

2014

# Security and privacy aspects of mobile applications for post-surgical care

---

<https://hdl.handle.net/2144/14279>

*"Downloaded from OpenBU. Boston University's institutional repository."*

BOSTON UNIVERSITY  
GRADUATE SCHOOL OF ARTS AND SCIENCES

Thesis

**SECURITY AND PRIVACY ASPECTS OF MOBILE APPLICATIONS  
FOR POST-SURGICAL CARE**

by

**XIANRUI MENG**

B.S., Bloomsburg University of Pennsylvania, 2010

Submitted in partial fulfillment of the  
requirements for the degree of  
Master of Science

2014

Approved by

First Reader

---

Steven Homer, PhD  
Professor of Computer Science

Second Reader

---

Tanya Zlateva, PhD  
Professor of Computer Science

# SECURITY AND PRIVACY ASPECTS OF MOBILE APPLICATIONS FOR POST-SURGICAL CARE

XIANRUI MENG

ABSTRACT

Mobile technologies have the potential to improve patient monitoring, medical decision making and in general the efficiency and quality of health delivery. They also pose new security and privacy challenges. The objectives of this work are to (i) Explore and define security and privacy requirements on the example of a post-surgical care application, and (ii) Develop and test a pilot implementation Post-Surgical Care Studies of surgical outcomes indicate that timely treatment of the most common complications in compliance with established post-surgical regiments greatly improve success rates. The goal of our pilot application is to enable physician to optimally synthesize and apply patient directed best medical practices to prevent post-operative complications in an individualized patient/procedure specific fashion. We propose a framework for a secure protocol to enable doctors to check most common complications for their patient during in-hospital post-surgical care. We also implemented our construction and cryptographic protocols as an iPhone application on the iOS using existing cryptographic services and libraries.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Overview of the secure mobile application . . . . .	3
<b>2</b>	<b>Preliminaries</b>	<b>6</b>
2.1	Cryptographic Primitives . . . . .	7
2.1.1	Message Authentication Codes . . . . .	8
2.1.2	Digital Signature . . . . .	9
<b>3</b>	<b>The Security of the Post-Surgical Care</b>	<b>11</b>
3.1	The environment . . . . .	11
3.2	The Threat Model and The Security of The Protocol . . . . .	12
3.3	The Post-Surgical secure protocol/application . . . . .	13
3.4	The Secure Communication on the Complication Calculation . . . . .	14
3.5	Security Analysis . . . . .	15
<b>4</b>	<b>Deployment on a iOS device</b>	<b>18</b>
4.1	Performance . . . . .	22
<b>5</b>	<b>Open problems and Future work</b>	<b>24</b>
	<b>Bibliography</b>	<b>26</b>

## List of Figures

3.1	The communication protocol b/w doctor and the database . . . . .	17
4.1	Prompt Patient ID View . . . . .	19
4.2	List of Patient View . . . . .	19
4.3	Patient's Complication View . . . . .	20
4.4	Doctor's decision to the patient's complication will added to the Database .	21
4.5	DataBase Tables . . . . .	22
4.6	Communication time on local machine between iPhone and Java Server . .	22

## List of Abbreviations

Dec	Key decryption algorithm, page 6
Enc	Key encryption algorithm, page 6
Gen	Key generation algorithm, page 6
MAC	Message authentication code, page 8
negl.	negligible function, page 6
p.id	Patient identifier, page 20
pake	Password-base-authenticated key exchange, page 17
CCA	chosen-ciphertext attack, page 7
IND-CPA	indistinguishable chosen-plaintext-attack, page 7
PRF	Pseudorandom Functions, page 14
SEM	Semantic Security, page 7
sid	Session identifier, page 16

## Chapter 1

### Introduction

A secure model for electronic health records is needed for current hospital health systems. As the mobile-phone becomes the main communication tool for doctors and patients, a secure and privacy-preserving model/framework is required for such a mobile system.

#### 1.1 Background

Significant attention is now being given to organized assessment and reporting of quality of medical care across the United States, and this focus on quality will continue to play an ever more important role in patient care delivery. Patient outcomes following surgery are receiving particular scrutiny because of the attendant risks and increasing volumes of surgical procedures. For instance, some payments to hospitals are linked to outcomes and quality measures and there are already 8-9 events for which insurance does not reimburse hospitals no matter what happens to the patient. The American College of Surgeons (ACS) has adopted the National Surgical Quality Improvement Program (NSQIP [http://www.acsnsqip.org/main/aboutacs/about\\_overview.jsp](http://www.acsnsqip.org/main/aboutacs/about_overview.jsp) ), that tracks observed versus expected ratios for participating hospitals on a semi-annual basis. Other benchmarking approaches are in active discussion and development (<http://www.acssurgery.com/acs/chapters/ch0003.htm>).

Although each surgical procedure has its unique set of possible complications, there are a number of common complications that occur in the post-operative period. These include wound infection, venous thromboembolism (VTE), urinary tract infection, pneumonia, and myocardial infarction. While the number of these common complications is relatively small (typically 6-7) their prevention will have a large effect on outcomes and the quality of health care. There is a body of scientific evidence that links certain medical practices with decreased likelihood of post-operative complications. For instance, the incidence of VTE can be diminished with appropriate chemoprophylaxis, pneumatic compression boots, and patient mobilization. Unfortunately, compliance with post-operative regimens is often suboptimal due to a number of hospital and patient factors. Increasing work and time pressures have made it difficult for physicians to optimally synthesize and apply patient directed best medical practices to prevent post-operative complications in an individualized patient/procedure specific fashion. Although pre-operative checklists have become accepted practice in the operating room, such checklists and their optimal application have not been successfully translated into the post-operative arena.

One proposed solution on calculating the maximum likelihood post-surgical complications is to create an individualized risk profile in patients undergoing surgery. This profile takes into account significant elements of a patient's past medical history, records of the present illness, and the problems associated surgical procedure performed. Also, it is typically linked to the Electronic Medical Records (EMR) system. We incorporate all relevant data into a patient's medical identifier chip, and all relevant data will then be updated to the patient's EMR. Next, a wireless connection will occur between the surgeon's handheld device/phone and the patient's identifier on daily rounds. At that time, a checklist will appear on the surgeon's wireless device/phone. This checklist will be patient/procedure specific and succinct. The surgeon will address specific issues that need to be taken into account to minimize the possibility of post-operative complications. The surgeon's responses will, in a wireless fashion, communicate with the EMR and lead to appropriate orders being

generated.

## 1.2 Overview of the secure mobile application

Recently, issues related to database security and privacy have attracted increased attention, and many database applications can preserve privacy of both server and client. The technique of Differential Privacy has already been applied to data publication [4, 20, 22], counting queries [15, 21], histogram queries [8], log queries [9], and spatial data queries [5]. Furthermore, there are some recent works on applying differential privacy to time series databases [19, 18]. The objective of both these papers focuses on answering aggregated information from time series databases. For an exact match, especially for queries on particular medical records, differential privacy is not easy to apply since it is defined for statistical queries on databases. More importantly, the practical implementation of differential privacy seems difficult for mobile phone software systems. On the other hand, for exact queries, recent work utilizes applied cryptographic techniques such as private information retrieval [16] and oblivious transfer [17]. On our side, we note that it will be easier if we can build a mobile application using some heavy cryptographic techniques. However, considering the practical issues, one needs to consider the mobile phone's computational capability, power consumptions and other resources.

As mobile devices are convenient and easy to use, mobile health services can be provided via a phone that is capable of capturing patient information, images of patient symptoms, as well as a patient's audio information. Patient information transmitted through the system is uploaded to a secure, web-based medical record system, where doctors can provide remote medical advice, and automated algorithms can help determine patient risk profiles for a host of ailments, schedule clinic appointments and improve preventative care.

While there are significant opportunities to leverage these devices to increase the effective-

ness of mobile workers, there are also significant concerns about the privacy of sensitive data stored on the devices that IT must handle. We propose a framework, more specifically, a protocol for doctors to check all the necessary complications for their patient during post-surgery care. Specifically, a wireless connection will occur between the surgeon's handheld device/phone and the patient's identifier on daily rounds. At that time, a checklist will appear on the surgeon's wireless device/phone. The surgeon will address specific issues that need to be taken into account to minimize the possibility of a post-operative complication, and the surgeon's responses will, in a wireless fashion, communicate with the electronic medical record (EMR) and lead to appropriate orders being generated and treatment carried out.

Consider the following scenario: Suppose doctor Alice wants to check all types of complications for her patient Bob from a hospital's database. She must acquire all necessary fields and information and pass them to an existing "expert system", which will compute all related complications. Doctor Alice, on the other hand, will make a judgment of the complications, and she will also be able to update relevant decisions to Bob's medical records in the hospital's database. Due to the sensitivity of the health records, data sharing and third party computation require confidentiality and privacy. Although the potential benefits of the 'eHealth' mobile phone include better post-surgery service, time saving, efficient communication, and reduced cost, many privacy and security challenges arise in the system. Before we address the security model and privacy issues, we first posit some assumptions about the environment: 1). The central database system must be maintained by the hospital, namely, we may not use some other service from a third party since these may cause problems with trustworthiness and privacy-preservation. 2). The wireless communication between mobile devices will be totally unprotected. Thus, we must consider attacks such as 'man-in-the-middle' attack for our system.

The framework we proposed will be a secure and efficient protocol that enables the system to provide the confidential and integrated data to the doctor. Furthermore, our system

must avoid the exposure of the data to unauthorized parties while protecting the data from tampering. On the other hand, we also need to minimize communication overhead and computation by both server and client.

## Chapter 2

### Preliminaries

Throughout the presentation, we will freely use some cryptographic notations [10] for our security analysis.

**Definition 2.0.1.** *A function  $f$  is negligible if for every polynomial  $p(\cdot)$  there exists an  $N$  such that for all integers  $n > N$  it holds that  $f(n) < 1/p(n)$ .*

An equivalent formulation of the above is to require that for all constants  $c$  there exists an  $N$  such that for all  $n > N$  it holds that  $f(n) < n^{-c}$ . We typically denote an arbitrary negligible function by  $\text{negl}$ .

**Definition 2.0.2.** *A private-key encryption scheme is a tuple of probabilistic polynomial-time algorithms  $(\text{Gen}, \text{Enc}, \text{Dec})$  such that:*

1. *The key-generation algorithm  $\text{Gen}$  takes as input the security parameter  $1^n$  and outputs a key  $k$ ; we write this as  $k \leftarrow \text{Gen}(1^n)$ . We will assume without loss of generality that any key  $k$  output by  $\text{Gen}(1^n)$  satisfies  $|k| \geq n$ .*
2. *The encryption algorithm  $\text{Enc}$  takes as input a key  $k$  and a plaintext message  $m \in \{0, 1\}^*$ , and outputs a ciphertext  $c$ . We write this as  $c \leftarrow \text{Enc}_k(m)$ , since  $\text{Enc}$  may be randomized.*
3. *The decryption algorithm  $\text{Dec}$  takes as input a key  $k$  and a ciphertext  $c$ , and outputs a message  $m$ . We assume without loss of generality that  $\text{Dec}$  is deterministic, and so*

*write this as  $\text{Dec}_k(c) = m$ .*

It is required that for every  $n$ , every key  $k$  output by  $\text{Gen}(1^n)$ , and every  $m \in \{0,1\}^*$ , it holds that  $\text{Dec}_k(\text{Enc}_k(m)) = m$ . We will use the IND-CPA (indistinguishable chosen-plaintext-attack) to denote the security of a encryption scheme, which is defined as follows: for a probabilistic polynomial time (ppt) adversary, given the oracle access of the encryption oracle, the adversary can query the encryption function algorithm  $\text{Enc}_k(\cdot)$  polynomial many times. The adversary will then choose two messages  $m_0$  and  $m_1$  and send them to the challenger, and the challenger will flip a bit  $b \leftarrow \{0,1\}$  and send the adversary  $c^* \leftarrow \text{Enc}_k(m_b)$ . By seeing the challenge ciphertext  $c^*$ , the adversary will output a bit  $b'$  and wins if  $b = b'$ . We say the scheme is IND-CPA-secure for all ppt adversary, if probability that the adversary can win the CPA game is no greater than half plus a negligible function in the security parameter  $n$ . For CCA-secure (chosen-ciphertext attack), in addition to the oracle access of  $\text{Enc}$ , the adversary also gets the oracle access to  $\text{Dec}$ . Semantic Security (SEM) was first proposed by Goldwasser and Micali [6]. It captures the idea that a secure encryption scheme should hide all information about an unknown plaintext. This definition may match our intuition about what secure encryption ought to achieve better than does IND-CPA. It has been shown [1] that the IND-CPA implies SEM-CPA. We note that in our practical implementation of the protocol, it suffices to use the CCA-secure encryption scheme to provide the confidentiality of our data, while the encryption scheme also prevents the data being tampered in the mean time.

## 2.1 Cryptographic Primitives

This section reviews other cryptographic primitives used throughout this presentation.

### 2.1.1 Message Authentication Codes

A message authentication code, or MAC, is a short piece of information used to protect a messages integrity and authenticity. While anyone can generate a hash of a given value, a MAC assumes that the generator and the verifier share a common secret. The MAC algorithm accepts as input a secret key and an arbitrary-length message to be authenticated, and generates a MAC as output.

**Definition 2.1.1.** *A message authentication code (MAC) is a function  $h : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{R}$ . Here  $\mathcal{K}$  is called the key space  $\mathcal{K} = \{0, 1\}^k$ ,  $\mathcal{M} = \{0, 1\}^*$  is the message space, and  $\mathcal{R} = \{0, 1\}^n$  is the range for some  $n \geq 1$ .*

To authenticate a message  $m$ , an entity with a pre-shared key, say  $k'$ , computes  $\text{MAC}_{k'}(m) = (m, t)$ , where  $t$  is the tag (a checksum) on  $m$ . To verify  $(m, t)$ , a different entity owning the same pre-shared key  $k'$  checks that  $\text{MAC}_{k'}(m)$  does indeed equal  $(m, t)$ . The main idea of the MAC is that an adversary without the knowledge of the key should be unable to forge a valid tag for a given message that has not yet been authenticated. A MAC must therefore be able to resist adaptive chosen-plaintext attacks in order to be considered secure. This implies that no two messages  $m$  and  $m'$  should yield the same MAC under some unknown key. Message authentication codes share some similarities with conventional encryption, for instance in the way that communicating parties need a prior established shared key. However, the key is only used in a one-way function which build on the difficulty of computing certain mathematical primitives. This makes the MAC less vulnerable to attacks than regular encryption. For a more comprehensive review of cryptographic MAC algorithms and hash functions, see [10].

### 2.1.2 Digital Signature

Digital signature schemes allow a signer  $\mathcal{S}$  who has established a public key  $\mathbf{pk}$  to “sign” a message in such a way that any other party who knows  $\mathbf{pk}$  (and knows that this public key was established by  $\mathcal{S}$ ) can verify that the message originated from  $\mathcal{S}$  and has not been modified in any way. Signature schemes can be viewed as the public-key counterpart of message authentication codes, though there are some important differences as we will see below. The algorithm that the sender applies to a message is now denoted  $\text{Sign}$  (rather than  $\text{MAC}$ ), and the output of this algorithm is now called a signature (rather than a tag). The algorithm that the receiver applies to a message and a signature in order to verify legitimacy of the message is denoted  $\text{Vrfy}$ . We now formally define the syntax of a digital signature scheme.

**Definition 2.1.2.** *A signature scheme is a tuple of probabilistic polynomial-time algorithms  $(\text{Gen}, \text{Sign}, \text{Vrfy})$  satisfying the following:*

1. *The key-generation algorithm  $\text{Gen}$  takes as input a security parameter  $1^n$  and outputs a pair of keys  $(\mathbf{pk}, \mathbf{sk})$ . We assume for convenience that  $\mathbf{pk}$  and  $\mathbf{sk}$  each have length at least  $n$ , and that  $n$  can be determined from  $\mathbf{pk}, \mathbf{sk}$ .*
2. *The signing algorithm  $\text{Sign}$  takes as input a private key  $\mathbf{sk}$  and a message  $m$  from some underlying message space (that may depend on  $\mathbf{pk}$ ). It outputs a signature  $\sigma$ , and we write this as  $\sigma \leftarrow \text{Sign}_{\mathbf{sk}}(m)$ .*
3. *The deterministic verification algorithm  $\text{Vrfy}$  takes as input a public key  $\mathbf{pk}$ , a message  $m$ , and a signature  $\sigma$ . It outputs a bit  $b$ , with  $b = 1$  meaning valid and  $b = 0$  meaning invalid. We write this as  $b \leftarrow \text{Vrfy}_{\mathbf{pk}}(m, \sigma)$ .*

We require that for every  $n$ , every  $(\mathbf{pk}, \mathbf{sk})$  output by  $\text{Gen}(1^n)$ , and every message  $m$  in the appropriate underlying plaintext space, it holds that  $\text{Vrfy}_{\mathbf{pk}}(m, \text{Sign}_{\mathbf{sk}}(m)) = 1$ . The security of the digital signature is that, for a ppt adversary who trying to break the scheme,

given the  $\mathbf{pk}$  the adversary can ask polynomial many signatures of his choice give the oracle access of  $\text{Sign}_{sk}(\cdot)$ . The adversary will finally output a forged pair  $(m, \sigma)$ , where  $m$  has not been queried to the signing oracle. We say the signature scheme is unforgeable under an adaptive chosen-message attack if for all ppt adversary, the probability of the adversary forging a signature is negligible.

Together with the CPA-secure and CCA-secure encryption scheme, the MAC and digital signature are the main tools that we'll use for building the secure channel for the communication between the doctor and the hospital's database during the post-surgical phase.

## Chapter 3

# The Security of the Post-Surgical Care

Consider the following scenario: Suppose doctor Alice wants to check all possible types of complications for her patient Bob from a hospital's databases. She must query all necessary fields and information and pass them to an existing "expert system". The "expert system" will compute all related complications. The doctor Alice, on the other hand, will then make a judgment as to the complications and she will also send an update of relevant decision to Bob's medical record in the hospital's database. Due to the sensitivity of the health records, data sharing and third party computation require confidentiality and privacy.

### 3.1 The environment

The potential 'eHealth' softphone has many potential benefits, including better post-surgery service, time saving, efficient communication, and reduced cost. However, the system requires many privacy and security challenges. Before we address the security model and privacy issues, we first will state some assumption about the environment: 1). The central database system must be maintained by the hospital, namely, we may not use some other service from a third party since it may give rise to problems regarding the trustworthiness and privacy-preservations. 2). The wireless communication between the mobile device will be assumed totally unprotected. Thus, we must consider the possibility of 'man-in-the-middle' attacks on our system.

### 3.2 The Threat Model and The Security of The Protocol

We study the model where a malicious doctor/user want to learn some data entires from the database. In this case, authentication needs to be added to both our server and our client side. We note that in this threat model a malicious participant in the protocol will not gain any information about the EMR. We can prevent the malicious user from doing this by adding credential logins from the application itself. Health workers or doctors are authenticated via password and required to change their credentials on a monthly or weekly basis. Any read/write actions on a record are recorded with user ID and cannot be deleted. Reports are run on a regular basis to look for suspect access patterns. More generally, we study the man-in-the-middle threat model. The wireless connection between the doctor and the database can be exposed to other parties, who may be able to see all the transmitted message in the clear. Therefore, the man-in-the-middle threat model needs to be considered. A doctor who needs all necessary data from the electronic medical record will preform the computation on relevant complications to his/her patient. No other third party should be able to know the patient's record data. Our protocol should provide a secure communication between the doctor and the database. A malicious user can try to learn some portion of the data by seeing the messages in the clear. On the other hand, we also require authentication and identity-protection for the secure communication. Lastly, in the system security level, all plaintext will be stored in the softphone using a standard hash function, e.g. SHA-256. Namely, all the hash values will be stored in the softphone's memory addresses. We only consider the case where our mobile device is protected by the health worker, and it is secure in the system level. In practice, we note that our implementation on the iPhone uses the hash function to 'hash' the identifier of the patient, or the password if we will use the password-authenticated setting, which will be mentioned in the following sections.

Therefore, the general strategies for the attacker to acquire the sensitive information are the

following: When seeing the messages in the clear, the adversary tries to infer the medical records for some entries in the database; the adversary tries to ‘break into’ the hospital’s database; or the adversary tries to comprise possible users/doctors in the system. We’ll provide the formal security analysis in the section 3.5.

### 3.3 The Post-Surgical secure protocol/application

To enable the system to provide the confidential and integrated data to the doctor, our system must avoid the exposure of the data to other unauthorized parties while protecting the data from tampering. On the other hand, we also need to minimize communication overhead and the computation of both server and client. A secure channel is a communication that provides confidentiality and integrity to the communication between parties. Due to the sensitivity of the EMR, we certainly need to establish a secure channel for the communication between the doctor and the hospital’s database. We need the doctor to initiate a secure channel to the hospital’s DB. Canetti and Krawczyk [2] showed how to build a secure channel. Specifically,

**Theorem 3.3.1.** *[2] If Enc is a symmetric encryption scheme secure in the sense of IND-CPA and MAC is a secure then method Encrypt-then-Authenticate (Enc, MAC) implements secure channels.*

Theorem 3.3.1 provides a construction to building a secure channel. Numerous constructions of CPA-secure encryption have been built based on some well-known hardness results, e.g. the Discrete-Log, RSA. Our medical records will be sent through the secure channel that has been built between the a client/doctor and the server. As stated in the previous section, the secure channel will provide not only the confidentiality of the encrypted data, but also the integrity and authenticity of the data. One important aspect that we are using in this strong security tool is that we are trying to prevent the data from tampering while protecting its confidentiality.

### 3.4 The Secure Communication on the Complication Calculation

Doctor Alice will first initialize the protocol by getting the patient's ID from the sensor (e.g. scanning the a barcode using the iPhone or some other mobile device). Assume that the symmetric encryption scheme  $(\text{Enc}, \text{Dec})$  is IND-CPA secure, and that the asymmetric encryption scheme  $(\text{Enc}', \text{Dec}')$  is IND-CCA secure.

A patient typically has a unique Patient ID, denoted  $\text{p.id}$ , on his/her wristband. We may consider that this unique id number corresponds to the primary id in the hospital's database. Also, for security, we assume this phase is the face-to-face communication, thus no attacker can 'steal' the id at this time. The stored id is in Alice's mobile phone for future inquiry. We denote the patient's id to be  $C_{\text{p.id}}$ .

Alice will first initiate a 'handshake' with the DB, moreover, mutual authentication and key exchange will be done between Alice and the central DB. A 'Session' is a local procedure maintained by one party's activation of the protocol. A party, either of the client (doctor) or the server (DB), locally instantiates a run of the protocol and produces outgoing messages and processes incoming messages. A unique identifier can be assigned for each session maintained by both the doctor and the DB. This identifier can be derived from the system timestamp, patient's care time-clock, etc. We'll use the Sigma Key Exchange ( $\Sigma$ -KE) protocol proposed by [14] to derive some session keys, which implicitly achieving the mutual authentication. Let  $s_{\text{id}}$  denote the session identifier for a communication activated by doctor Alice. Alice will generate a random  $x$ , and send  $s, g^x$  to the database. Upon, receiving  $s, g^x$ , the DB will send  $s, g^y, id_{DB}, MAC_{K_1}(s, id_{DB}), SIG_{DB}(s, g^x, g^y)$  back to Alice. Alice will then send  $s, id_{Alice}, MAC_{K_1}(s, id_{Alice}, g^x), SIG_{Alice}(s, g^y, g^x)$  to the DB.

Both Alice and DB will generate some session keys using the shared seed  $(g^{xy})$ . This could be done using the Pseudorandom Functions (PRF), a family of functions  $\{f_k\}$  that are indistinguishable from truly random function, i.e.  $f_{g^{xy}}(0), f_{g^{xy}}(1), \dots$ , etc. We can use

padding to pad the id with different strings in order to make the input for the PRFs be the equal length.

Alice will produce a triple  $(x, y, z)$  where  $x = \text{Enc}'_{pk}(alice_{id}, session_{id})$ ,  $y = \text{Enc}_{K_e}(c_{id})$ ,  $z = \text{MAC}_{K_a}(alice_{id}, y)$ . On receiving the message  $(x', y', z')$ , the server DB will have a function to verify the uniqueness of message ‘identifier’  $\text{Dec}'_{sk}(x')$  and the validity of the MAC tag (computed on  $(x, y)$ ); if the tag checks succeed,  $y'$  is then decrypted under key  $K_e$  and the resultant plaintext accepted as a valid query  $c_{id}$ .

Upon receiving the decrypted query, the DB should be able to return the necessary fields to the doctor using the same key derived from the KE phase, and thus initiate another communication to the client(Alice). Also, the protocol may use a digital signature to verify the integrity of the content, i.e.  $SIG_{SK}(m)$ , where  $m$  is the messages that have been sent.

After all the necessary fields passed to the mobile device, the “expert system” will then compute the likelihood of the relevant complications for this patient. Alice, at this point, should be able to make a decision for a treatment to Bob. Moreover, the new decision/treatment will then be sent back to the corresponding medical record in the hospital’s DB. Relevant content written up to the DB will be encrypted using the previous key (from the KE phase); also a signature will be appended to this decision for the DB’s verification.

### 3.5 Security Analysis

We give an overview of how our protocol accomplishes the security goal of protecting the patient’s information, i.e. confidentiality, integrity, and authenticity. For each interaction between the doctor and the hospital’s database, we will consider the a ‘session’ as stated in section 3.4. The key exchange part will implicitly achieve mutual authentication where the two parties will hold a shared key which looks ‘random’ to the adversary in the middle.

Sessions are denoted by the name of the party holding the session and a session identifier. Here we simplify our presentation by implicitly referring to matching sessions as those that have the same session identifier. In practice this requires that parties create session identifiers interactively (before or during the KE run). Specifically, we assume the common practice where (as part of the protocol)  $A$  sends to  $B$  a value  $sid_A$ ,  $B$  sends to  $A$  a value  $sid_B$ , and they both define the session identifier as  $s = (sid_A; sid_B)$ . We say the protocol achieves mutual authentication secrecy if the following two properties are satisfied:

1. Except with negligible probability, the client (doctor) and the server (Database) will agree on a shared secret key for a unique session identifier.
2. Given the truly random key and the shared session key, there is no probabilistic polynomial adversary who can distinguish the session key from the truly random key with probability greater than half plus negligible.

We will refer the reader to the rigorous proofs for the KE protocol in [2, 3].

Once a shared key is established, the client and the server can start the conversation using the CCA-secure symmetric encryption, where the symmetric keys are derived from the shared key in the KE phase. As we claimed in Theorem 3.3.1, an encryption-then-authentication can build a secure channel, which is essentially a CCA-secure symmetric authenticated encryption. Therefore, the information we send in the clear has been fully protected, and there's a negligible probability that a ppt adversary can infer any information from this protocol assuming the underlying assumptions (such as Diffie-Hellman, digital signature, MAC) holds.

In addition, we also give another theoretical approach for this protocol. As the users in this system need credentials to login, we consider the client and the server may use the 'password' to initiate a communication. Therefore, we consider the password-based-

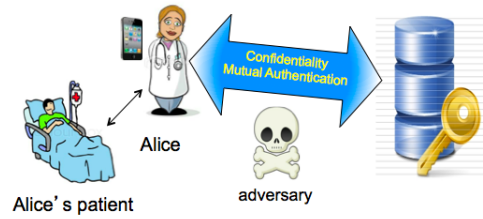


Figure 3.1: The communication protocol b/w doctor and the database

authenticated key exchange (**pake**) in this approach. Both the server and the client now possess a secret ‘password’. Specifically, the client will have a password associated with him/her, and the server has the corresponding password associated with his client maintained in the database. The server and the client will now agree on a shared key using the the pre-shared ‘password’. From the adversary’s point of view, for each session the shared key will be totally fresh and random to him. For an adversary (man-in-the-middle), the security of the **pake** is resistant to off-line dictionary attacks, i.e. any adversary who tries to enumerate the computation to guess the password off-line will fail in distinguishing the session shared key from random. Since passwords are chosen from a small space, an adversary can always try each possibility one at a time in an impersonation (on-line) attack. Thus, we say a protocol is secure (informally) if this exhaustive guessing is the best an adversary can do. For a real-world adversary, such on-line attacks are the hardest to mount, and they are also the easiest to detect. It is very realistic to assume that the number of on-line attacks an adversary is allowed is severely limited, while other attacks (eavesdropping, off-line password guessing) are not. There have been many settings proposed for building **pake** ([12, 7, 13, 11]), and here we just mention that **pake** would be a possible way for the client/doctor and the Database to bootstrap a high-entropy shared key using their pre-shared password.

## Chapter 4

### Deployment on a iOS device

We implements our secure protocol on the iOS platform, currently, the app can be run under iOS 6.0. Suppose doctor Alice wants to check all of the most likely complications for her patient (see Fig. 4.2). Alice must acquire the significant data in the patient's record from the hospital's DataBase (DB). She uses this data to compute the complication(s) using an existing 'expert' system app on her phone. The application will produce a checklist which appears on the surgeon's wireless device (see Fig. 4.3). Alice will then address specific issues that need to be acted on to minimize the effects of the post-operative complication(s). Her responses will, in a seamless and wireless fashion, communicate with the electronic medical record (EMR) and lead to appropriate orders being generated (see Fig. 4.4b). Alice will finally commit to her decision on the specific complications. She will update the patient's EMR on the hospital's DB.

#### Data Structures

*PatientCoreData*: PatientCoreData class is used to hold the data. When a patient's data has been 'scanned' to the phone, we convert this data entry into the PatientCoreData type on the phone. PatientCoreData is the basic data structure where we hold the patient's information. The complication calculation will get access to this data, therefore, conduct the computation.

*ExpertSystem*: The patient data will then be passed to the expert system. The 'expert

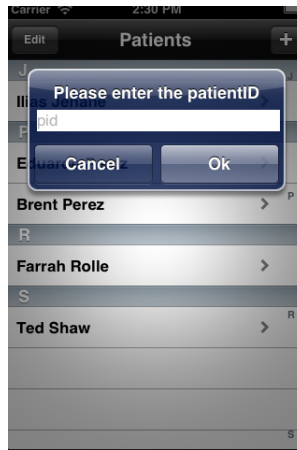


Figure 4.1: Prompt Patient ID View

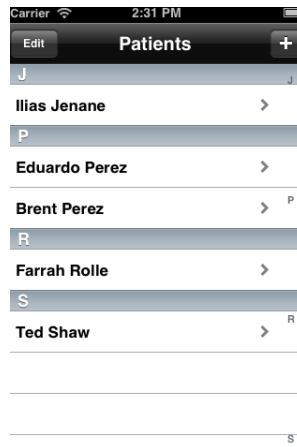


Figure 4.2: List of Patient View

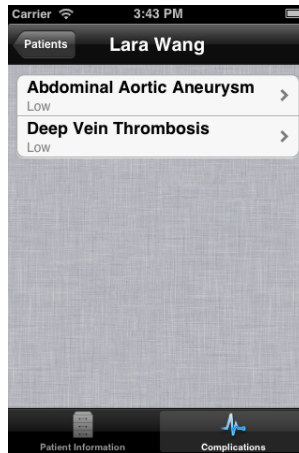


Figure 4.3: Patient's Complication View

system' will calculate the post-surgical risk level for the patient. Currently, the *dvtRiskCalculator* maintains a sample complication calculation for Deep Vein Thrombosis. Its identifiers are Age, Gender, BMI, Walking, Congestive Heart Failure. Risk Levels are Low, Medium, High.

*PatientInit*: We use the UIAlertView in the iOS to detect the input of the patient id *p.id*. When a doctor inputs some *p.id*, the button action listener will call the *PatientInit* to initialize the connection between the iPhone and the server. Several cryptographic classes and methods are being used in this phase. The connection will then send out the authentication encrypted query to the server. The server side will verify the message authenticated code and decrypt the ciphertext based on the established shared key (see section 3.4). (We use the *CommonCrypto.m* from iOS and *OpenSSL* to implement the encryption scheme.)

**Views**: We only present a sample TVC complication view for this presentation, while other complications views are very similar to the one presented.

*PatientInformationTVC*: Displays the patient information as queried from the database where dictionary keys are table section headers and the object for the key is displayed in the cell. It controls one of the views of the tab controller segued from *PatientTableView-*

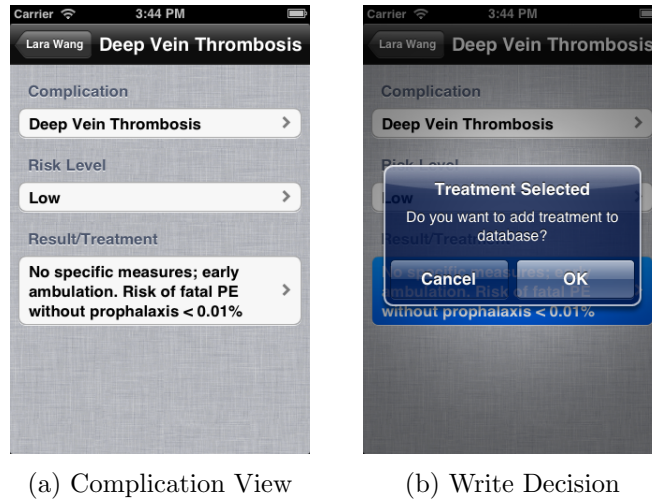


Figure 4.4: Doctor's decision to the patient's complication will added to the Database Controller. The controller does not initiate any segues.

*PatientTableViewController*: Controller for the view that displays the listing of patients stored in CoreData. Subclass of CoreDataTableViewController which controls much of the interaction with the CoreData model. Also implements editing and adding of patients to the database. This controller contains the ExpertSystem model by which patients are initialized. Main view of the application and segues to the tab view (no explicit controller) which contains PatientInformationTVC and ComplicationTVC

*ComplicationTVC*: Displays (Fig. 4.3) the patient's complications as calculated by the expert system. Each complication is represented by one cell. It controls one of the views of the tab controller segued from PatientTableViewController. The controller initiates the segue to TreatmentTVC view.

**The Medical Record Database**: For completeness, we have used a simple database to test our implementation. The database was written in MySQL on the server. We have the following main figure (4.5) mainly for the post-surgical database: `patient_comps_history`, `patient_comps`, `patient_data`). The iPhone app will first interact with the Fig( 4.5c) (the patient information) table to retrieve the patient's information. When all the data securely

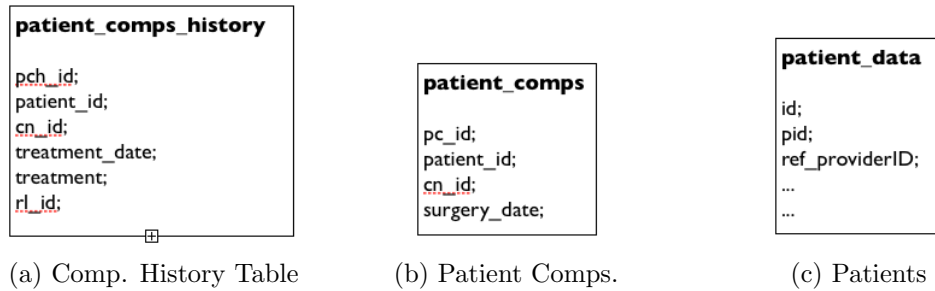


Figure 4.5: DataBase Tables

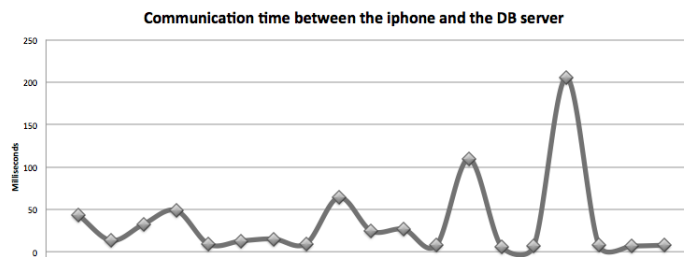


Figure 4.6: Communication time on local machine between iPhone and Java Server

passed to the app, the app will perform the computation and show the result to the surgery professionals and wait for the professional to update the complication decision. Then, the app's expert system will securely update the complications tables (Fig 4.5a, 4.5b) in the patient's medical records.

## 4.1 Performance

We tested our program on the local machine (MacPro with 2.2 GHz Intel Core i7) using the local Xcode iPhone simulator and a JAVA Server to control the sample database. The performance (Fig 4.6) shows that the actual communication time between the iPhone simulator and the server is very fast, mainly because the communication takes place in the same machine. On the other hand, we note that the cryptographic techniques that has been deployed in this protocol do not significantly increase the communication time overhead. (The practical cryptographic tools we use here are AES256 for encryption and

HMAC-SHA256 for authentication. Fig .4.6 fluctuates mainly because the machine CPU may be busy with some other local jobs.)

## Chapter 5

### Open problems and Future work

We'll address future work and several open problems in this section.

Our main concern in this thesis is the security aspects of the medical record. We provide a protocol for protecting the medical data using current cryptographic tools. However, in our implementation of this app on iPhone, the expert system has not been fully designed. Current solution and analyzation on different types of complications should be well-embedded into the expert system. As we are not medical surgical expert, we will leave the 'expert system' as a part of the open and future work for the medical experts and researchers. Also, it'll be more interesting to address of role-based access to the EMR data. One can use the cryptographic protocol based on attribute-based encryption (ABE). We note the reader that ABE will be quite useful when doctors gaining different informations on their patients, and, on the other hand, the patient won't reveal other irrelevant information their their surgical professionals. When a record is created, each node within the record is evaluated by a policy engine, which can be the hospital's database in our case. The database determines whether encryption is necessary, and derives a policy and a set of attribute tags that are appropriate for the record. The operation of the policy engine is flexible, and can be configured according to institutional requirements. If the record is marked for encryption, it can be encrypted using ABE ciphertext-policy or encrypted using a key-policy ABE scheme under a set of attributes identified by the policy engine (These may include record type, patient age, date, and other non-sensitive attributes related to

the record).

On the other hand, as we're interested in the security and privacy-preserving aspect of the sensitive medical data, there are the following future direction. As current service could become popular, the third party cloud computing service can provides more powerful capability of computing the data. When the third party computation is involved, the privacy and security causes new issues to arise. The hospital may be use the cloud as a service to maintain the data, while all the medical record will be encrypted in the database. An honest client can still query this untrusted party without revealing his/her information. There are known technique like Private Information Retrieval (PIR) and Delegation Computation that can possibly solve this problems. It will be interesting if we could use cloud services such as Google Health or Microsoft HealthVault to provide our post-surgical computation for determining all possible complications.

## Bibliography

- [1] Mihir Bellare<sup>1</sup> and Phillip Rogaway. *Introduction to Modern Cryptography*. 2005.
- [2] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Europe Cryptology*, pages 453–474, 2001.
- [3] Ran Canetti and Hugo Krawczyk. Security analysis of ike’s signature-based key-exchange protocol. *International Association for Cryptologic Research Cryptology ePrint Archive*, 2002:120, 2002.
- [4] Rui Chen, Noman Mohammed, Benjamin C. M. Fung, Bipin C. Desai, and Li Xiong. Publishing set-valued data via differential privacy. *Proceedings of Very Large DataBase*, 4(11):1087–1098, 2011.
- [5] Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. Differentially private spatial decompositions. In *International Conference on Data Engineering*, pages 20–31, 2012.
- [6] Shafi Goldwasser, Silvio Micali, and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *ACM Symposium on Theory of Computing*, pages 365–377, 1982.
- [7] Adam Groce and Jonathan Katz. A new framework for efficient password-based authenticated key exchange. In *ACM Conference on Computer and Communications Security*, pages 516–525, 2010.
- [8] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of Very Large DataBase*, 3(1):1021–1032, 2010.
- [9] Yuan Hong, Jaideep Vaidya, Haibing Lu, and Mingrui Wu. Differentially private search log sanitization with optimal output utility. In *International Conference on Extending Database Technology*, pages 50–61, 2012.
- [10] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography: Principles and Protocols*. Chapman and Hall/CRC, 2007.

- [11] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient password-authenticated key exchange using human-memorable passwords. In *Europe Cryptology*, pages 475–494, 2001.
- [12] Jonathan Katz and Vinod Vaikuntanathan. One-round password-based authenticated key exchange. *International Association for Cryptologic Research Cryptology ePrint Archive*, 2010:368, 2010.
- [13] Jonathan Katz and Vinod Vaikuntanathan. Round-optimal password-based authenticated key exchange. In *Theory of Cryptography Conference*, pages 293–310, 2011.
- [14] Hugo Krawczyk. Sigma: The ‘sign-and-mac’ approach to authenticated diffie-hellman and its use in the ike-protocols. In *Cryptography Conference*, pages 400–425, 2003.
- [15] Chao Li and Gerome Miklau. An adaptive mechanism for accurate query answering under differential privacy. *Proceedings of Very Large DataBase*, 5(6):514–525, 2012.
- [16] Stavros Papadopoulos, Spiridon Bakiras, and Dimitris Papadias. Nearest neighbor search with strong location privacy. *Proceedings of Very Large DataBase*, 3(1):619–629, 2010.
- [17] Russell Paulet, Md. Golam Kaosar, Xun Yi, and Elisa Bertino. Privacy-preserving and content-protecting location based queries. In *International Conference on Data Engineering*, pages 44–53, 2012.
- [18] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Special Interest Group on Management of Data Conference*, pages 735–746, 2010.
- [19] Elaine Shi, T.-H. Hubert Chan, Eleanor G. Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Network and Distributed System Security Symposium*, 2011.
- [20] Manolis Terrovitis, John Liagouris, Nikos Mamoulis, and Spiros Skiadopoulos. Privacy preservation by disassociation. *Proceedings of Very Large DataBase*, 5(10):944–955, 2012.
- [21] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. In *International Conference on Data Engineering*, pages 225–236, 2010.
- [22] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Ge Yu. Differentially private histogram publication. In *International Conference on Data Engineering*, pages 32–43, 2012.