Security Analysis of Mobile Applications: A Case Study of a Collaboration Tool in Healthcare

Julian Jang-Jaccard, Jane Li, Surya Nepal, Leila Alem
CSIRO Computational Informatics (CCI)
{firstname.lastname}@csiro.au

Abstract— **Mobile-based collaboration tools are increasingly used for communication and information sharing in delivering healthcare services that need collaboration across different geographical locations. Some of the typical features found in the collaboration tools include video conferencing facility, images/documents exchange in real-time, and annotations to point and draw on shared rich media content. Though the innovations and conveniences of such collaboration tools are well understood, security implications of such systems are often overlooked. As a result, necessary security mechanisms are not supported by them. This can lead to serious security threats and privacy violations. In this paper, we first present a collaboration tool which was developed to facilitate the collaborations among health care providers using pervasive mobile devices for delivering health services to remote and regional areas. We provide a comprehensive security analysis of the tool. The aim of the analysis is to understand a variety of end-to-end security mechanisms needed in different layers of the system. We also provide security recommendations which can improve the overall security of the system.**

Keywords- Mobile Devices, Security Analysis, Health Applications, Collaboration

I. INTRODUCTION

In many countries, community health workers play an important role in assisting patients living at their homes. For example, in Australian rural areas, health workers such as community nurses visit patients regularly at their home to carry out many clinical tasks (i.e., interviewing patients, assessing their health progress, collecting medical samples and images, administering medical procedures, etc.). Providing clinical assistance at home environment however faces a number of unique challenges that are different from hospital environment. One such challenges is the difficulty of getting supports from expert clinicians who reside in different geographically locations [10].

The use of collaboration tools in healthcare services, assisted by information and communication technology (i.e., also often called as *telehealth*), has been considered as an effective way to connect and share information among people who need collaboration from distance [11]. Typical features found in many collaboration tools include video conferencing facility, support for images/documents sharing in real-time, as well as some level of shared annotations where people at different locations can point and draw pictures simultaneously.

Mobile devices, such as smartphones and portable tablets (e.g., iPads), have increasingly become popular and are being used to support such collaboration. The drive for using mobile devices comes from health workers' mobility needs [10]. In addition, mobile devices offer portability. Health workers can readily carry them around for patient home visits and use many media rich features to communicate and share information with other healthcare professionals in remote locations. Though the advances in mobile collaboration tools have brought much innovations and conveniences for people working in health domain (and other areas as well), the security implications of them are often overlooked. Therefore, supports for appropriate security mechanisms are often absent in such tools. This can lead to serious security threats and privacy violations. Hence, there is a need to perform a thorough end-to-end security gap analysis of such tools and to provide innovative solutions to address the identified gaps.

In this article, we first present a mobile collaboration tool developed to support remote collaboration on health service delivery. We then provide a comprehensive security analysis on the tool. The application, named *ReColl* (Remote Collaboration Platform) was developed to meet the requirements suggested by health professionals to use in home care environment [10]. We start the article by giving an overview of *ReColl* and describing its underlying architecture and major features. Then we define end-to-end security requirements that need to be considered while developing a system similar in features to *ReColl*. This is followed by a comprehensive security analysis based on the defined security requirements. The result of the security analysis is to provide a set of guidelines for mobile application developers towards the understanding of where to provide a variety of security mechanisms in different layers of the system. We hope this helps to raise an awareness of security implications in the absence of appropriate security mechanisms. We also present a list of security recommendations that can assist in improving the overall security of *ReColl* (and other similar systems).

COLLABORATECOM 2013, October 20-23, Austin, United States

DOI 10.4108/icst.collaboratecom.2013.254212

The rest of the paper is structured as follows. In Section II, we describe the overview of *ReColl*. In Section III, we define a list of security requirements to be considered for *ReColl*. In Section IV, we describe the details of end-to-end security analysis on *ReColl*. In Section V, we provide a list of security recommendations. In Section VI, we present the related work. Section VII presents the concluding remarks and future work.

II. RECOLL PLATFORM

ReColl is a portable collaboration platform which was developed by researchers in CSIRO (Commonwealth Scientific and Industrial Research Organization) [7]. Although it was designed as a generic collaboration platform, its use as a teleheath application was extensively studied in terms of design and implementation [10]. *ReColl* is a result of extensive discussions with healthcare professionals and the design has incorporated their work practices and collaboration requirements.

One of the trends in healthcare services delivery has been moving care into patients' homes. To reflect the trend, *ReColl* contains features to use in collaborations involving home care. Beyond the context of home care, *ReColl* also supports other aspects of collaborations in different scenarios. For example, it can be used for patients living in rural and remote areas to connect to urban medical specialists to get advices. Nursing home residents with limited mobility can use it to access medical resources provided by their local health districts without having to visit them.

A. Architecture

ReColl is housed in a hand-held tablet device as a hardware interface since it is portable and small enough to give individuals (e.g., home visiting health workers) flexibility to carry it around.

The tablet device used for *ReColl* is iPad3 with 9.7 inch display. As for the development environment, the applications that run on *ReColl* were developed under iOS version 5.1 using Xcode as SDK and Objective-C as a coding language. *ReColl* allows exchange of rich media contents such as video/audio, picture/image, and text/document. In the communication layer, it supports wireless Wi-Fi and mobile 3G/4G network. The receiving side (e.g., expert clinicians in remote locations) can use any platforms (e.g., smartphones, tablets, desktops) to communicate and share the rich content with *ReColl*. The architectural overview of *ReColl* is described in Figure 1.

B. Major Features

Remote medical consultations using collaboration tools typically involve a specific set of interactions. Designers of these tools need to understand the way conversation is carried out, how information is shared, and how certain gestures are captured and analyzed [9]. In this paper, we particularly address the consultation that requires a care assistant onsite with a patient. There are three parties involved in this type of consultation, a clinician that oversees the consultation remotely, a patient, and a care assistant who co-locates with the patient to assist with any tasks that the patient might need. A typical medical consultation can involve the following activities. The remote expert clinician examines a patient record, discusses it with the care assistant as a part of sanity check, then the clinician directs the care assistant to examine the patient (e.g., asking to do some motions, checking skin color, testing blood pressure or temperature, etc.) while the remote clinician observing the examination from the distance.

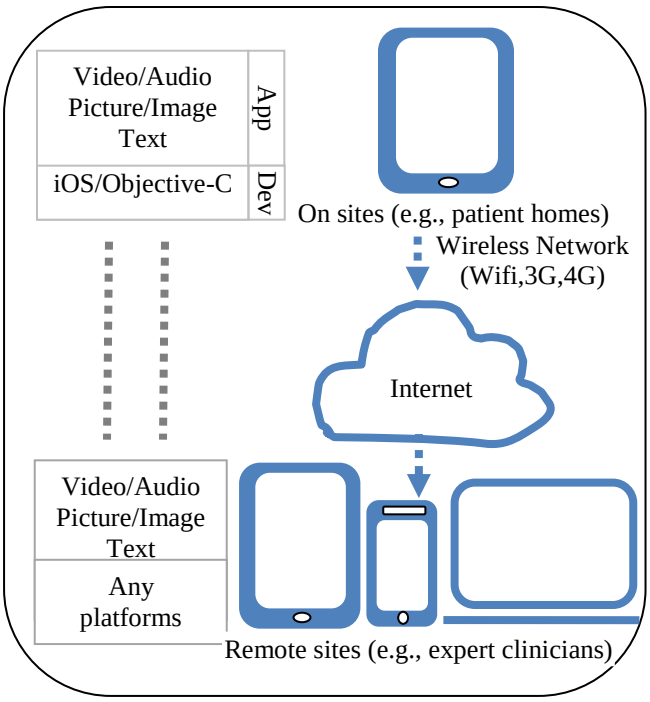

Figure 1. ReColl Overview

To support these interactions, a set of features is provided by ReColl, including:.

- *Video Conferencing*: This feature provides a facility for healthcare workers and the remote clinician to see each other and have clear verbal communication. *ReColl* also provides a multi-party video conferencing facility which allows more than one expert clinician to participate in a consultation. Outside of medical consultation scenario, video conferencing can also be used by patients to connect them to their family members, friends or peer groups to discuss their concerns and progress.
- *Real-time Information Sharing*: With the availability of high speed connectivity, many collaborating tools support capability for real-time interactions through the

exchange of HD images and rich text-based documents. With *ReColl*, parties involved in medical consultations can share images (e.g., X-ray, wound capture, etc.) and pdf documents (e.g., electronic patient records).

- *Asynchronous Data Exchange:* Traditional store-and-forward technique involves the exchange of pre-recorded data between clinicians offline. It is still considered as an efficient way of communication at certain situations where the availability of bandwidth becomes an issue in telehealth applications. For example, it is valuable in situations such as a real-time interaction is not practical due to limited bandwidth in certain areas (e.g., remote areas of Australia) or the clinician is not available at the time of request. *ReColl* provides asynchronous data exchange to cover this situation as well.
- *Shared Annotation:* Studies have shown that a shared view of a task space is essential to enable two collaborating parties to mutually reference to the physical environment [12]. The shared annotation assists real-time interaction by allowing the people in the medical consultation environment to be able to point and draw over shared documents and images.

III. SECURITY REQUIREMENTS

We first define a number of security requirements that are considered for the system developed as a collaboration tool, including *ReColl*. We outline general security issues that are involved with protecting the system (e.g., mobile device), the users using the system and the data stored in the system. We also examine security requirements of a number of special features often found in health services delivery support similar to that of *ReColl*.

- *System Protection*: This is to ensure that the system inside the mobile device (i.e., before running any applications) supports a set of appropriate security mechanisms to protect it from running malicious software.
- *Device Control*: This is to ensure that the physical mobile device is protected from any unauthorized access. For example, the iPad assigned to a certain patient is only accessed by that patient when he/she correctly enters a password. The device becomes unusable (e.g., data is safely deleted) if it's accessed by unauthorized personnel.
- *Application Protection*: This is to ensure that an application currently running is not influenced or modified by other applications. A security mechanism is required to ensure each application runs on its own isolated environment and "peeking out" of what other applications are doing is prohibited.
- *Identity Management*: This is to ensure that the identity (e.g., person/entity) accessing applications (and the data it manages) is correctly validated. For example, only authorized health professionals can log in to *ReColl*.
- *Access Control*: The system provides controlled access to data to ensure only authorized people access the data. For example, patient A's x-ray image and health records can only be accessed by his/her case assistants or doctors.
- *Data Integrity*: The system needs to incorporate mechanisms to both protect the stored data as well verify that the stored data has not been tampered with. Appropriate encryption mechanisms that work effectively on a mobile device are required. The keys that participate in encryption operations are safely kept in a secure place.
- *Transmission Security*: This requires the system to prevent unauthorized access of the data during transmission over the network and the data has not been modified in transit.
- *Video Conferencing Security*: All equipments used in a video conferencing (e.g., video screen, projector, audio equipments, etc.) are secure against containing malware. All software used in video/audio streaming supports appropriate security mechanisms (e.g., encryption) to protect the data while being streamed across the public network. In addition, there are supports for a proper identity management and an access control mechanism is in place to ensure that only authorized people participate in video conferencing session and share information.
- *Real-time Adaptive Security:* This requires that the system supports an adaptive security approach which can block real-time threats based on the analysis of real-time traffic (i.e., rather than traditional off-line static analysis based on block list)
- *Security in Shared Annotation:* This requires that the system provides a mechanism to restrict access to a particular group of trusted users to protect intellectual properties and personal privacy. In addition, the integrity of annotations is verified to ensure that the annotations have not been tampered while in transit.

IV. SECURITY ANALYSIS

We analyze the security features of *ReColl* in details according to the security requirements we defined earlier. Primarily the security analysis is done on iOS platform as *ReColl* was developed on iPad. However, we also provide corresponding Android security features wherever applicable.

A. System Protection

Security starts by having a secure system that provides a number of measurements to protect its properties (e.g., hardware, software, data, etc.). Before delving into security

in other layers, we examine a set of security features provided by a mobile device at a system level.

iOS provides a number of features to protect the system (e.g., especially any hardware and software components that run below iOS kernel level) from running any malicious code. This is one of the reasons why iPad was chosen as a platform to implement *ReColl*. At the lowest level, iOS supports a chain-of-trust operation, the concept that is similar to Trusted Platform Module (TPM) [8]. When an iOS device is turned on, its application processor immediately executes a code from read-only memory known as the Boot ROM. This immutable code is embedded during the chip fabrication, and is implicitly trusted. The Boot ROM code contains the Apple Root CA public key, which is used to verify that the Low-Level Bootloader (LLB) is signed by Apple before allowing it to load. This is the first step in the chain of trust where each step ensures that the next step is signed by Apple. When the LLB finishes its tasks, it verifies and runs the next-stage bootloader, iBoot, which in turn verifies and runs the iOS kernel. This secure boot chain ensures that the lowest levels of software are not tampered with, and allows iOS to run only on validated Apple devices [21].

The secure boot chain is automatically done by the device. After this step, it now depends on developers to ensure that any additional software running after boot-up is trustworthy as well. This can be done by code signing. To ensure that all apps come from a known and approved source and have not been tampered with, iOS requires that all executable code be signed using an Apple-issued certificate. To obtain an Apple certificate, developers who want to develop and install apps on iOS devices must register with Apple by joining the iOS developer program. The identity of each developer is verified by Apple before the certificate is issued. This certificate enables developers to sign apps and submit them to the App Store for distribution [21]. *ReColl* follows the code signing procedures by obtaining an Apple certificate, and signing the application with the certificate. The signed *ReColl* code is submitted to the Apple for distribution.

Unlike iOS which provides support from the system level, Android is more of an application execution platform comprised of an operating system, core libraries, development framework, and basic applications [13]. Android operating system is built on top of a Linux kernel though it has been diverged from the original desktop-based Linux kernel [14]. The divergence is there to adapt in the mobile environment by adding (or extensively modifying) vendor-specific drivers and modules. The Linux kernel is responsible for executing core system services (e.g., memory access, process management, access to physical devices through drivers) including security. Atop the Linux kernel is the Dalvik virtual machine along with basic system libraries. The Dalvik VM is a register based execution engine used to run Android applications. In order to access the lower level system services, Android provides an API through the system libraries [13].

Similar to iOS code signing, Android system requires that all installed applications must be digitally signed (code and non-code resources). This is done by each Android application as a package in an .apk archive. The .apk archive is similar to a Java standard jar file in that it holds all the code and all the application's non-code resources such as images. The developer signs the .apk with a certificate. The signed .apk is valid as long as its certificate is valid and the enclosed public key successfully verifies the signature.

B. Device Controls

Next security concern is at the device level to ensure that the mobile device is accessed only by authorized people. *ReColl* protects device theft by implementing iOS passcode to ensure iPad is only accessed by authorized personnel. Currently *ReColl* uses the default four digit PIN. However, this should be strengthened by recommending users to specify a longer and alphanumeric passcode. In addition, *ReColl* should also support a function with the device automatically wiped after pre-configured number of failed passcode attempts. This is to protect the device from brute-force attacks.

Android powered mobile devices also provide a device level protection through passwords using a feature called pattern lock. Users get a screen with 9 dots and the user draws a pattern of his/her choice using 4 of them. Similar to passcode, the device is automatically locked after failed pattern attempts. Remote wipe is not provided in Android.

C. Application Protection

Security vulnerability often happens through exploits enforced by different applications by influencing or modifying execution of any other applications [21]. The possibility for *ReColl* to be exploited by other applications is prevented by "sandboxing" mechanism supported by iOS. Application sandboxing controls that each application has its own environment where it keeps its own running processes and data that is isolated from other concurrently running applications. The sandboxing mechanism can reduce a potential attack surface where a malware (i.e., disguised as a legitimate Apple approved application) could intercept running processes owned by other applications to corrupt or access any sensitive information. With sandboxing, each app has a unique home directory for its files, which is randomly assigned when the app is installed. System files and resources used by the system are also shielded from the user's apps. The majority of iOS runs as the non-privileged user "mobile," as do all third-party apps. Native system APIs does not allow apps to escalate their own privileges to modify other apps or iOS itself [21].

However, Jailbreaking can break the protection provided

by application sandboxing. This is done by giving the user elevated privileges. Jailbreaking is achieved through exploiting bugs throughout iOS to give a user access to the kernel to which the user can gain root access. To jailbreak, it takes multiple bugs or flaws within iOS each of which gains access to a deeper level until kernel is reached. Once reached, iOS on the device may be modified to install non Apple authorized apps or to further jailbreak the device [15]. There are simple user friendly tools online that carries out jailbreaking process for a user even though he/she has no knowledge of how jailbreaking works. As of this writing, the latest available jailbreaking tool, known as 'evasi0n', can break the latest operating system iOS 6.1.2[16].

Application sandboxing is also provided in Android powered mobile devices. In Android environment, application sandboxing is performed at the Linux kernel level. In order to achieve the isolation, Android utilizes standard Linux access control mechanisms. Each Android application package (.apk) is assigned to a unique Linux user ID. This approach allows the Android to enforce standard Linux file access rights. Since each file is associated with its owner's user ID, applications cannot access files that belong to other applications without being granted appropriate permissions. Each file can be assigned read, write and execute access permission. Since the root user owns system files, applications are not able to act maliciously by accessing or modifying critical system components. Furthermore, each application is running on its own process to achieve memory isolation (i.e., each application has its own memory space assigned [13]).

D. Identity Management

Identity theft is a serious security implication that needs be addressed. A set of combined techniques in authentication/authorization is required to allow only authorized people access classified information (e.g., personal health records, patients' medical history, etc.). In addition, devices that exchange classified information need to be authenticated to ensure information is send/received by authorized devices.

In *ReColl*, patient authentication is implemented by a log-in page that verified username/password combination as seen in Figure 1.

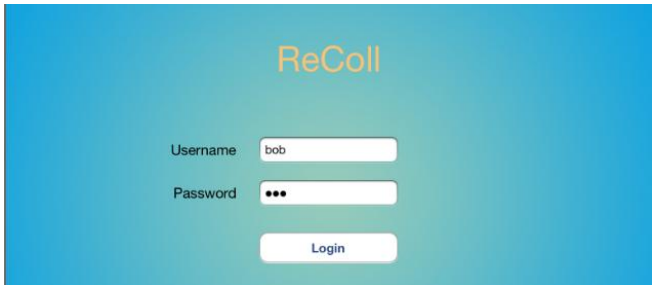

Figure 2. ReColl Log-in Screen

This authentication method is susceptible to a variety of well-known attacks and needs to be strengthened; for example, using two-factor authentication [17]. If *ReColl* is used for a real-time video conferencing, the identity theft can be prevented easily as people at other end can see the person interacting with the system. However, with asynchronous data sharing, if the combination of username/password was the only way to decide an identity of a person, the possibility of identity theft would increase.

In addition, a mechanism to link the patient with his/her data is required to prevent from data being interpreted incorrectly to avoid treatment errors. It may be possible to use biometric data for authentication, or more correctly, for identity verification [18]. Several studies propose methods based on features from electrocardiography (or similar biometrics) to verify patient identity [19].

Though not implemented in *ReColl*, device authentication is helpful in providing a better security mechanism. This is to ensure that medical information are exchanged between valid mobile devices (i.e., truly the devices they claim to be), untampered (i.e., not compromised by an adversary), and correct (i.e., they are the right devices assigned to the desired patient). Mutual authentication can assist the device authentication so that the requests came from the device owned by a patient and response is sent from the device owned by an expert [8].

Figure 3. ReColl Access Control Screen

E. Access Control

Access control provides a mechanism for controlled access to a patient's health record to ensure that only legitimate health professionals can access patient information so that the privacy of the patient is always maintained. With improper access control management, unauthorized access

to personal health information (or other sensitive medical records) could happen. For example, if an access policy is too permissive, patients may mistakenly modify their data. Insiders may modify medicinal records intentionally to use for a malicious purpose (e.g., to obtain reimbursement via insurance fraud [20]). Now we look into the access control mechanisms implemented in *ReColl*.

ReColl provides a simple mechanism where a patient (or a care assistant) can choose multiple doctors s/he wants to conduct medical consultation with (see Figure 3). This model only supports a very broad privacy by allowing any chosen doctors to see all medical information relate to a patient. A more fine grain approach is recommended such as adding extra functionality to capture patient's expressed consent as to who are allowed to read (or write or delete) of any parts of his/her records, and whether his/her medical records can be distributed to other providers (e.g., for medical researchers or public-health officials). Open interoperable consent standards such as consent directives [22] can be used to specify the management of machine-interpretable consents to ensure *ReColl* supports a standard way to access, collect, use and disclose of patient data.

The current access model of *ReColl* supports access rights by each individual. This is inflexible to incorporate cases such as a case assistant or a doctor needs to be subsituted by another. Role Based Access Control (RBAC) model has been used for years to enforce access control in traditional healthcare IT systems. Although RBAC is not "privacy-aware" (e.g., access is simply defined either by grant or deny), Ni et al. [23] discuss how to extend standard RBAC to make it "privacy-aware" and enforce authorizations at a finer level of granularity.

Another shortfall of the current model is that it does not contain any features to override access control rules in medical emergency cases.

F. Data Integrity

All aforementioned security measurements help to ensure the system runs only valid software and access is limited to authorized people. However, it needs additional security features to protect user data even in cases where other parts of the security infrastructure have been compromised.

The majority data protection is done by encrypting the data and hiding the keys in a secure place. In iOS-based devices, the keys are stored in a keychain which is a SQLite database in the file system. A separate key that encrypts the keychain itself is hardwired by a dedicated AES 256 cypto engine built between the flash storage and main system memory. The crypto engine contains a device's unique ID (UID) and a device group ID (GID), which are AES 256-bit keys fused into the application processor during the manufacturing process. No software or firmware can read them directly but can see only the results of encryption or decryption. The UID is unique to each device and is not recorded by Apple or any of its suppliers. Burning these keys into the silicon prevents them from being tampered with or bypassed, and guarantees that they can be accessed only by the cryptographic engine [21]. The UID allows data to be cryptographically tied to a particular device so if the memory chips are physically moved from one device to another (without knowing the passcode that unlocks the device), the files are inaccessible. The hardware-based crypto engine also helps in making file encryption more efficient which is critical as cryptographic operations can be complex and introduce performance degradation or battery life problems.

To guarantee data protection, it is critical that application developers understand security mechanisms that are offered by the underlying platform. To take advantage of what iOS offers, *ReColl* should support a mechanism to create a key to encrypt all user data as well as any data that is regarded as sensitive. The keys must be stored in a keychain with a regular backup.

G. Transmission Security

In addition to the measures to protect user data stored on a mobile device, there needs a secure communication to safeguard information from an adversary who wishes to obtain confidential medical information by observing the network transmission between mobile devices.

The adversary may inspect the wireless network packets and obtain sensitive medical data. *ReColl* could resolve this problem by encrypting all communications with an encryption key (i.e., encrypt all outgoing communication) and then store the encryption key in a secure place (i.e., keychain). However, even if the wireless network traffic is encrypted, a more sophisticated adversary can use traffic analysis tools to determine characteristics of the traffic (i.e., side-channel attack). *ReColl* could implement some of the countermeasures to trigger against such side channel attacks. Potential approaches include introducing noises so that the physical information cannot be directly displayed, filtering some parts of physical information, and making/blinding which seeks to remove any correlation between the input data and side channel emission [24]. In more sophisticated cases, an active adversary may inject frames or may selectively interfere with wireless frames to cause collisions. These methods enable the adversary to create a main-in-the-middle situation or to compromise the devices in a way that divulges their secrets. There have been increasing concerns regarding the wireless communications of implanted medical devices as reported in [25].

H. Video Conferencing Security

Video conferencing is becoming increasingly popular within hospital environment as the internet speed gets faster and emerging technologies enable medical consultation (or even some treatments) possible over distance. However,

over the course of the past few years, the questions about the security of video conferencing have sprung up as it involves systems (e.g., video/audio equipments), people (e.g., video conferencing participants), and (video/audio streaming) software.

For *ReColl* to ensure security on its video conferencing functionality, it first needs to ensure that all video conferencing equipments are robust (e.g., the equipments are all bought from authorized dealers) as many threats come from hardware that contains data stealing malware (e.g., Hardware Trojans). Illegal clones of hardware have been reported as source of hardware-based threats since the chances of illegally counterfeited hardware to contain hardware Trojans increase [24]. The chance to produce unauthentic hardware have increased with a new trends in IT company trying to reduce their IT expenses via outsourcing and buying off untrusted hardware from online sites.

Secondly, *ReColl* also needs to examine all video conferencing software to ensure there is no vulnerability as some streaming protocols used in video conferencing are reported to be vulnerable to a variety of attacks [26]. *ReColl* should also ensure video traffic is encrypted if sensitive information is being discussed.

Thirdly, *ReColl* needs to examine the underlying network and the policy that governs its use. Currently, *ReColl* does not support a firewall which can be added to control/monitor video conferencing traffic by specifying new ports for video conferencing traffic and monitor them, and control the time of day when video traffic is allowed.

Lastly, *ReColl* needs to educate its users in terms of video conferencing use. It is necessary to inform users to leave their video conferencing facility (e.g, in our case iPad) in a safe place to prevent the device theft. During video conferencing, the user needs to be in a position so he or she cannot be easily overheard or overlooked to prevent shoulder surfing.

I. Real-Time Adaptive Security

With the improved Internet speed and sophistication of information technology, many health applications are designed to share rich media content in real-time. For example, *ReColl* supports real-time video/audio sharing and document annotation between patients and clinical experts.

In this dynamic real-time environment, it is crucial to detect threats as simultaneously as possible as they occur before they cause damage and spread the threats to other parts of interconnected systems. *ReColl* at the moment does not support any mechanisms to support intrusion detection neither off-line nor real-time. Implementing adaptive security mechanism in *ReColl* could assist to watch a network for malicious traffic, search for any unusual end point (e.g., network ports) access attempts. In addition, real-time adaptive security can detect behavioral anomalies that are designed to target specific system components (e.g., kernel or Core API service) or people (e.g., certain doctors), as well as identifying real-time changes to systems. More advanced adaptive system can automatically remedy any damage done by threats or at least report the damage so that human operators can act quickly to reduce further damage [27].

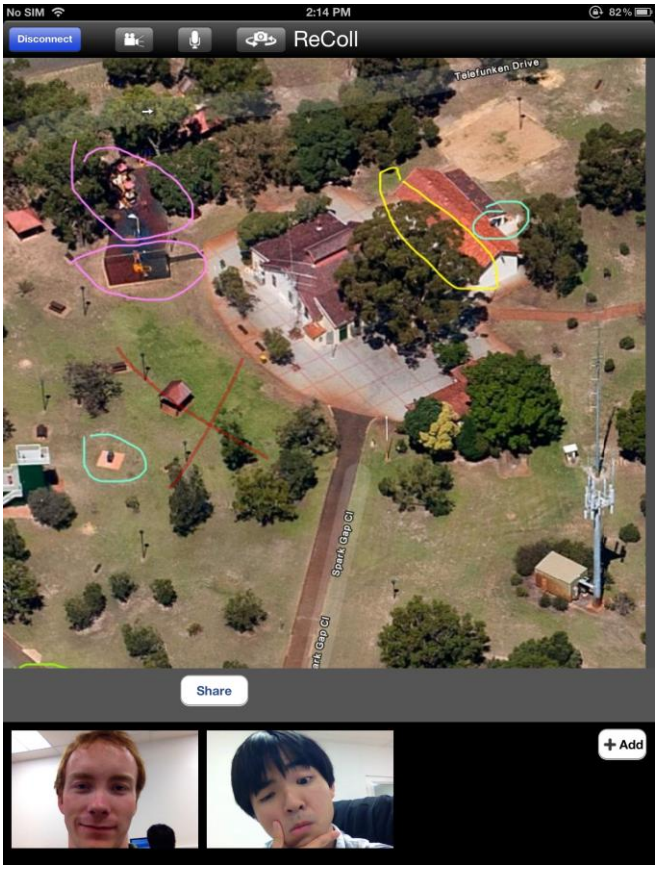

Figure 4. ReColl Shared Annotation

J. Security in Shared Annotation

Shared annotation is becoming increasingly popular to enable "value-adding" to digital resources. Through the mechanism provided by the shared annotation, collaborating entities (e.g., two expert clinicians) are enabled to attach additional data for comments, references, reviews, corrections as well as other types of external and subjective remarks [28]. The annotation facilitates group discussion and capture collective intelligence by enabling communities to attach and share their views on particular data and documents [28]. *ReColl* supports shared annotation mechanism where multiple entities can share and annotate on images and pdf documents in real-time. Figure 4 illustrates shared annotation capability of *ReColl*. The screen capture shows the way how two remotely located participants share a map then annotate on the map (e.g., pink, yellow, green circles and red cross) to find a specific place.

To reduce any possibility of unauthorized people to access the annotation, a security mechanism which can restrict access to the annotation to a particular group of trusted users is necessary. However, the access control to annotations provided by *ReColl* is rather too permissive and is not privacy aware. *ReColl* allows anyone with a correct log-in (and their contacts) to have access to any images and documents. A better approach would be done by adding an access policy. The access policy defines permissible types of access by individual users and resource types. For example, a specific image can be defined with access permission to create, read, edit, or delete. Then each person is assigned to that resource with the access permission.

In addition, the integrity of annotation needs verification to ensure that the original annotation has not been tampered while in transit. This mechanism is not supported by current *ReColl* but can be improved by signing the annotation with user's public key. A new technique called Provable Data Possession (PDP) generates a probabilistic proof for the data integrity based on only a small portion of the file [29].

V. SECURITY RECOMMENDATION

From the security analysis, we see *ReColl* has some level of security mechanisms in different parts of the system, but still remain vulnerable in many areas that open up to many types of threats. The following are some recommendations that can help *ReColl* (and other similar collaborating tools) to improve its security.

A. Privacy-Aware Identity Management

There are many device level identity management mechanisms *ReColl* can add to improve that only authorized patients and their case assistants have access to the device. *ReColl* administrators, who are responsible to distribute iPad to patients, need to educate the patients to use stronger passcode, other than default 4 digit PIN. *ReColl* administrators can enforce complex passcode requirements and other polices (e.g., passcode length and format, setting up auto wipe after predefined number of passcode attempts to prevent brute-force attack) in a centralized manner. Administrators also use a Configuration Profile that allows them to distribute and control configuration information on multiple iPads. Settings, which are defined by the Configuration Profile, must be configured in a way that the user cannot change them. If the user deletes a Configuration Profile from the iPad he/she owns, all settings derived from the Configuration Profile should also be removed. Configuration Profiles must be encrypted too. To completely prevent any potential removal, the settings can be configured to be locked to a device. In case of the report of the device theft, *ReColl* must be wiped out remotely. The wipe out is done by securely discarding the block storage encryption key when the wipe event is triggered rendering all data unreadable [21].

Currently *ReColl* only supports a rather simple identity management using log-in screen that allows which users are allowed to get into the system. Once log in is successful, the user virtually have no restriction to access any type of resources. This needs to be improved by implementing more privacy-aware mechanisms that provide a finer grain access policy.

Data anonymization is a technique to de-identify individual records to protect data from potential unauthorized access while still being used for analysis. This can be an interesting addition to *ReColl* to allow the use of the medical information by other vendors (e.g., research or public health domain). If provided, data anonymization technique must ensure that any potential data that could be used to identify the patient or the patient's relatives, employers, or household members are removed. The technique also requires providing a mechanism to have no actual knowledge that the remaining information could be used alone or in combination with other information to identity the patient.

B. Restricted Access Control

Like the identity management, the access control mechanism provided by *ReColl* is too permissible. Once a user is authorized to enter to the *ReColl* system, the user is allowed to access all resources without any restriction. This feature has a great potential to be a threat to privacy by allowing the user to see more than what is allowed. This can also contribute towards users modifying sensitive information mistakenly. *ReColl* can benefit by adding more sophisticated access control policy which allows each patient to express his/her consent in a fine granularity.

In addition, current *ReColl*'s access control mechanism is based on specific individual. This model is inflexible if the case assistants or assigned doctors are changed as the system needs to be updated. More natural approach would be Role Based Access Control (RBAC) model where the access is defined by roles. With RBAC, if assigned medical professionals change, the role of the replacing person would still be the same, and therefore does not need an update.

C. Data Encryption

For data protection in *ReColl*, we recommend to encrypt all sensitive data wherever possible. Utilizing advanced hardware-based cypto scheme provided by underlying iOS, *ReColl* needs to ensure that the key used for data encryption is stored safely in the keychain. If keychains are backed up over iCloud, *ReColl* needs to encrypt keychain itself. The key used for keychain encryption must be hardwired within the AES crypto and tied with user's passcode. This can effectively prevent adversaries somehow have access to keychain (e.g., either from the device or from iCloud) but still unable to decrypt the keychain to retrieve sensitive keys

without having the correct knowledge of passcode. The user must be educated to keep his/her passcode securely (e.g., without writing it down on a piece of paper where other members of family or visiting friends can easily see).

Encryption should also be done to all data that are transmitted outside of the device (e.g., to be transmitted to doctors for the further analysis, or for backup purpose). Again, the secret keys must be kept in a secure place so that they cannot be accessed by adversaries.

Often data that is used in video conferencing facilities is not encrypted. This is because developers often fall into misconception thinking that the video conferencing software provides the necessary protection. This can be far from true and *ReColl* must validate that the underlying video conferencing facility provides such protection. Otherwise, *ReColl* must provide an appropriate mechanism to encrypt all video/audio streaming data as well as any other sensitive data managed by the video conferencing facility.

D. Adding Logging and Auditing

No matter how secure a system is deemed to be, attackers always find ways to penetrate the system. It would be too naïve thinking that *ReColl*, with all the suggested recommendations implemented, would be 100% secure.

We recommend *ReColl* to add logging and audit trails which can identify any misbehavior when a misuse is detected. A number of automated log files can be created in *ReColl* to record any access to information; for example, log files to record user log-ins and log-offs, application started, or files accessed. *ReColl* must ensure that the history logs should be sufficient enough to provide evidence for any later disputes. Identification of critical information for logging, as what to be recorded, is an important issue to be considered before loggings are implemented. In addition, it also requires a mechanism to ensure the loggings cannot be modified by adversaries.

VI. RELATED WORK

Collaboration tools, such as *ReColl*, provide a facility for remotely located people to communicate and share information. These tools have been developed both by commercial vendors and academic institutes.

In the realm of commercial offerings, Skype [1], Apple Facetime [2], and Google Hangouts [3] are arguably the most well known services. They offer one to one video conferencing and multiparty conferences often for a fee with easy-to-use user interface. In the commercial telecommunication hardware devices, there are Polycom, Tandberg, and Avaya. Their focus is more on business communication solutions. These devices are equipped with multi-point control unit and PC display enabling seamless and real-time face-to-face collaboration at the desktop. In addition, these hardware devices have built in security functionality that is often supplanting or integrating with existing firewalls to provide a trusted route for remote users into room [32]. Though they offer a variety of features, they remain as more general telecommunication services without offering neither more sophisticated document sharing nor annotation features.

In more research oriented open source offerings, ooVoo [4] allows the user to have multi party video conferences with other people. Camfrog [5] allows a user to chat in public chat rooms with other users over text, voice and video. Video chat is only restricted to do by a single person at a time. The closest application to *ReColl* in terms of number of features it provide is Fuze Meeting [6]. The Fuze Meeting is an online meeting application that allows the user to have multiparty video conferences. It also allows document sharing and annotation as well as shared video watching. The user interface is not as simple as the other video conferencing applications with many features hidden in sub menus and the manual that details the insight of the platform is not available.

In terms of security analysis of health applications, Tan et al. [30] provides a security analysis on remote obstetrics care monitoring system. Their analysis provides security comparison between traditional monitoring system utilizing wired connection and a more advanced monitoring system that uses wireless connection. Though their security analysis on advanced Wi-Fi monitoring system has some level of similarity with ours, their analysis does not provide any detailed analysis on the security mechanisms implemented on both types of the monitoring system. Kotz [31] described a taxonomy of threats in mobile-based health applications (i.e., also referred as mHealth) with focus on privacy concerns. Some of the taxonomies he described in the paper were considered in defining our security requirements.

VII. CONCLUSION

We provided an end-to-end security analysis derived from a case study of *ReColl*. *ReColl* was developed as a general mobile-based collaboration tool to support communication and information sharing between distributed healthcare professionals in care deliveries.. The result of our comprehensive security analysis shows that *ReColl* only supports security mechanisms in limited places (e.g., device unlocking, log-in page). Even if some degree of security mechanisms were implemented, we evaluated that they were often too permissive and were not privacy aware. This leaves much room for privacy violation as it increases more chances for users to modify resources mistakenly or for adversaries to get access to the system and user data. Loggings and auditing features are not implemented in *ReColl* making it harder to identify when a misuse is detected.

We believe that the shortcomings identified in various

layers of the *ReColl* system existin other similar telehealth applications. The paper unlocks security issues at different layers of the system. Therefore, it provides a generic guide to the developers developing mobile application for sharing sensitive documents. We recommend that a security expert to be involved in a design and implementation phase of such systems.

ACKNOWLEDGMENT

We would like to thank Bo Yan for implementing the initial *ReColl* system and our summer students Richard Pilgrim and Jack Wong for the contribution towards the further development of *ReColl* and initial iOS security analysis.

REFERENCES

[1] Skype. www.skype.com/
[2] Facetime. www.apple.com/au/ios/facetime/
[3] Hangouts. www.google.com/+/learnmore/hangouts/
[4] ooVoo. www.oovoo.com/
[5] Camfrog. www.camfrog.com/
[6] Fuze Meeting. https://www.fuzebox.com/products/fuzemeeting
[7] Commonwealth Scientific and Industrial Research Organization (CSIRO). www.csiro.au/
[8] Trusted Computing Group. http://www.trustedcomputinggroup.org/
[9] D. R. Stevenson, "Tertiary-level telehealth: A media space application." *Computer Supported Cooperative Work (CSCW)* 20, no. 1-2 (2011): 61-92.
[10] J. Li, and L. Alem, "Supporting distributed collaborations between mobile health workers and expert clinicians in home care." *In CHI'13*, pp. 493-498. ACM, 2013.
[11] K. S. Rheuban, "The role of telemedicine in fostering health-care innovations to address problems of access, specialty shortages and changing patient care needs." *Journal of telemedicine and telecare* 12, no. suppl 2 (2006): 45-50.
[12] A. Leila, and W. Huang, "Developing mobile remote collaboration systems for industrial use: some design challenges." *In Human-Computer Interaction–INTERACT 2011*, pp. 442-445. Springer Berlin Heidelberg, 2011.
[13] D. Goran, M. Silic, and J. Krolo. "Emerging security threats for mobile platforms." *In MIPRO, 2011 Proceedings of the 34th International Convention*, pp. 1468-1473. IEEE, 2011.
[14] A Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, and S. Dolev. "Google Android: A state-of-the-art review of security mechanisms." *arXiv preprint arXiv:0912.5101 (2009).*
[15] M. H. Wolk, "iPhone Jailbreaking Exemption and the Issue of Openness, *The." Cornell JL & Pub.* Pol'y 19 (2009): 795.
[16] D. Gall, IPad 3 Secrets: How to Get the Most from Your IPad: IPad Mastery Made Easy Guide to Mastering Your IPad. CreateSpace, 2013.
[17] J. Moore, "The feds and PHR privacy," *Government Health IT*, Jan. 2009. Available at http://www.govhealthit.com/Articles/2009/01/26/The-feds-and-PHR-privacy.aspx
[18] S. Cherukuri, K. K. Venkatasubramanian, and S. K.S. Gupta. "BioSec: A biometric based approach for securing communication in wireless networks of biosensors implanted in the human body." *In Parallel Processing Workshops, 2003. Proceedings. 2003 International Conference on*, pp. 432-439. IEEE, 2003.
[19] J. Sriram, M. Shin, T. Choudhury, and D. Kotz, "Activity-aware ECGbased patient authentication for remote health monitoring," *in Proc. Intl. Conf. on Multimodal Interfaces and Workshop on Machine Learning for Multi-modal Interaction (ICMI-MLMI)*, Nov. 2009.
[20] P. Dixon, "Medical identity theft: The information crime that can kill you," The World Privacy Forum, May 2006. Available at http://www.worldprivacyforum.org/pdf/wpf medicalidtheft2006.pdf
[21] iOS Security – Apple. October 2012. Available at images.apple.com/iphone/business/docs/iOS_Security_Oct12.pdf
[22] "TP-30: HITSP manage consent directives transaction package," Healthcare Information Technology Standards Panel, Aug. 2008. Available at http://www.hitsp.org/ConstructSet Details.aspx?&PrefixAlpha=2&PrefixNumeric=30
[23] Q. Ni, D. Lin, E. Bertino, and J. Lobo, "Conditional privacy-aware role based access control," *in Proc. European Symposium On Research In Computer Security (ESORICS)*, ser. Lecture Notes in Computer Science, vol. 4734. Springer-Verlag, Sep. 2007, pp. 72–89. DOI10.1007/978-3-540-74835-9 6
[24] J. Jang-Jaccard and S. Nepal. A survey of emerging threats in cybersecurity. *Journal of Computer and System Science*. (in press)
[25] D. Halperin, T. Kohno, T. S. Heydt-Benjamin, K. Fu, and W. H. Maisel. "Security and privacy for implantable medical devices." *Pervasive Computing, IEEE* 7, no. 1 (2008): 30-39.
[26] LifeSize Security, "Video Conferencing and Security", White Paper. 2009
[27] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo. "Practical real-time intrusion detection using machine learning approaches." *Computer Communications* 34, no. 18 (2011): 2227-2235.
[28] I. Khan, R. Schroeter, and J. Hunter. "Implementing a secure annotation service." *In Provenance and Annotation of Data, pp. 212-221*. Springer Berlin Heidelberg, 2006.
[29] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. "Provable data possession at untrusted stores." *In Proceedings of the 14th ACM conference on Computer and communications security*, pp. 598-609. ACM, 2007.
[30] A. C. Tan, L. Bai, D. S. Mastrogiannis, and J. Wu. "Security analysis of emerging remote obstetrics monitoring systems." *In e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, pp. 329-334. IEEE, 2012.
[31] D. Kotz. "Threat taxonomy for mHealth privacy." *In Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, pp. 1-6. IEEE, 2011.
[32] http://www.ivci.com/videoconferencing-polycom-video-border-proxy-vbp.html