Mobile Health Care on a Secured Hybrid Cloud

Khaled A. Nagaty

***Abstract*—This paper presents a secure mobile health application which is based on hybrid cloud architecture combined with cryptographic techniques to protect privacy, integrity and security of patients and health care givers data and with role based access control to authenticate and authorize users. Hybrid cloud platform combines the advantages of both the private cloud which guarantees privacy and safety of data and the public cloud which provides a platform for reduced services costs. Integrating cryptography and role based access control with hybrid cloud computing ensures the safety of patients' medical records and enable user authentication and authorization for access control. This integrated technology can provide the mobile health care the required safety and privacy to flourish.**

***Index Terms*—Cloud computing, cryptography, cloud security, mobile health, hybrid cloud**

I. Introduction

IN cloud computing, storage and processing resources are all located remotely where users share these resources in a controlled and efficient manner with the illusion of computer network and storage capacity. Cloud computing resources can be used to run users applications and can be released when they are no longer needed. Users can acquire storage and processing resources as demanded which helps organizations with limited budget to lower their IT costs. Although cloud computing is becoming more popular among users but the number of cloud subscribers are beyond expectations as indicated by a survey conducted by the International Data Corporation [1]. Most IT Executives and CEOs are not interested in adopting cloud services due to the risks associated with security and privacy because cloud datacenters are managed by cloud providers and not owned or managed by cloud users which raises many concerns about security, data privacy, confidentiality, accidental or malicious attempts. Security on cloud storage should be provided to cloud users as a service with minimal additional cost. In [2] the authors present privacy as a service (PaaS) where security protocols which enforce user data privacy, legal compliance and user trusted need to be considered at every phase of cloud computing architectures. PasS allows for secure storage and processing of users' confidential data by leveraging the tamper-proof capabilities of cryptographic coprocessors. Huge efforts have been put to build secure cloud computing environments and infrastructures in order to guarantee secure communication and more specifically authentication, integrity and confidentiality between users and processes in the cloud. The fundamental operations for security on cloud computing are encryption and role based access control (RBAC). Encryption transforms data from something understandable to an intruder into something not understandable. Encryption helps users to check if their data had been tampered with or not which means that encryption provides the cloud with data integrity. RBAC supports integrity which means that data and processes must be modified only in authorized ways and by authorized users. In this manner, integrating cloud computing with cryptography and RBAC can help increasing privacy, confidentiality and protecting data in the cloud which encourage more security and privacy concerned. A model for a secured mobile health system which integrates cryptography, RBAC and hybrid cloud is discussed in section II, section III is dedicated for analysis while section IV is dedicated for conclusions.

Manuscript received January 30, 2014.

K. A. Nagaty is with the Computer Science Department, British University in Egypt, El Sherouk City, Egypt , on leave from Faculty of Computer Science and Information Systems, Ain Shams University, Abbassia, Egypt (phone: +2-0100-1012413; fax: (+202)-2687-5889; e-mail: khaled.nagaty@ bue.edu.eg).

II. SYSTEM ARCHITECTURE

Mobile health based on cloud computing is revolutionizing the way healthcare is provided to patients. Applications of mobile health installed on mobile devices such as mobile phones, tablet computers and personal digital assistants (PDAs) help patients to continuously receive healthcare and medical treatment anytime, anywhere and anyhow and help practitioners to real-time monitoring their patients and direct provision of healthcare. Usually, mobile health applications store health information in an electronic health records system (EHR). Due to the limited resources of mobile devices such as low CPU, low memory, low storage capacity and battery powered environment, mobile health clients should use the unlimited resources of the cloud providers such as unlimited storage and high processing power of the cloud servers. EHR systems can increase their storage capacity by uploading the medical information to the cloud storage where healthcare practitioners in different hospitals and medical institutions can share the medical records of the patients in order to make correct diagnosis and give them the proper medication they need. EHR systems and mobile health applications should be characterized by scalability, security, non-repudiation and availability in a ubiquitous interoperable manner in order to provide patients with quality on demand healthcare services [3]. Scalability refers to the ability of the mobile health systems to handle the growing demand on accessing the EHR

system. Users of mobile health applications are interested in storing their medical and personal information on cloud servers unexposed. Security challenges which may be caused by intruders must be targeted in order to provide a secured mobile health environment include confidentiality, integrity and authentication. Confidentiality ensures that information is accessible only to those authorized to have access to it and never disclosed to unauthorized entities. Integrity ensures that patient medical information during its transfer is not modified or corrupted by unauthorized accesses to ensure patient's safety. A security framework should provide privacy and security features to all mobile health users [4] and must be highly scalable to be adaptively handled without degradation in performance. Security also involves auditing, where an independent third party systematically examines medical records activities, unauthorized accesses, modifications and performance [5]. Non-repudiation is an important feature to ensure that a message sender cannot deny having sent this message. Availability of health information is critical to effective healthcare delivery system. Availability of a service is defined as the property of being accessible and useable upon demand by an authorized entity [8]. Sharing medical information with practitioners across many administrative boundaries can violate patient information confidentiality, integrity and privacy [6], [7]. Authentication enables a party to ensure that he/she is communicating with the correct party on the other side. Without authentication, unauthorized access to resources and confidential information could be gained by an unauthorized party. Role Based Access Control (RBAC) is a scheme that restricts unauthorized users to access a system. It offers a satisfactory level of safety & security for authorized mobile health users who accessing EHR system resources & medical records through a set of rules & policies put into effect for patients, doctors or any other personnel in the form of login & password [9]. In [10] the authors proposed an EHR sharing and integration system in healthcare clouds and analyze the arising security and privacy issues in access and management of EHRs. Securing mobile health applications running on a mobile device is therefore an important area, if we want the applications to be trustworthy and reliable [11]. A mobile health cloud computing system consists of three main components which are [1]:

A. Mobile Client

Clients of mobile health applications utilize the storage services and the processing power provided by the cloud service provider. Applications of mobile health are embedded in the mobile devices of clients to interact with EHR system on the cloud server using mobile networks. Mobile networks are composed of three parts [12]:

--Base stations: send and receive transmissions to and from clients.

--Mobile telephone switching offices: transfer calls between national or global phone networks and base stations.

--Subscribers (clients): connect to base stations by using communication devices.

Mobile devices should register to a carrier service or provider which is licensed to offer services within a certain geographic areas. If a mobile device is out of the geographic area of its provider then roaming occurs where service connectivity can be extended in locations far from the base station where the service was first registered. Two technologies are used to improve digital communications: Time Division Multiple Access (TDMA) which divides each channel into six time slots one for transmission and one for reception. Code Division Multiple Access (CDMA) which transmits multiple encoded messages over a wide frequency and decode them at the receiver. CDMA is very useful in mobile health systems [13].

B. Cloud Service Providers

The cloud service provider efficiently manages, operates, maintains, allocates cloud resources and provides mobile health users security services such as intrusion detection, key management, encryption, authentication and authorization. The healthcare cloud system is consisted of public and private cloud providers where hospitals can construct besides the public cloud environment their own private cloud environment. Medical information for patients can be stored in both private cloud server and public cloud server [10]. Auditing can be implemented using a cloud service monitor which records clients' activities such as accessing or modifying medical records chronologically.

C. Trusted Third Party

A trusted third party is needed for configuring and installing tamperproof crypto-coprocessors on the cloud. Multiple registered mobile clients are associated with each crypto-coprocessor. The crypto-coprocessor distributes secret key (SK) with associated mobile clients and generates a message authentication code on behalf of mobile clients. Fig. 1 shows the architecture of mobile health hybrid cloud computing environment.

D. EHR Phases

A hybrid cloud EHR scheme is composed of the following phases [10]:

--New Electronic Record Creation.

--Electronic Medical Record Access.

--Emergency Access for Electronic Medical Record.

Our mobile health system that depends mainly on using mobile devices to create and access medical records on a secured hybrid cloud has the above mentioned phases [14].

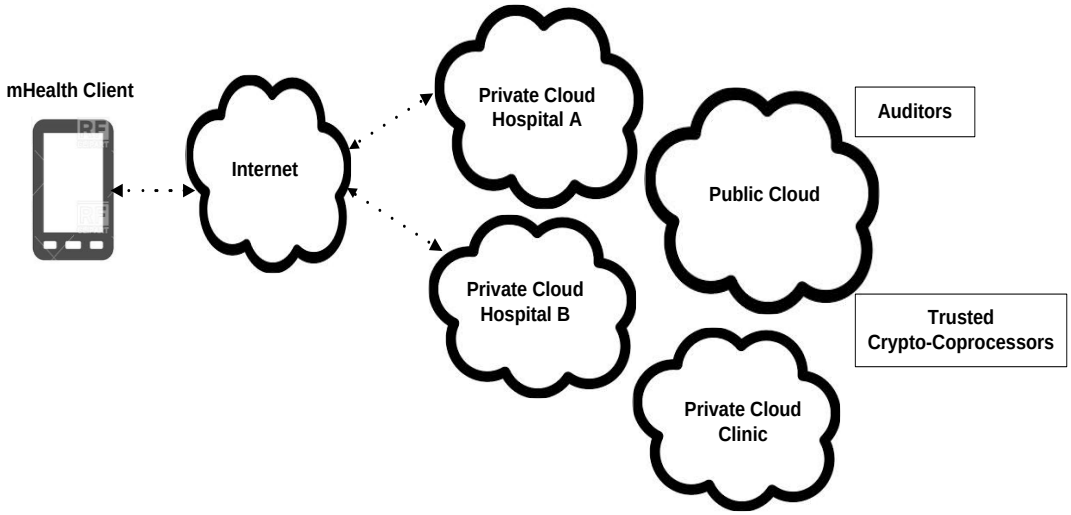

Fig.1. Architecture of mobile health hybrid cloud computing environment

Phase 1: Mobile Health Medical Record Creation

The hospital owns the medical information of patient (i) creates a new medical record for this patient. The medical record consists of: *Patient ID* (ID_i) and *Medical record file* ($file_{\mathrm{i}}$). The medical file corresponding license ACL_i is composed of: *Patient ID* (ID_i), *Patient's Right* ($pRight$), *Patient Secret Key* (pSK). The patient secret key (pSK) is generated using the patient ID_i and identity based encryption (IBE). The patient medical file $file_{\mathrm{i}}$ is encrypted by the patient secret key (pSK) to generate the encrypted medical file $Mfile_{\mathrm{ci}}$. Identity based cryptography uses the patient ID_i to generate the patient's public key (e_{i}) that can be used for encrypting information before transmission. The trusted third party, called the private key generator (PKG), generates the corresponding patient's private key (d_{i}). For the purpose of data privacy and confidentiality the generated corresponding license ACL_{i} is double encrypted firstly by the patient public key (e_{i}) to generate $eACL_{\mathrm{ci}}$ and secondly by the hospital public key (h_{public}) to generate $hACL_{\mathrm{ci}}$ as follows:

$$Mfile_{\mathrm{ci}} = E_{\mathrm{pSK}}(file_{\mathrm{i}}) \quad (1)$$

$$Hfile_{\mathrm{ci}} = E_{\mathrm{h_{public}}}(Mfile_{\mathrm{ci}}) \quad (2)$$

$$eACL_{\mathrm{ci}} = E_{\mathrm{e_i}}(ACL_{\mathrm{i}}) = E_{\mathrm{e_i}}\left(ID_{\mathrm{i}}||pRight||pSK\right) = \left(ID_{\mathrm{i}}||pRight||pSK\right)^{\mathrm{e_i}} \ mod\ N \quad (3)$$

The $eACL_{\mathrm{ci}}$ is then encrypted by the hospital public key (h_{public}) as follows:

$$hACL_{\mathrm{ci}} = E_{\mathrm{h_{public}}}(eACL_{\mathrm{ci}}) \quad (4)$$

The hospital signature $sACL_{\mathrm{ci}}$ is generated as follows:

$$sACL_{\mathrm{ci}} = E_{\mathrm{h_{public}}}\left(H(eACL_{\mathrm{ci}})\right) \quad (5)$$

Where: H is cryptographic hashing function such as *MD5*, *SHA1* or *SHA2* and h_{public} is the patient's hospital public key. The hospital private cloud generates a corresponding signature as follows:

$$ScID = E_{\mathrm{h_{public}}}(ID_{\mathrm{i}}, time) \quad (6)$$

The $Hfile_{\mathrm{ci}}$,$hACL_{\mathrm{ci}}$, $sACL_{\mathrm{ci}}$ and $ScID$ are all transmitted on a secure communication channel from the hospital's private cloud server to be stored on the public cloud server. The signature $ScID$ is stored in the public cloud for auditing purposes.

Algorithm 1: Mobile Health Medical File Creation

After finishing examination of a patient the doctor either updates the medical file of this patient or creates a new file on the patient's mobile device. Using this algorithm the patient encrypts the created or updated medical file and its corresponding license on his mobile device.

Step 1: The patient's mobile device must be authenticated by its associated trusted crypto-coprocessor in order to access the hospital private cloud.

Step 2: If authentication is succeeded the RBAC method validates the identity of the patient and checks his authorized role.

Step 3: If both validation and authorization of the patient are succeeded the doctor can update the medical file or create a new one on the patient's mobile device.

Step 4: After finishing updates the patient uses his secret key (pSK) to encrypt the created or updated medical file in order to generate $Mfile_{\mathrm{ci}}$ using eq.(1) and uses his public key (e_{i}) to encrypt the corresponding license in order to generate $eACL_{\mathrm{ci}}$ using eq.(3).

Step 5: The mobile device sends patient's ID_{i}, $Mfile_{\mathrm{ci}}$ and $eACL_{\mathrm{ci}}$ to the trusted crypto-coprocessor associated with it in order to send them to the patient's hospital private cloud.

Step 6: The hospital private cloud re-encrypts both the patient's encrypted medical file $Mfile_{\mathrm{ci}}$ and the encrypted corresponding license $eACL_{\mathrm{ci}}$ with the hospital's public key h_{public} using eq.(2) and eq.(4) respectively in order to generate a double encrypted patient's medical file $Hfile_{\mathrm{ci}}$ and double encrypted corresponding license $hACL_{\mathrm{ci}}$. In the mean time, the hospital's private cloud uses a cryptographic hash function H to generate a message digest $H(eACL_{\mathrm{ci}})$ for the encrypted license $eACL_{\mathrm{ci}}$ and encrypts this message digest using the hospital's public key h_{public} using eq.(5) in order to generate the hospital signature $sACL_{\mathrm{ci}}$. The hospital signature $sACL_{\mathrm{ci}}$ is used to verify that the encrypted license $eACL_{\mathrm{ci}}$ was not modified or tampered with by any intruder while it was being stored in the public cloud.

Step 7: The hospital's private cloud generates a corresponding signature $ScID$ using the patient's ID_i and system's time as in eq.(6). $Hfile_{ci}$, $hACL_{ci}$, $sACL_{ci}$ and $ScID$ are all transmitted on a secured communication channel to be stored in the public cloud. The $ScID$ is stored in the public cloud for auditing purposes.

Fig. 2 shows the block diagram for the interaction between a patient's mobile device, trusted crypto-coprocessor, the public and private cloud servers for creating an electronic medical file for the patient.

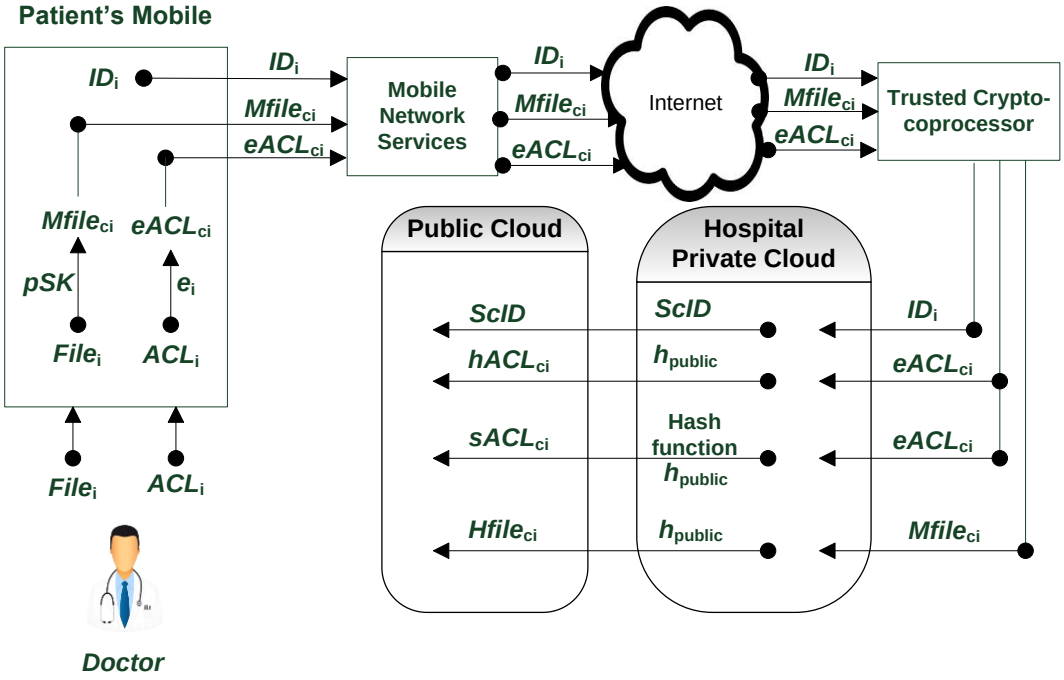

Fig. 2. Interaction between the patient's mobile device, its trusted crypto-coprocessor, public and private cloud servers for mobile health medical file creation

Phase 2: Mobile Health Medical File Access

We have three situations:

--Same hospital access situation: a doctor in a hospital wants to access a patient electronic health file $file_i$ that was created by the same hospital using the patient's mobile device.

--Cross hospital access situation: a doctor from another hospital wants to access the medical file of a patient that is owned by another hospital. This can be done after being granted permission from the owner hospital.

--Emergency access situation: the requesting emergency center can bypass the patient's permission to decrypt the medical file in emergency cases.

Algorithm 2: Same Hospital Access Situation

This algorithm is used when a patient wants to access the medical file that was created and owned by the same hospital in order to be investigated by the doctor.

Step 1: The patient's mobile device must be authenticated by its associated trusted crypto-coprocessor in order to access the hospital's private cloud.

Step 2: If authentication succeeded the RBAC method validates the identity of the patient and checks his authorized role.

Step 3: If validation and authorization are both succeeded the mobile health application installed on the patient's mobile device invokes the token generator to generate a "*token"*.

Step 4: The "*token"* and patient's (ID_i) are both sent to the trusted crypto-coprocessor which sends them to the hospital's private cloud.

Step 5: The hospital's private cloud generates a signature $ScID$ and sends the "*token"* and $ScID$ to the public cloud.

Step 6: $ScID$ is stored in the public cloud for auditing purposes while the "*token*" is used to retrieve the double encrypted patient's medical file $Hfile_{ci}$, double encrypted corresponding license $hACL_{ci}$ and the hospital signature $sACL_{ci}$ from the public cloud.

Step 7: The public cloud sends $Hfile_{ci}$, $hACL_{ci}$ and $sACL_{ci}$ to the hospital's private cloud which uses the hospital's private key $h_{private}$ to restore both the encrypted medical file $Mfile_{ci}$ and the corresponding encrypted license $eACL_{ci}$ as follows:

$$Mfile_{ci} = D_{h_{private}}(Hfile_{ci}) \quad (7)$$

$$eACL_{ci} = D_{h_{private}}(hACL_{ci}) \quad (8)$$

Step 8: The hospital's private cloud verifies the signature $sACL_{ci}$ which if verified both $eACL_{ci}$ and $Mfile_{ci}$ are sent to the patient's mobile device where the patient uses his private key (d_i) to decrypt $eACL_{ci}$ in order to restore the corresponding license ACL_i and uses his secret key (pSK) in order to restore his medical file $file_i$ as follows:

$$ACL_i = D_{d_i}(eACL_i) \quad (9)$$

$$file_i = D_{pSK}(Mfile_{ci}) \quad (10)$$

Step 9: The doctor can read the restored patient's medical file $file_{\mathrm{i}}$ and the corresponding license ACL_{i} on the patient's mobile device.

Fig. 3 shows the block diagram for retrieving a patient's medical file which was created and owned by the same hospital.

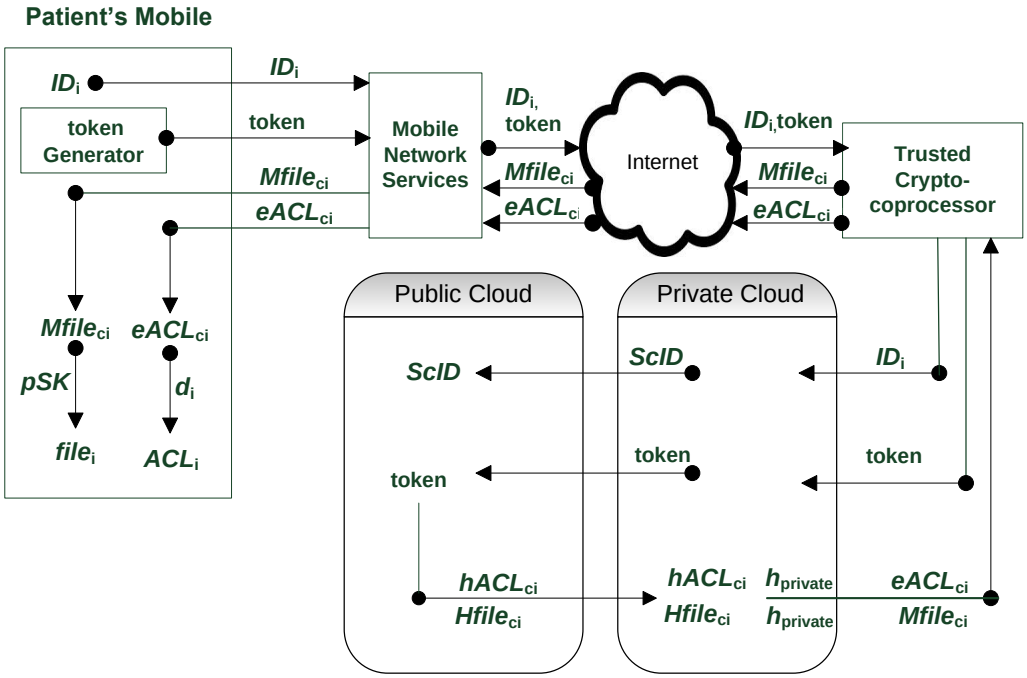

Fig. 3. Interaction between the mobile device, trusted crypto-coprocessor and the cloud server in order to retrieve the medical file of a patient

Algorithm 3: Cross Hospital Access Situation

This algorithm is used when a doctor from hospital "A" wants to access using his mobile device a patient's medical file that was created and owned by hospital "B".

Step 1: The mobile device at hospital "A" authenticates itself with its associated crypto-coprocessor "A".

Step 2: If authenticated the RBAC of hospital's "A" private cloud validates the identity of the doctor and checks his authorized role.

Step 3: If both validation and authorization are succeeded then the doctor from hospital "A" signs the patient ID_{i} to generate a doctor's signature $SdOC$ and generates a message Msg to be sent to the required patient.

Step 4: A doctor's signature is generated using equation (11), where the doctor uses his private key (d_{doc}) to encrypt the message digest $H(ID_{\mathrm{i}})$.

Step 5: The doctor's mobile device at hospital "A" sends patient ID_{i} , Msg, $SdOC$ and the doctor's public key (e_{doc}) to its associated trusted crypto-coprocessor "A" which in turn sends them to hospital's "A" private cloud where the patient ID_{i} is used to generate $ScID$ signature.

Step 6: The hospital's "A" private cloud sends $ScID$, ID_i , Msg , $SdOC$, hospital's "A" public key ${h^{\mathrm{A}}}_{\mathrm{public}}$ and e_{doc} to the public cloud.

Step 7: The public cloud searches for the hospital that owns the double encrypted medical file and double encrypted corresponding license of the required patient. We assume it to be hospital "B".

Step 8: The public cloud stores $ScID$ for auditing purposes and sends ${h^{\mathrm{A}}}_{\mathrm{public}}$, e_{doc} , $SdOC$, Msg and ID_{i} to hospital's "B" private cloud.

Step 9: The private cloud of hospital "B" verifies doctor's signature using ID_{i} and $SdOC$. To verify a doctor's signature, the private cloud of hospital "B" uses the doctor's public key e_{doc} to restore $H(ID_{\mathrm{i}})$, uses $SdOC$ and the same cryptographic hash function H to compute the message digest of the received ID_{i}. If both messages digests matched then the doctor's signature is verified because we know that the doctor at hospital "A" is the only one who owns the private key d_{doc}.

Step 10: The RBAC of hospital's "B" private cloud validates the patient's identity and checks his authorized role, which if validated and authorized the private cloud of hospital "B" sends both e_{doc} and Msg to the patient's mobile device.

Step 11: If the required patient from hospital "B" agrees to share his medical file with the doctor from hospital "A" then the mobile health application embedded in the patient's mobile device invokes the token generator to generate a token "*token*".

Step 12: This "*token*" is sent to hospital's "B" private cloud which in turn sends it to the public cloud in order to retrieve the double encrypted patient's medical file $Hfile_{\mathrm{ci}}$ and the corresponding double encrypted license $hACL_{\mathrm{ci}}$.

Step 13: Hospital's "B" private cloud verifies the signature $sACL_{\mathrm{ci}}$ which if verified it uses the hospital "B" private key ${h^{\mathrm{B}}}_{\mathrm{private}}$ to decrypt both $Hfile_{\mathrm{ci}}$ and $hACL_{\mathrm{ci}}$ in order to restore $Mfile_{\mathrm{ci}}$ and $eACL_{\mathrm{ci}}$ using eq. (12) & eq. (13) respectively.

Step 14: Those files are both sent to the patient's mobile device at hospital "B" where the patient uses his secret key (pSK) to decrypt $Mfile_{\mathrm{ci}}$ using eq.(14) in order to restore his medical file $file_{\mathrm{i}}$ and uses his private key (d_{i}) to decrypt $eACL_{\mathrm{ci}}$ using eq.(15) in order to restore the corresponding license ACL_{i}.

Step 15: The patient uses the doctor's public key (e_{doc}) to encrypt both the medical file $file_{\mathrm{i}}$ and the corresponding license ACL_{i} using eq.(16) & eq. (17) to generate $efile_{\mathrm{di}}$ and $eACL_{\mathrm{di}}$ respectively.

The following set of equations is used:

$$SdOC = E_{\mathrm{d_{doc}}}(H(ID_{\mathrm{i}})) \quad (11)$$

$$Mfile_{\mathrm{ci}} = D_{\mathrm{h^B{}_{private}}}(Hfile_{\mathrm{ci}}) \quad (12)$$

$$eACL_{\mathrm{ci}} = D_{\mathrm{h^B{}_{private}}}(hACL_{\mathrm{ci}}) \quad (13)$$

$$file_{\mathrm{i}} = D_{\mathrm{pSK}}(Mfile_{\mathrm{ci}}) \quad (14)$$

$$ACL_{\mathrm{i}} = D_{\mathrm{d_i}}(eACL_{\mathrm{ci}}) \quad (15)$$

$$efile_{\mathrm{di}} = E_{\mathrm{e_{doc}}}(file_{\mathrm{i}}) \quad (16)$$

$$eACL_{\mathrm{di}} = E_{\mathrm{e_{doc}}}(ACL_{\mathrm{i}}) \quad (17)$$

The patient at hospital B sends both $efile_{\mathrm{di}}$ and $eACL_{\mathrm{di}}$ to hospital's "B" private cloud using the embedded mobile health application in the mobile device. Hospital's "B" private cloud double encrypt $efile_{\mathrm{di}}$ and $eACL_{\mathrm{di}}$ using hospital's "A" public key $h^{\mathrm{A}}{}_{\mathrm{public}}$ in order to generate $Hfile_{\mathrm{di}}$ and $hACL_{\mathrm{di}}$ using eq.(18) & eq.(19) respectively.

The following set of equations is used:

$$Hfile_{\mathrm{di}} = E_{\mathrm{h^A{}_{public}}}(efile_{\mathrm{di}}) \quad (18)$$

$$hACL_{\mathrm{di}} = E_{\mathrm{h^A{}_{public}}}(eACL_{\mathrm{di}}) \quad (19)$$

Hospital's "B" private cloud sends both $Hfile_{\mathrm{di}}$ and $hACL_{\mathrm{di}}$ to the public cloud which in turn sends them to hospital's "A" private cloud. Hospital's "A" private cloud uses its private key $h^{\mathrm{A}}{}_{\mathrm{private}}$ to decrypt both $Hfile_{\mathrm{di}}$ and $hACL_{\mathrm{di}}$ in order to restore $efile_{\mathrm{di}}$ and $eACL_{\mathrm{di}}$ using eq.(20) & eq.(21) respectively.

Then hospital's "A" private cloud sends $efile_{\mathrm{di}}$ and $eACL_{\mathrm{di}}$ to the doctor's mobile device at hospital "A". The doctor uses his private key (d_{doc}) to decrypt $efile_{\mathrm{di}}$ and $eACL_{\mathrm{di}}$ in order to restore the patient's medical file $file_{\mathrm{i}}$ and corresponding license ACL_{i} using eq.(22) & eq. (23) respectively.

The following set of equations is used:

$$efile_{\mathrm{di}} = D_{\mathrm{h^A{}_{private}}}(Hfile_{\mathrm{di}}) \quad (20)$$

$$eACL_{\mathrm{di}} = D_{\mathrm{h^A{}_{private}}}(hACL_{\mathrm{di}}) \quad (21)$$

$$file_{\mathrm{i}} = D_{\mathrm{d_{doc}}}(efile_{\mathrm{di}}) \quad (22)$$

$$ACL_{\mathrm{i}} = D_{\mathrm{d_{doc}}}(eACL_{\mathrm{di}}) \quad (23)$$

Fig. 4 shows the block diagram for a request issued by a doctor at hospital "A" to restore the medical file of a patient that was created and owned by hospital "B".

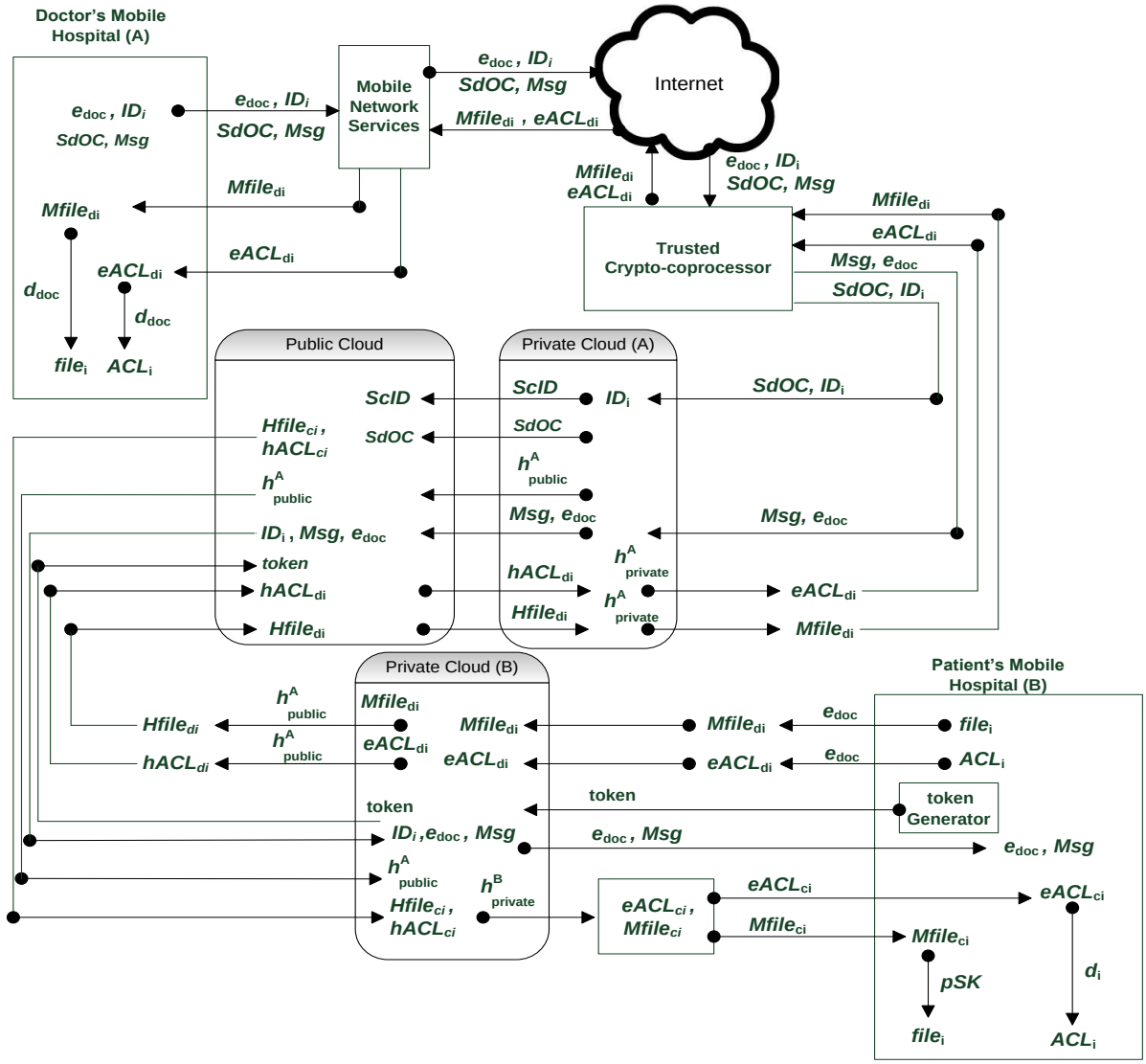

Fig. 4. Retrieval of a medical file that was created and owned by hospital "B" by a doctor from hospital "A"

Algorithm 4: Emergency Hospital Access Situation

If a patient in the emergency center can use his mobile device algorithm 2 can be used to retrieve the patient's medical file from the hospital owns this file for investigation by the emergency doctor. If a patient could not use his mobile device the emergency doctor can use this algorithm to access the medical file of the patient.

Step 1: The doctor's mobile device at the emergency center authenticates itself with its associated crypto-coprocessor.

Step 2: If mobile device is authenticated the RBAC at the private cloud of emergency center validates the identity of the emergency doctor and checks his authorized role.

Step 3: If both validation and authorization are succeeded the emergency doctor signs the patient ID_{i} to generate doctor's signature $SdOC$.

Step 4: The mobile device at the emergency center sends the emergency center public key em_{public}, $SdOC$ and ID_{i} to the PKG.

Step 5: The PKG validates doctor's signature which if validated the PKG uses its master private key to generate (d_{i}) which is the private key for the patient's ID_{i}.

Step 6: PKG encrypts the generated patient's private key (d_{i}) with em_{public} as follows:

$$d_{\mathrm{emi}} = E_{\mathrm{em}_{\mathrm{public}}}(d_{\mathrm{i}}) \quad (24)$$

Step 7: PKG sends d_{emi} to the emergency center private cloud which decrypts the d_{emi} using the emergency center private key em_{private} in order to restore the patient's private key (d_{i}) and sends it to the emergency center as follows:

$$d_{\mathrm{i}} = D_{\mathrm{em}_{\mathrm{private}}}(d_{\mathrm{emi}}) \quad (25)$$

Step 8: The private cloud of the emergency center sends a query that contains its public key em_{public} and patient ID_{i} to the public cloud in order to restore $file_{\mathrm{i}}$ and ACL_{i} for the patient in emergency.

Step 9: The public cloud searches for the patient's hospital which if found the public cloud sends em_{public} and patient ID_{i} to the patient's hospital private cloud.

Step 10: The patient's hospital private cloud RBAC method validates the identity ID_{i} of the patient. If validated the hospital's private cloud sends ID_{i} and em_{public} to the patient's hospital which invokes the token generator to generate *token*.

Step 11: The patient's hospital sends the *token* to the private cloud which consequently sends it to the public cloud.

Step 12: The public cloud uses the *token* to find $HFile_{\mathrm{ci}}$ and $hACL_{\mathrm{ci}}$ for the patient and sends those files to the patient's hospital private cloud.

Step 13: The patient's hospital private cloud uses its private key h_{private} to decrypt $HFile_{\mathrm{ci}}$ and $hACL_{\mathrm{ci}}$ as follows:

$$Mfile_{\mathrm{ci}} = D_{\mathrm{h}_{\mathrm{private}}}(Hfile_{\mathrm{ci}}) \quad (26)$$

$$eACL_{\mathrm{ci}} = D_{\mathrm{h}_{\mathrm{private}}}(hACL_{\mathrm{ci}}) \quad (27)$$

Step 14: The patient's hospital private cloud sends $Mfile_{\mathrm{ci}}$ and $eACL_{\mathrm{ci}}$ to the patient's hospital which encrypts them using the emergency center public key em_{public} as follows:

$$Efile_{\mathrm{em}} = E_{\mathrm{em}_{\mathrm{public}}}(Mfile_{\mathrm{ci}}) \quad (28)$$

$$EACL_{\mathrm{em}} = E_{\mathrm{em}_{\mathrm{public}}}(eACL_{\mathrm{ci}}) \quad (29)$$

The patient's hospital private cloud sends $Efile_{\mathrm{em}}$ and $EACL_{\mathrm{em}}$ to the public cloud which sends them to the emergency center's private cloud. The emergency center's private cloud uses its private key em_{private} to decrypt $Efile_{\mathrm{em}}$ and $EACL_{\mathrm{em}}$ as follows:

$$Mfile_{\mathrm{ci}} = D_{\mathrm{em}_{\mathrm{private}}}(Efile_{\mathrm{em}}) \quad (30)$$

$$eACL_{\mathrm{ci}} = D_{\mathrm{em}_{\mathrm{private}}}(EACL_{\mathrm{em}}) \quad (31)$$

The emergency center private cloud sends $Mfile_{\mathrm{ci}}$ and $eACL_{\mathrm{ci}}$ to the emergency center which uses the recovered patient's private key (d_{i}) to decrypt the $eACL_{\mathrm{ci}}$ as follows:

$$ACL_{\mathrm{i}} = D_{\mathrm{d_i}}(eACL_{\mathrm{ci}}) \quad (32)$$

The emergency center extracts the patient's secret key (pSK) from the decrypted corresponding license ACL_{i} in order to decrypt the encrypted medical record $MFile_{\mathrm{ci}}$ as follows:

$$file_{\mathrm{i}} = D_{\mathrm{pSK}}(MFile_{\mathrm{ci}}) \quad (33)$$

Fig.5 shows the block diagram for a doctor at an emergency center retrieving the medical file that was created and owned by a hospital.

III. Analysis

A. Security, Privacy and Availability

These are features provided by the proposed mobile healthcare system using hybrid cloud. They include:

– Privacy & Confidentiality: means that cloud providers do not learn about client's data and integrity means that users can detect any modification happened to their data.

– Integrity: means that data and processes must be modified only in authorized ways and by authorized users.

– Authentication & Authorization: enables a party to ensure that he/she is communicating with the correct party on the other side. Without authentication, unauthorized access to resources and confidential information could be gained by an unauthorized party.

− Non-repudiation: Non-repudiation is achieved by creating a signature $ScID = E_{h_{public}}(ID_i, \text{time})$ with each query of the patient's medical file. The signature contains the patient's ID_i and the time of request which is encrypted by the hospital's public key. This signature can be used in the auditing process.

− Auditing: is particularly useful to identify suspicious accesses and common access.

− Availability: The proposed mobile health system is highly available all year 24 x 7 because it uses the cloud technology of virtualization, resiliency, redundancy, data restoration and disaster recovery. This means that no service disruptions due to hardware failure, power outage or system upgrades or denial of service attacks can happen.

B. Privacy and Confidentiality

These features are improved by:

– Encrypting mobile devices that are used to transmit confidential information.

– Using crypto-coprocessors to authenticate mobile devices that want to communicate with cloud providers.

– Using the RBAC to control users' access which limits who can see what. The practice administrator identifies the users, determines what level of information is needed. For example, the physician, nurse and receptionist in a hospital, do not

access the same information because they have different responsibilities and tasks.

– Using IBE to generate patient's secret key, public and private keys which helps avoid keys collision.

– Using symmetric and asymmetric cryptography improves data privacy and confidentiality.

– Using the secured cloud storage to securely access the medical information in the cloud storage.

C. Psychological Effects

The proposed mobile healthcare system has positive psychological effects on both patients and doctors. The feel of safety and comfort as the patient knows that his medical records are kept confidential, his privacy is maintained, it is not possible for his medical records to be viewed or modified except by authorized people and his records are reviewed and audited only by authorized bodies. Doctors feel confident that they are dealing with the right patient with the correct medical file. They feel relieved when they know that their patients are monitored 24 hours per day and their patients can reach them at any time especially in case of emergency.

D. Improve Medical Services

Using a secured mobile health in cloud computing environment will speed up and improve medical services. With mobile health patients and their families can get health care service anywhere, anytime or anyhow. Doctors can keep monitoring their patients who in critical situations and give them medical advices. Mobile health devices can help elderly people by tracking their whereabouts and activities and allow them to call for help in case of emergency situations. Mobile health devices are smart enough to alert the patients against critical situations such as environments that may trigger allergy attack and can give real-time advice to them about treatments via mobile health applications. Mobile health systems can reduce the cost of healthcare as they provide accurate information on timely basis and on demand which improves the efficiency of medical care delivery. However, keeping privacy and confidentiality of personal information becomes important if these systems to flourish.

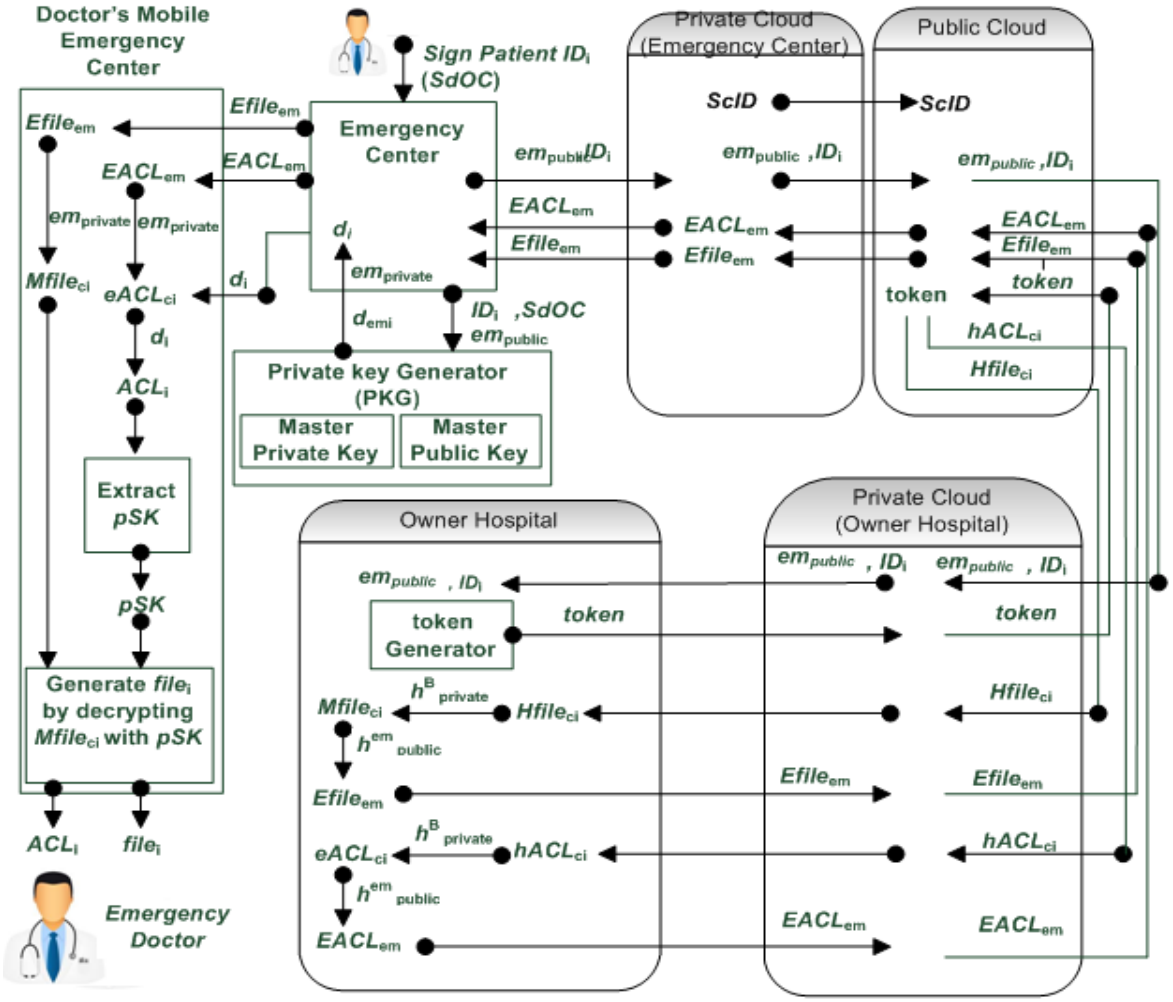

Fig. 5. Interaction between the emergency center, emergency center private cloud, owner hospital private cloud.

E. Limitations

Limitations for using the secured mobile health system include:

– Mobile devices can easily be misplaced, damaged, or stolen.
– Authenticating and encrypting new mobile devices that are used to transmit confidential information could be a long process and time consuming. Help can be provided to the user as required.
– Data can be hacked, manipulated, or destroyed by internal or external users, so security measures and ongoing educational programs must include all users.
– Some security measures that protect data integrity include firewalls, antivirus software, and intrusion detection software must be continuously upgraded.
– A full security program must be in place to maintain the integrity of the data, and a system of audit trails must be operational.
– A security officer must be designated to regularly evaluate security measures and identify the security weaknesses and threats; assign a risk or likelihood of security concerns in the organization and address them.
– Physicians mistakenly enter data on the wrong patient when working with an electronic list of patients and physicians have to choose a patient from an extensive listing.

IV. CONCLUSIONS

In this paper, we discussed the cryptographic primitives that can be used to protect data privacy, confidentiality and integrity in the cloud storage. We discussed the role based access control method (RBAC) that is used to authenticate and authorize cloud users. We also discussed the technology of the secured cloud storage which allows end users to securely access cloud storage. Finally, we presented a secured mobile health system that is based on hybrid cloud architecture and uses some of the cryptographic techniques. The proposed system uses crypto-processors to authenticate mobile devices which want to access the cloud servers and uses the identity based encryption to generate the patient's secret key, public key and private key. The secured storage technology is used to enable end users to securely access the requested information from cloud servers using generated tokens. Three modes for using the mobile health system were discussed: the first mode is the same hospital access which when a doctor from the same hospital that owns the patient's medical records wants to access the patient's data. The second mode is the cross hospital access which when a doctor from a hospital wants to access the medical record of a patient that is owned by another hospital. The third mode is in case of emergency which when a patient is entered to an emergency center and the doctor wants to access the patient's medical record that is owned by a hospital.

REFERENCES

1. A. N. Khana, M. L. Mat Kiah, S. Khanb, S. Madanic, "Towards secure mobile cloud computing: A survey", *Journal Future Generation Computer Systems*. 29, 2013, pp.1278–1299.
2. W. Itani, A. Kayssi, A. Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing on cloud Computing Architectures", in *Proc. 8th IEEE Int. Conf. Dependable, Autonomic and Secure Computing*, Chengdu, China, 2009, pp. 711-716.
3. P. Germanakos, C. Mourlas, G. Samaras, "A Mobile Agent Approach for Ubiquitous and Personalized eHealth Information Systems", in *Proc. 10th Int. Conf. User Modeling (UM'05) Proc. of the Workshop on Personalization for e-Health*, Edinburgh , 2005, pp. 67–70.
4. A. Yeratziotis, D. Van Greunen, D. Pottas, "A Framework for Evaluating Usable Security: The Case of Online Health Social Networks", in *Proc. 6th Int. Symposium Human Aspects of Information Security & Assurance*, *Interaction Design Foundation*, Greece , 2012, pp. 97-107.
5. JL Fernández-Alemán, IC Señor, PÁ Lozoya, A. Toval, "Security and privacy in electronic health records: A systematic literature review", *Journal of Biomedical Informatics* 46 (3), 2013, pp. 541–562.
6. R. Zhang, L. Liu, "Security models and requirements for healthcare application clouds". ", in *Proc. 3rd IEEE Int. Conf. Cloud Computing*, Florida, 2010, pp. 268-275.
7. H. Linden, D. Kalra, A. Hasman, J. Talmon, "Interorganization future proof HER systems-A review of the security and privacy related issues", *Int. J. Med. Inform.* 78, 2009, pp. 141–160.
8. ISO/EN 13606. Available: http://www.iso.org/iso/home.htm
9. J. Bacon, D. Evans, D. Eyers, M. Migliavacca, P. Pietzuch, B. Shand, " Enforcing End-to-End Application Security in the Cloud", LNCS 6452, Springer, 2010, pp. 293–312.
10. Y. Chen, J. Lu, J. Jan, "A Secure EHR System Based on Hybrid Clouds", *Journal of Medical Systems* 36, (5), 2012, pp. 3375-3384.
11. M. T. Nkosi, F. Mekuria, "Cloud Computing for Enhanced Mobile Health Applications", in *Proc. 2nd IEEE Itn.Conf. Cloud Computing*, Indianapolis 2010, pp. 629 – 633.
12. H. Bidgoli: Management Information Systems, Cengage Learning, Connecticut (2013).
13. D. Falconer, F. Adachi, B. Gudmundson,: Time division multiple access methods for wireless personal communications. IEEE Communications Magazine 33 (1), 1955, pp. 50-57.
14. K. A. Nagaty, "A Secured Hybrid Cloud Architecture for mHealth Care" In: "Mobile Health (mHealth): The Technology Road Map",LNCS, *Springer* , to be published.

Khaled A. Nagaty was born in Giza, Egypt on October, 1960. Nagaty earned B.Sc. in statistics from Cairo University, Giza, Egypt in 1982, MS.c. in computer science from Cairo University Giza, Egypt in 1990 and Ph.D. in computer science from Cairo University, Giza, Egypt in 1999. He is currently an Associate Professor at the Computer Science Department at the British University in Egypt, on leave from the Faculty of Informatics and Computer science at Ain Shams University. His Ph.D. dissertation was published by Lap Lambert publishers in a form of printed book in 2012. He has three book chapters published by IGI Global publisher and another book chapter to be published by Springer. He has many publications in international journals and conferences. His research interests include cloud computing, image processing, pattern analysis and cryptography. Dr. Nagaty is a reviewer for IEEE international conference on image processing (ICIP) and IEEE transactions on image processing.