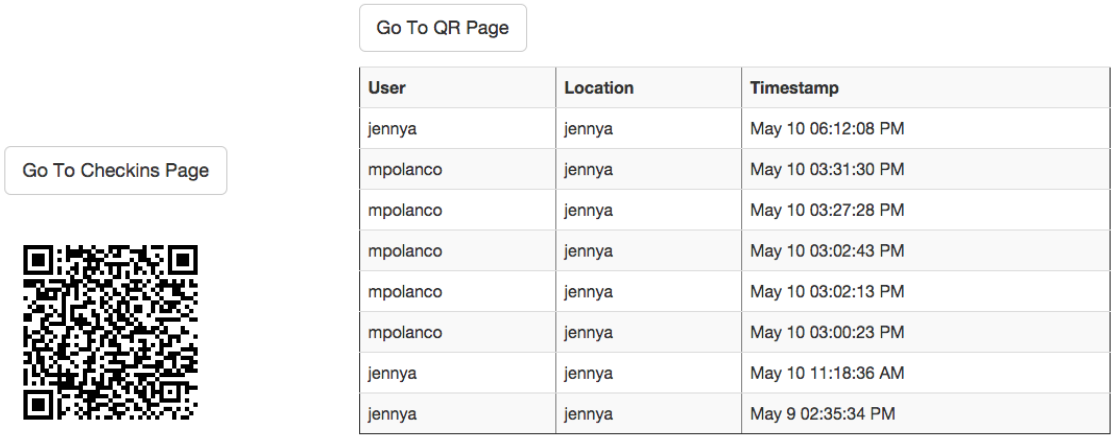
7.1 Android Application

For our mobile component, we contribute an implementation of the QUARTIRS mobile app on the Android platform. The application builds off of an extensible camera view that can be included in any application. With each incoming frame, the application analyzes the image for potential QR codes using publicly available QR code reading libraries [9]. Upon identifying a QR code, the code is scanned to identify the encoded message and, if the code directs to a URL prefixed with our trusted server's domain, the application attempts to establish secure communications with it over the internet using SSL. Should communications succeed, a web view is opened enabling the user to select their certificate as a means of authentication. If the server recognizes the provided certificate as a valid one provided by the MIT CA, the application displays a view detailing who is attempting to receive the user's identification information and asks that the user confirm that they still wish to authenticate with that service.

7.1.1 Application Permissions

The application uses the following permissions:

- **CAMERA** - Used to scan the QR code generated by the trusted server in our system
- **READ_EXTERNAL_STORAGE** - Necessary to access certificates stored on the device
- **INTERNET** - To establish communications with our trusted server



(a) QR Interface

(b) Check in List

Figure 3: In Figure 3a, the QR code interface is shown for the authenticator. After a valid certificate is presented to the server, the server sends back a QR code as shown, which encodes a unique URL generated by the server. Figure 3b shows the interface displaying the list of checked in users.

7.2 Web Application

We wrote the web application in Django. We used Django modules for generating QR codes [10] and requiring MIT certificates [11]. In the following subsections, we discuss in more detail how certificate authentication was achieved, the databases used, and the HTML pages displayed to the users.

7.2.1 MIT Certificate Authentication

In order to implement user authentication with MIT certificates, we needed to host our application on the MIT Scripts servers. By accessing the server over HTTPS and using port 444, Scripts requests the server from the user and parses information from it into environment variables to be used by our Django view code. We used Django's `login_required` method decorator in order to require certificate authentication for all functions. This decorator takes in the login URL as a parameter and only allows the function to be called if there is an authenticated user logged in.

7.2.2 Databases

In our implementation of QUARTIRS, we used two different databases. They are `QRTable` and `ValidatedUsers` and their fields are specified as follows:

1. `QRTable`

   - `entity_b`, a `CharField` storing the user who requested the QR code and will receive the identity of the authenticated user.
   - `entity_a`, a `CharField` storing the user who scanned the QR code and has chosen to be authenticated to the user who requested the QR code.

- `qr_hash`, a `CharField` storing the hash of the QR code requested by `entity_b`. This hash is part of the unique URL generated by the server and encoded in the QR code. The default value is '', which represents that the QR code hasn't been used yet.

2. `ValidatedUsers`

   - `entity_a`, a `CharField` storing the user whose identity was sent to the user requesting the QR code (the same as in `QRTable.entity_a`).
   - `entity_b`, a `CharField` storing the user who requested the QR code (the same as in `QRTable.entity_b`).
   - `check_in_time`, a `DateTimeField` storing the time at which the check in occurred.

When a user requests a QR code, the server creates a new `qr_hash` and stores it in the `QRTable` with the user's username as `entity_b`. `Entity_a` is left blank to represent that the QR code has not been used yet. When another user scans the code and is directed to the link encoded by it and presents their certificate, the server checks that no other user has scanned the code before by ensuring that the corresponding entry in `QRTable` has property `entity_a` equal to an empty string. If the code hasn't been used, the entry corresponding to that QR code's hash is updated, setting `entity_a` to be the user who scanned's username. An entry in `ValidateUsers` is also created with the same values for `entity_a` and `entity_b`. If the code has been used previously, the server informs the user that he should scan a fresh QR code.

7.2.3 HTML Pages Returned to the Users

The web application consists of three HTML pages seen by users, they are:

1. *QR Code page.* This page, shown in Figure 3a, displays an unused QR code to be scanned. It contains JavaScript that makes an AJAX call to the server to check if the code is still unused. This call is performed on a timer every second. If the server responds that the code has been used, the page refreshes and requests a new code.

2. *Checked in List page.* On this page, shown in Figure 3b, the user sees a list of users who have checked in with QR codes he has requested, or any entry in `ValidatedUsers` in which the user is `entity_b`. This page performs an AJAX call every second to refresh the table data. Therefore when a new check in is logged, the table will automatically refresh to see the change in data.

3. *Check in Confirmation page.* This page, shown in Figure 2c, is displayed after a user scans a QR code and is displayed on the Android device. It asks the user if he would like to check in with the username of the user who generated the QR code. When the user clicks 'Yes', the check in is handled by the server and the user is taken to a success page if the check in was successful and a failure page otherwise.

8 Discussion

We can assess the success of our implementation by revisiting the threats listed in our threat model (Section 4) and ensure that each threat is not possible with our implementation. The threats to address are stealing/spoofing identification, false ID generation, MITM-based attacks, and phishing.

Stealing/Spoofing Identification: This type of attack could occur in two ways: spoofing the authenticating party or spoofing the authenticated party. False identities of either party will not be accepted by the server because it only accepts valid MIT certificates as authentication and MIT only grants certificates to a party who can prove that they are in fact that party. Certificates cannot be spoofed or stolen, so the identity encoded by the certificate cannot be spoofed to the server. Additionally, since MIT requires that a party implements a passcode on their smartphone if they would like to download certificates on the device, our system inherently uses Two Factor Authentication without us having to implement it. This prevents a certificate from being used on a stolen device as long as the passcode is secure. Another way in which this attack could appear would be by spoofing the QR code to take a user who scans it to a malicious website. This is a form of spoofing the authenticating party because it presents a QR code as if it was presented by the authenticating party. We mitigate this threat with our custom Android application. The application only takes a user to a URL that is hosted by the QUARTIRS domain.

False ID Generation: This type of threat would occur if a party could make up a false identity and use it to check in with QUARTIRS. Since we use MIT certificates to authenticate a party and certificates cannot be gained for a false ID, QUARTIRS is safe from this threat.

MITM-Based Attacks: A MITM-based attack would occur if a third party could gain access to identification information without participating in the protocol we designed. By using only HTTPS connections, which ensures both encryption of data transmitted and authentication of the client and web server, to communicate with our server from both types of clients, a MITM-based attack is not possible.

Phishing: This attack could occur if the user who is scanning a QR code was tricked into thinking that their identity information would go to a party they trust rather than it actually going to a malicious party. We provide the user scanning the code with the username of the party who generated the QR code, who is the same party as will receive the user's identity information. The user's identity information is only given to the party who generated the QR code if the user scanning the QR code verifies that this is the correct party. Since the interaction is in person, the user scanning the QR code can ensure that the username which their identity information is being sent to is correct.

8.1 Future Work

In the future, QUARTIRS can be extended to add applications and make the system more useful for the MIT Community. The first step to extending QUARTIRS would be to add admin privileges and scale the system to be able to handle many users. After QUARTIRS is ready to handle more traffic, it could be extended to allow, for example, TechCash transactions or dorm access.

In order to prepare QUARTIRS to function with the number of users on scale with the size of the number of MIT certificates issued, we would need to implement admin privileges. This would include adding special pages for admin users and a way of distinguishing them from non-admin users, since the only form of authentication we currently use is certificate-based. We would need to store a table of usernames which have admin privileges, but this presents the vulnerability that someone could potentially add themselves to the list of admins. This would need to be implemented in such a way to mitigate this security risk. As stated in our security policy in Section 5, the admins should be able to modify any data in the system including the databases, the internal site, the policies related to the system, and perform site maintenance.

Once work is done to make QUARTIRS be able to handle the traffic it would need to at the scale of the entire MIT certificate-holding community, the system can be extended to support more applications. Two interesting applications in particular are handling TechCash transactions and integrating dorm access.

Two potential use cases for the implementation of TechCash transactions with QUARTIRS are handling vendor transactions as well as user-to-user transactions. For vendor transactions, one can imagine La Verde's or Stata Cafe accepting transactions with QUARTIRS. The vendor would type in the transaction amount and the user would approve the transaction and send their identity by pressing the 'Yes' button. In the second example, user-to-user transactions, this application could serve as a Venmo for TechCash. Users would be able to pay other users in TechCash for anything that is currently paid in cash, check, or with Venmo. Both of these applications require that MIT supports a TechCash API that the QUARTIRS server would talk to. TechCash currently has an API, but the amount of effort that would be required to allow QUARTIRS to communicate with it is unknown.

The second extension we will discuss is integration of dorm access. QUARTIRS could be connected to the current dorm access methods, where students tap their ID to a scanner and their picture appears on the screen and is checked by security guard. QUARTIRS could act in a very similar manner. A QR code would be scanned at the security guard's desk and the student's identity information would be given to the guard. A picture would not even be required since MIT certificates are much more difficult to spoof compared to an ID card. The username could be automatically checked against a list of usernames which have access to the dorm and access could be granted or denied. If MIT wanted additional security beyond that which certificates provide, when the student's identity is presented to the security guard, the student's ID picture could also be accessed to provide an additional method of authentication. This would be useful to students so that they would not have to get their ID out when they want to access a dorm. More likely than not, students have their smartphones in their hands and would not have to worry about carrying an additional ID card.

9 Contributions

This paper presented motivations for and the design, implementation, and security analysis of QUARTIRS, a proposed certificate based replacement for the traditional MIT ID number based identification system. We have described in detail the proposed Android application and web application, as well as the means in which users would interact with both interfaces. Furthermore, we have identified potential threats and addressed them within our design, implementation, and security policy. QUARTIRS will become a foundation on which future work around physical authentication can be built upon, and has the potential to play a key role in improving the security and usability of many existing services at MIT, like dorm access and TechCash transactions. The ideas on identification in the physical world described in this paper have the possibility of being further expanded on in settings beyond the MIT community, into any environment in need of a means of physical authentication that already relies upon certificates as a means of online authentication.

References

[1] Scott Thorne *People Related Projects* 1994: http://web.mit.edu/mitid/www/t_info.admin-arch.37707.TXT

[2] MIT Division of Student Life. "Student Groups - TechCASH Sales or Reader Rentals." *TechCash Services.* http://studentlife.mit.edu/techcash-sales-and-reader-rentals

[3] Jonathan A. Ives *The History of the MIT ID*, 1999: http://web.mit.edu/mitid/www/history.html

[4] MIT Certificates Guide. https://ist.mit.edu/certificates/guide

[5] MIT IS&T. "What is stored in an MIT Personal Certificate." *The Knowledge Base.* http://kb.mit.edu/confluence/pages/viewpage.action?pageId=3908944

[6] MIT Department of Facilities. *Security Policies.* http://web.mit.edu/semo/security/policies.html

[7] Josh Mandel, Austin Roach, Keith Winstein. *MIT Proximity Card Vulnerabilities.* http://web.mit.edu/keithw/Public/MIT-Card-Vulnerabilities-March31.pdf

[8] https://github.com/ericadu/quartirs

[9] https://github.com/dm77/barcodescanner

[10] https://github.com/pablorecio/django-qrcode

[11] http://web.mit.edu/snippets/django/mit/