J Med Syst (2016) 40: 1
DOI 10.1007/s10916-015-0365-5

MOBILE SYSTEMS

A Study on the Secure User Profiling Structure and Procedure for Home Healthcare Systems

Hoon Ko[1] · **MoonBae Song**[2]

Received: 13 March 2015 / Accepted: 2 October 2015 / Published online: 29 October 2015

Abstract Despite of various benefits such as a convenience and efficiency, home healthcare systems have some inherent security risks that may cause a serious leak on personal health information. This work presents a Secure User Profiling Structure which has the patient information including their health information. A patient and a hospital keep it at that same time, they share the updated data. While they share the data and communicate, the data can be leaked. To solve the security problems, a secure communication channel with a hash function and an One-Time Password between a client and a hospital should be established and to generate an input value to an OTP, it uses a dual hash-function. This work presents a dual hash function-based approach to generate the One-Time Password ensuring a secure communication channel with the secured key. In result, attackers are unable to decrypt the leaked information because of the secured key; in addition, the proposed method outperforms the existing methods in terms of computation cost.

Keywords Hash function · One-time password · Attacks · User profiling structure · Healthcare system

This article is part of the Topical Collection on *Mobile Systems*

✉ Hoon Ko
dr.hoonko@gmail.com

MoonBae Song
mbsong@gmail.com

[1] GECAD/ISEP/IPP, R. Dr. Antonio Bernardino de Almeida, 431, 4200-072, Porto, Portugal

[2] Information Technology & Mobile Communications Division, Samsung Electronics, Maetan3-dong, Suwon, Kyunggi-do, 443-742, Korea

Introduction

In the last few years, ubiquitous computing has been evolving into the new era of Internet of Things (IoT) with convergences of IT (Information Technology) - BT (Biotechnology) [1]. There are various applications such as mobile healthcare (mHealth) services, mobile home network, SNS and so on. The health services are able to support seamless monitoring of patient conditions using sensors in home, and provide real-time healthcare services [2]. To see the patient's vital signals and to detect their behaviour, wearable technologies such as wearable sensors and activity trackers will be used. Moreover, it easily can guess that air condition and indoor temperature, which is an effect of environmental factor to keep health condition also will be involved in healthcare systems that can realize their state in real-time. These functions have been emerged with various IoT devices in our daily life so far, it is to keep going continually. A well-designed system could have security holes. Security attacks on a healthcare system has a severe damage in terms of a social aspect [3, 4]. Possible attacks are a drain of a personal health record, an unacceptable modification of a personal health data and a communication inferring of between doctor and patient communication. Now, network-based healthcare system is increasing, Many emergency monitoring systems with an in-body sensor is built in a hospital. To use it, a patient has to communicate with a hospital, and then the data on this channel will be a target to an attacker. In some countries, each hospital has its own health data store for archiving personal health data, and transmits the data to a central server maintained country-scale data store [5]. As the result, a systematic approach for handling health-related personal data and protecting patient's privacy is critical to a secure healthcare system. This means that there is no attribute to protect the data in each structure to store their information.

To solve the problem, we propose a secure user profiling structure and a procedure which uses a one-time password to authenticate and a dual-hash function to keep an encryption/decryption key. The rest of the paper is organized as follows: Section 2 describes the security risks of an existing healthcare system. Section 3 explains a system model which introduces our proposed model for user profiling and procedure for home healthcare service. In Section 4, we discuss possible security attacks, solutions and cost comparison. Finally, we conclude in Section 5.

Related work and security problem

An expert looks out that 'It will be for a cyber-attackers year in 2015 in a healthcare industry' [6]. In comparison to other service domains, a security consideration is relatively slow down, but the financial value of the stolen health-related data is drastically increasing [6]. In February, 2015, Anthem, the second-largest health insurance company in the US, had announced that they were attacked by hackers, and personal and health-related data including names, birthdays, Social Security numbers (SSNs), medical IDs, email addresses and other sensitive personal information were stolen [7]. It appears that the company didn't apply to any encryption technology on the stored personal health records. The company usually uses an encryption only for the case of moving in or out of its database [8]. In July 2014, Community Health Systems (CHS) confirmed that personal health information on its 4.5 million patients was stolen as a part of a cyber-attack. The US government believes that these two attacks have originated in China [7]. During the last decade, major parts of a cyber-attack are focused on the area of finance and currency; however, as the industry keeps taking a preparatory action for possible a cyber-attack to their networks (e.g., DDoS attacks), hackers found a healthcare system as their new target. David Kennedy, founder and CEO of TrustedSec, pointed out that a vast amount of health-related data is available as healthcare information is rapidly digitized; however, the healthcare industry is relatively weak in security aspect. The stolen personal health information could be abused to an identity theft, a medical fraud, such as an illegal prescription and a financial fraud [9]. Art Coviello, CEO of RSA (the Security Division of EMC), sent an email to his customers' companies mentioning that a group of skilled hackers is preparing a massive cyber-attack to a healthcare company and its-related content provider company. ID Experts CEO Bob Gregg, also emphasized that personal information such as a name, an address, a Social Security number (SSN), health insurance information are able to be funded, and subjected to sell illegally. As in early-stage, the current healthcare systems do not support proper encryption to the stored health data and rely only on an ID and a password-based access control. In [10, 11], M. R. Ogiela and U. Ogiela had proposed a linguistic protocol, which is how to share, to keep and to manage their information safely. However, because each country has own language, also it has own characteristics, it has a potential trouble in flexibility and utilization. In result, there can be various security risks. Differentiated approach to security is needed for excessive security can incur the huge overhead of encryption/decryption of user data. That is, there are two big security problems in home healthcare systems: One is a personal information leak problem, the system in a hospital or in a medical centre usually has a low-level security configuration these days. Therefore an attacker can know how to attack by scanning simply and can connect to the system; finally they can get the information illegally. The other is non-encryption problem of the information. Although an attacker has the data illegally, if the data has been stored after encrypting, then it can avoid the worst case. However, still, many systems keep them without any security process like encryption.

System model

To study our suggestion, we first describe a scenario which can be a possible situation in our daily lives.

Scenario

An old guy is almost alone in his home. Although he lives with his son and grandchildren, usually they go out every morning to work or to study. That's why every day he is in a home alone. In addition, he has chronic diseases such as hypertension and diabetes, so he has to be deeply cared by another family or another friend or a healthcare system by a hospital. That is, someone has to care him every day. A healthcare system that it has set already at home for him needs to monitor and to check him periodically or non-periodically. At the same time, in an emergency case also, the system has to see his state in real time. Next the hospital has to use his profiling information to know who the patient is.

A profiling that a patient use in a healthcare system is stored in patient's smart device such as smartphones. Table 1 shows the notations used in this study.

The smart device is usually connected to ~~a~~ the hospital server / network [12, 13]. There is *V*, *SN*, *P.ID*, *H.ID*, *D#*, *DI* and *NRD* which are recorded with based on a patient's certificate in a profiling in the smart device and a server in a hospital keeps same information. At the same time, the medical records are going to be stored in ~~a~~ Patient Disease List (PDL) and in Patient Disease Record (PDR) [6, 9]. It usually defines the name of diseases that the patient has in PDL and PDR is for having the explanation of diseases, a patient status / a condition and its prescription. Figure 1

Table 1 Notation

Notation	Content
X.Certificate	X.509v3 Certificate
V	Profiling Structure Version
SN	Serial Number
P.ID	Patient IDentificate
H.ID	Hospital IDentificate
D#	Disease Unique Number
DI	Doctor Information
NRD	Next Reserved Date
DC#	Doctor Call #
PDL	Patient Disease List
PDR	Patient Disease Record
P	Prescribe
Ns	Notices

shows the definition of a patient profiling structure. A patient profiling (PP) is applied by using the extensions field of an X.509.v3, and the description of the field is in Fig. 3. A version and a serial number in a profiling structure is the unique information about a hospital, a patient. ID has the unique identification of a patient, and a hospital.ID is kept a hospital unique identification. Surely, a patient may have more hospitals, so it can be multiple hospital.ID in a patient profiling. Basically, a *disease.#* links a field of the database in a server which stores a medical record of a patient, on the other hand, the PP is not supposed to have any contents of a disease, it has only the *disease.#*. Also, it has an emergency call number of a responsible doctor; it can automatically call to the doctor when the patient has a serious problem. In addition, by adding next reserver date, the system alarms next agenda to the patient such as the next visit.

Figure 2 shows all fields which are going to be used in a Patient Profiling Structure in a Proxy Certificate (PC) Extension. A *Patient Profiling Structure* defines next five fields, *pPatientID*, *pHosptialID*, *pDiseaseN*, *pDoctorInfo*

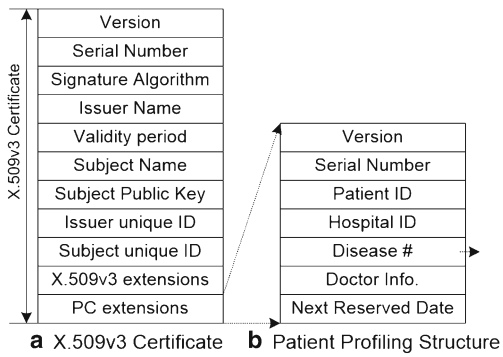

Fig. 1 A patient profiling structure in X.509v3 certificate

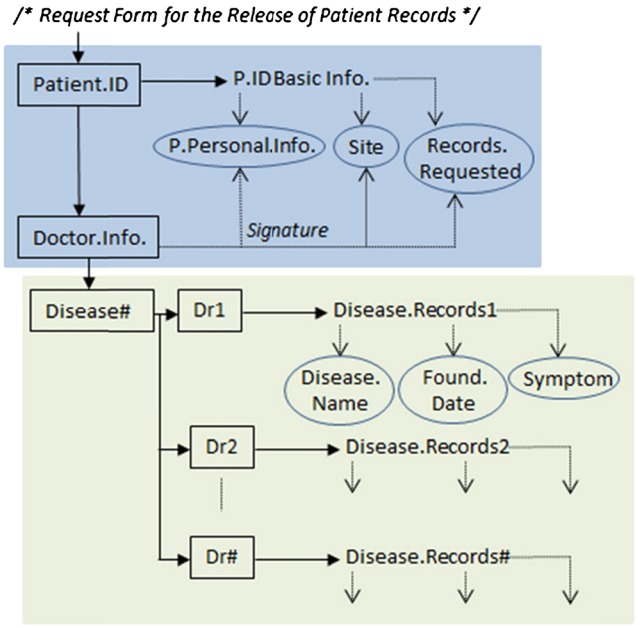

Fig. 3 A server profiling structure

and *pNextRDate*. All of them have an OBJECT/IDENTIFIER to store a patient ID who is going to register but *pDoctorInfo*. *PDiseaeN* which has OBJECT/IDENTIFIER can store all diseases. In case there is no disease, it may define their periodic or nonperiodic diagnosis comments. In 4 items that it mentioned *pPatientID*, *pHosptialID*, *pDiseaseN* and *pNextRDate*, *pPatientID* keeps the patient's identification which is unique. This ID can be the patient's security number or just a id number that the hospital makes for the patient, and *pHosptialID* is a unique ID number in a country also the hospital declares this number after registering with the department. *pDoctorInfo* has a doctor's information (a Private doctor or a doctor who cured the patient), also according to the

```
//Extensions
id-pkix OBJECT IDENTIFIER ::= { iso (1) identified-
organizations (3)
        dod (6) internet (1) security (5) mechanisms pkix
(7) }
id-pe OBJECT IDENTIFIER ::= {id-pkix 1}
id-pe-proxyCertInfo ::= SEQUENCE {
        pCPathLenConstraint INTEGER (0..MAX)
OPTIONAL,
        proxyPolicy ProxyPolicy }
proxyPolicy ::= SEQUENCE {
        policyLanguage OBJECT IDENTIFIER,
        policy OCTET STRING OPTIONAL }

//Patient Profiling Structure
proxyStructure ::= SEQUENCE {
pPatientID OBJECT IDENTIFIER,
pHospitalID OBJECT IDENTIFIER,
pDiseaseN OBJECT INTEGER (0..MAX) OPTIONAL,
pDoctorInfo OBJECT IDENTIFIER,
pNextRDate OBJECT IDENTIFIER }
```

Fig. 2 A patient profiling structure

number of the disease or the number of a doctor, there can be registered more than two doctors in pDiseaseN. If the patient needs a long time to care, it registers next appointment date in *pNextRDate*, also all patients who will plan to meet with a doctor can save the reserved date in this field. How to process the security for the PP such as algorithm information and an authentication method is defined in a security field and we can see a process mechanism in a mechanism field. A '*id-PE-proxyCertInfo*' has an authentication information about a home device which a patient uses, a '*pCPathLenConstraint*' keeps an authentication information of registered devices, and any information about a policy and a device is set in a '*proxyPolicy*'. All fields in the PP that it suggests are defined in a '*proxyStructure*'. There are '*pPatientID*', '*pHospitalID*', '*pDiseaseN*', '*pDoctorInfo*' and '*pNextRDate*' in a '*proxyStructure*'

Figure 3 shows how to process with 5 items which defined in Fig. 2, each fields are going to use in an official document like a prescription from a hospital. First, *pPatientID* which has '*P.ID Basic Info*' and '*Doctor Infor* (*pDoctorInfo*)' has a basic information about a patient, and there are a patient name/sex/age/address and a telephone number in *pPatientID*. The last field of a *pPatientID* will be set with a patient signature.

```
/* Request Form for the Release of Patient Records*/
P.ID Basic Info ::= P.Personal.Information||SITE||Records Requested {
P.Personal.Information OBJECT Char/Integer, //Name, Sex, Birth, Age, Address, Phone number,
SITE OBJECT Char,
Records.Requested OBJECT Checkable,
Doctor Infor ::= Signature,}
```

In a page which keeps the patient detail information such as a disease name, it follows next type, 'Disease # ->#. Disease Records' and there is a disease name, when it finds, symptom and so on in the page. According to the number of a disease and a doctor, the number of a field will be increased.

```
Disease# ::= Dr#||Dr#|,..,||Dr# {
Dr1 ::= Disease.Records {
Disease.Name OBJECT Char,
Found.Data OBJECT Date,
Symptom OBJECT Char,
…,}
Dr2 ::= …{…}
:
Dr# ::= … {…}
}
```

Figure 3 also shows the Server Profiling Structure (SPS) which defines in a server. Especially, before the medical records or the disease records are stored in a database, they have to be encrypted. At this time, also it can set its security grade, and if it makes the grade, it is stored in a 'Security Module (*Sig*. |*Encrypt*)' [12]. According to the security grade, a server can use lots of security algorithms, also the system can decide the security algorithm following a patient, for example, they can apply 'ECC algorithm' for 'Patient A' who has a special medical record or they can use 'SEED algorithm' for 'Patient B' also who has a different medical record. Surely the records have to be encrypted and got their signature by a responsible doctor before storing.

Discussion

Proposed scheme

There are totally 3 steps in client/ server's security procedure, and now it explains how doing the procedure of an authentication between a client and a server in a section [14, 15]. To get a high level security, it applies an One-Time password (OTP) and a Dual Hash Function (DuH) in the suggesting model. The OTP usually uses a password which never used before, it means that the OTP doesn't use the password which had already used even one time, it always updates new password. A DuH that it suggests are used in this model which means this model has two times the hash functions. Each function has a each input value in each step, they are different value. In the first hash function, *P. Counter* and *P. Secret* are will be an input value, and *Ks* and *ID* are the second input value of the second hash function. Why it uses a hash function is that it is impossible to get an input value with the hash value so far. However, some experts mention that because of improving computing technique and reverse algorithm which can trace an encryption algorithm, it can be possible to get the input value from the hash value. That's why; it suggests a DuH to avoid this problem. There are totally 4 steps to process from a patient to a server. In Fig. 4, it explains each step. The first step is key generation step which makes a hash value by using an One-Time Password. The next step is the comparison step that tries to match two hash values; one is generated by a client and another is made by a server. They certainly use a same algorithm to make them. Finally the last step is a security step to encrypt and to decrypt that a client can read after decrypting his/her medical records with K_p, also a server sends their medical records to a client after encrypting with K_s. Also, all records usually are stored after encrypting with K_s. Without K_s, no one can read them.

Next is a hospital model. There is a lot a patient that they have already registered in a hospital, and some of them will be diagnosed remotely or some of them are enjoying their life as usual without this system. If A hospital model defines 'Hospital $(P.ID_1,\ldots,P.ID_n)'V_n(j)=\sum_{i=0}^{I}(P_{ij}\cdot X_{ji}+,\ldots,P_n\cdot X)$ is for a

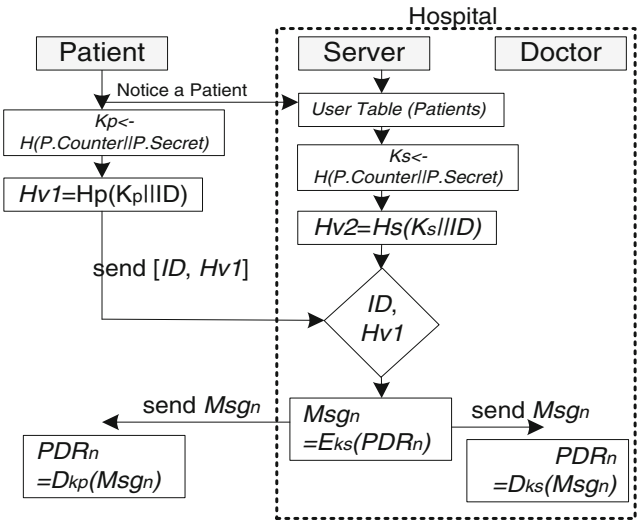

Fig. 4 The process of a patient to a server (a block diagram)

patient who have registered at a hospital, $P_{rn}(j) = \frac{e^{y}}{\sum_{j} e^{y}}$ is for a patient who is in a diagnose.

STEP 1 Key Generation Step, before a client connects to a server, firstly the client has to generate K_p with a patient counter and a patient secret. At that same time, a server also begins to make K_s by using same functions which are the same as a patient's done because the client notices a client's connection to the server. That is, a server also makes the Ks with a patient counter and a patient secret by OTP algorithm. Next it gets Hv_1 by doing a hash function with a patient.ID [16] and K_s, and then it sends the Hv_1 and a patient.ID [17].

Algorithm 1

Input: (*Patient.Counter*(*pc*), *Patient.Secret*(*ps*), *Seed*), *where GENERATE=null*, *when Kp is ready*, *ps->Seed*(*Random value*);

0: *Initialize pc*, *ps*, *Kp*(*kp*)=*0*, *hash values* (*Hv1*);

1: *p.ID=0*, *Kp=0*, *pc=0*, *ps=0*, *Hv1=0*

2: **get** *ps<- random.function* (*ps*), after *ps<-Seed*()

3: **while** *generate*(*pc*||*ps*)

4: **for** *Kp<-0* with *pc=0*

5: **activate** *readystate*(*Kp*)

6: *Kp<- process.done* (*generate*)

7: **endfor**

8: **process** *Hv1<-* (*Kp*||*ID*)

9: **send** (*Hv1*, *ID*) and *ID Database*

10: Message(*WakeUp*) and **ask** *activation* (*Server*)

12: **endwhile**

13: **follow** '*algorithm 3*' to **encrypt**() and to **decrypt**() *a message*

STEP 2 Comparison Step, the server, which is asked to generate a key, makes K_s with the same way that a patient done [18]. The patient has been using this hospital after registering their information in a server to generate an OTP value. Because a patient has already registered two items (a patient counter and a patient secret) in a server, the server can make Hv_1. Also with the same way, the server can generate Hv_2. Although they are birthed in a different area (in a patient and in a server), the values are same since they used a same algorithm. After making them, it begins to confirm a patient with the received *patient.ID* and next it tries to authenticate a patient by matching between Hv_1 and Hv_2. Once, after confirming, it may ask the security process for a patient's data which is going to be transferred.

Algorithm 2

Input: (*Patient.Counter*(*pc*), *Patient.Secret*(*ps*), *Seed*), *where GENERATE=null*, *when Ks is ready*, *ps->Seed*(*Random value*);

0: *Initialize pc*, *ps*, *Ks*(*ks*)=*0*, *hash values* (*Hv2*), *patients*(*ID*) *Database*;

1: *p.ID=0*, *Ks=0*, *pc=0*, *ps=0*, *Hv2=0*,

2: **receive** *ID* and *Message*(*WakeUp*)

3. **find** *p.ID<- Database*(*ID Database*) or **reject** *p.ID*

4: **while** *generate*(*pc*||*ps*)

5: **for** *Ks<-0* with *pc=0*

6: **activate** *readystate*(*Ks*)

7: *Ks<- process.done*(*generate*)

8: **endfor**

9: **process** *Hv2<-* (*Ks*||*ID*)

10: **if** *Hv2 == Hv1*

11: *go* or **reject** *p.ID*

12: **endif**

13: **endwhile**

14: **follow** '*algorithm 3*' to **encrypt**() and to **decrypt**() *a message*

STEP 3 Security Step, when it sends the medical record between a patient and a

server, the data is just going to be transferred to them without a decryption[19]. A patient and a doctor keep each key; K_p is for a patient and K_s is for a doctor, so that they can decrypt to read with each key.

Algorithm 3
Input: (*MSGn*, *Ks*, *Kd*)
1: **process** *Encryption*)*Ks* or (*Decryption*)*Ks*
2: **send** '*MSGn*<− *Eks* (*PDRn*)' **to** *a Patient*
3. **send** '*MSGn*<− *Eks* (*PDRn*)' ***to*** *a Doctor*
4: ***decryption*** *PDRn*<− *Dkp*(*MSGn*)
5: ***decryption*** *PDRn*<− *Dks*(*MSGn*)

Security discussion

In this section, it defines what security problems are and how it tries to protect and to proof.

Problem 1. Security Key outflow problem If the K_p is leaked by attacking or because of carelessness treatment, an attacker can read the medical reports by decrypting them with K_p, it surely becomes a security problem.

Proof 1. A patient and a server are keeping and sharing a patient counter and a patient secret of two items, in a patient and in a server, to generate an OTP. With two codes, they generate with a key at that same time, however, they are not supposed to be sent, instead, they are only used to encrypt and to decrypt the medical records, which is the 1st hash function used. The first value will be kept in each part (Keep.SECRET (K_p). Next it goes to do the 2nd hash function with the first value and a patient.ID, and get the 2nd hash value (Hv1). A patient sends [(Hv1), a patient.ID] to a server. Because this system doesn't send K_p to a server, an attacker cannot get K_p, therefore an attacker can't decrypt the medical records.

$S \in \{(P.ID_1, P.ID_2,\ldots,P.ID_n)\ (Patient_1,\ldots n.Counter) \cap (Patient_1,\ldots n.Secret)\}$;
READABLE $(P.ID_1, P.ID_2,\ldots,P.ID_n)$;
$Keep.SECRET\ \{(Patient_1.Counter) \cap (Patient_1.Secret)\}$;
$P.ID_1 \in \{[(H.ID_1,..,H.ID_n),\ldots, P.ID_n] \in (H.ID_1,.., H.ID_n)\}$;
$K_p <- Hash\ (Patient.Counter||Patient.Counter)$;
Keep.SECRET (K_p);
Hv_1<−*Hash* $(K_p \| ID)$;
Send (*ID*, *Hv1*) *to Server*;
Attacker∉ K_{p1};

Problem 2. Abuse patient ID In a case, an attacker tries to login with a patient.ID that the attacker has already kept it by attacking.

Proof 2. To confirm the login process, a patient sends a patient.ID and the second hash value, which generated with a patient.ID and K_p that is from the first hash function with a patient counter and a patient secret, to a server. At that same time, a server runs to get the value with the same way, also whenever the value changes, it has to be sent to a server. It means that the value is going to be renewed whenever a patient connects to a server or a session closes. Therefore, because only a patient and a server know and keep this value, an attacker cannot generate this value, so the attacker can't decrypt the medical records, also, this system can detect the forgery patients or an attacker since they can't have this key to confirm.

$Hv_1 \in \{(Patient.Counter, Patient.Secret, P.ID, K_p)$;
K_p<− (*Patient.Counter*||*Patient.Secret*);
GENERATE Hv_1<− $H_p(K_p \| ID)$;
READABLE $(P.ID_1, P.ID_2,\ldots, P.ID_n)$;
SEND (*ID*, Hv_1);
KEEP $(K_{p1}, K_{p2},\ldots, K_{pn})$ *in Server*/*in Clients*;
Attacker ∉ (*Patient.Counter*, *Patient.Secret*);

Problem 3. Forward secrecy problem In the existing ways, once a patient registers an *ID* and a *password*, the *password* would not be changed or updated before the patient does it. Therefore, in case an attacker takes their *ID* and *password*, they can login easily and let get all patients mixed up. That is, the password should not be used after logout.

Proof 3. The one of the advantages of an OTS has been always to generate a new key in real-time. It means that the key, which have already used, isn't going to use again and also is never generated the same key. In this study, we have designed a dual hash function to get the second hash value with the first hash value which is from an OTP, so it is impossible to use again the key in the next time; therefore, it is safe against a forward secrecy.

KEEP (*Patient.Counter*, *Patient.Secret*);
K_p<− (*Patient.Counter*||*Patient.Secret*);
UPDATE NEW (*Patient.Counter*||*Patient.Secret*) <− (*Patient.Counter*||*Patient.Secret*);
Although Attackers (*Patient.Counter*||*Patient.Secret*);
Attacker ∉ {*UPDATE NEW* (*Patient Information*)};

Problem 4. Password guessing attacks In the existing authentication protocols, usually they use a password to login, however the way to use the password is still belonging to a low-level security grade, that is, it surely is a weak secret because an attacker can attack with the way how to guess a password. Also, we have to study new authentication protocol to protect it and it certainly will be causing a big overhead.

Proof 4. The suggested method uses a hash value which is generated by a hash function with two items, a key which is from an OTP and a patient ID. The value always changes whenever a patient tries to login. Therefore, it is impossible that the attacker guesses the password. In addition, it is not to add a new key, but to update the value after running the hash function, so there is no any overhead.

$KEEP(Patient.Counter, Patient.Secret)$;
$UPDATE\ NEW\ (Patient.Counter||Patient.Secret) <- (Patient.Counter||Patient.Secret)$;
$KEEP.SECRET(K_{p1})$;
$GENERATE\ Hv1 <- (K_{p1}||ID)$;
$IMPOSSIBLE\ GETTING\ K_{p1} <- Hv_1$;
$Attacker \notin GUESS(K_{p1})\}$;

Problem 5. Spoofing attack The following messages between a patient and a server can be forwarded by an attacker with a simple skill to an attacker's computer or to another computer that the attacker defined, and then the attacker can receive them and can read them. Then the patient is going to be dangerous.

Proof 5. To pass the login, first it checks a patient's ID and then the next step is to compare the Hv_1 which is from the second hash function with K_p which it is from the first hash function with a patient.counter and a patient.secret. Although the attacker can pass the server authentication step with an ID and Hv_1 that are a patient's, the attacker can't decrypt the encrypted medical records because the attacker doesn't have Kp or K_s. Only a patient and a server keep each key, K_p in a patient and K_s in a server. Two keys are generated through the first hash function with (a patient ID, a patient secret) and (a server ID, a server secret), and they never send them, instead of sending they use when it encrypts and decrypts. Because they don't send them, the attacker cannot get the key which generates it in the first hash function. So the attacker can't guess the K_p or K_s also. Therefore, if they receive the encrypted medical records illegally, the attacker can't read them.

$Attackers \in \{(ID, Hv_1) \cap (MSG_n)\}$;
$Attackers \notin \{(K_{p1}) \cap KEEP(K_s)\}$;
$IMPOSSIBLE\ Decrypt(MSG_n)\ without(K_{p1}\ or\ K_{s1})$;

Cost comparison

In this section, it evaluates the efficiency of the study of Khan et al. [18], Chen et al. [16], Q. Jiang [14] and our proposed scheme (Table 2). There are three notations, *T*, *C* and *n*, that is, *T* is the time complexity of the hash computation, *C* is the time complexity of the symmetric encryption/decryption, and *n* is the number of registered users [14]. The existing methods [14, 16, 18] uses the hash function to register a patient in a hospital, they usually spend as much as *T* for a patient and *2 T* and *T+3* for a server (a hospital). Why the proposed scheme has less than them is because this scheme uses OTP, therefore the new scheme doesn't use a hash function. In a login, because the new scheme uses a dual hash function, it uses *2 T*. On the other hand, the existing schemes spend a hash function as much as *3 T+C*, *4 T* and *5 T* in a user and *4 T*, *(n+4)T* and *3 T+3C* in a server. When they store the medical records, they usually do the encryption process. In case the existing scheme, they need the *TCn* cost in a user that it needs only one time to read, however, the server needs the cost as much as twice of a user's needed because a server usually has to do an encryption to read and a decryption to store. On the other hand, the new scheme needs only *Cn* for a patient and *2Cn* for a server. In the new scheme, they don't need to use *T* because two parts usually get the key to encrypt and to decrypt when they login. As it analysed in Table 2, the proposed scheme needs less cost than the existing methods.

Table 2 Computation Cost // T: the time complexity of the hash computation, C: the time complexity of the symmetric encryption/decryption, n: the number of registered users

Cost		Khan et. al [18]	Chen et. al [16]	Q.Jiang [14]	Proposed Scheme
Reg.	User	T	T	T	–
	Server	2 T	2 T	T+C	–
Login	User	4 T	5 T	3 T+C	2 T
	Server	4 T	(n+4)T	3 T+3C	2 T
Encrypt./Decrypt.	User	TCn	TCn	TCn	Cn
	Server	2TCn	2TCn	2TCn	2Cn

Conclusions

It has studied about how to keep a patient's security key to authenticate between a patient and a server in a hospital and about how to safely communicate the data like disease information or personal information from a server to a patient in this paper. It also has studied the old who lives alone at home, who is an inconvenient with a disease or who is a disability, and a server in a medical centre which keeps all medical records without a security process such as an encryption algorithm. If a patient feels that something is wrong with the body or if they want to know how its state goes right now, surely they will think of using the healthcare system by using their smart devices to connect to the server. As soon as the server detects the asking from a patient, it starts to decide what they want after confirming if the patient is correct. Although the patient gets them via the smart device, there may be a potential security problem in this process such as their data drain while they communicate between two parts, a hospital server and a patient. Surely an attacker will try to use to login with the patient's ID which had already stolen them, therefore the attackers can login as if the attacker is a real patient because now there needs only an ID and a password to login in many existing systems. Apart from it, there is also a lot of skills to give a patient bothering in a home healthcare system. This study showed some assumes about how to attack and it has defined the attack scenarios to find the better solution to protect. As the result, it suggested a secure user profiling structure which defines all attributes like a patient name/id/password/secure key / all disease information, and a procedure how to go which uses a one-time password with a hash function to make an encryption / a decryption key, and do the two times hash function to authenticate, then next it sends the value with a patient. ID to a server. As it has already been mentioned, this study has studied the solution to protect a cyber-attack, also, in the worst case, the attacker has already taken a patient's ID and it could make patient's key passing through a dual hash function, the attacker can't read the patient's medical records because the attacker can't know the key which would be got with the first hash function, to decrypt. Finally, with research results, safely we can use the healthcare system in between a patient and a server, however, it can't cover all securities in other areas such as between a smart device and a sensor which will be hung on a patient's body to detect a health-trouble in real-time, that it, a monitoring function in real-time may be used in unsecure state. Therefore, it plans to study about how to protect the security of a smart device and a sensor in a future. In addition, the structure that it suggests in this paper, usually defines and updates the patient' condition and state in detail. Also, because it completely keeps the history of their condition, we can realise all courses of treatment after detecting the disease. This information will be shared with all hospitals. In future, it is going to lead in an international project which studies a full home-healthcare system and develop by extending this structure, also it will be based on Internet of Thing (IoT) and biotechnology (BT) that will be expected the integrated output with both IoT and BT for all patients. Finally, it can help them in their safe life.

References

1. Guo, R., Wen, Q., Jin, Z., and Zhang, H., An efficient and secure certificateless authentication protocol for healthcare system on wireless medical sensor networks. *Int. J. Distrib. Sens. Netw.* 2013(4):1–7, 2013.
2. Fragopoulos, A. Gialelis, J. and Serpanos, D. Security framework for pervasive healthcare architectures utilizing MPEG-21 IPMP components, *International Journal of Telemedicine and Applications*. 2009:1–9, 2009.
3. Kim, H., Ryu, E.-K., and Lee, S.-W., Security considerations on cognitive radio based on body area networks for u-healthcare. *J. Secur. Eng.* 10(1):9–20, 2013.
4. Zhang, G.H., Chung, C., Poon, Y., and Zhang, Y.T. A review on body area networks security for healthcare, Int. J. Distrib. Sensors Networks, 2011.
5. Haque, S.A., Aziz, S.M., and Rahman, M. Review of Cyber-Physical System in Healthcare, Int. J. Distrib. Sensors Networks, 2014.
6. EMC, Cybercrime and the Healthcare industry, http://www.emc.com/collateral/white-papers/h12105-cybercrime-healthcare-industry-rsa-wp.pdf
7. Forbes, Hackers Stole Data On 80 Million Anthem Customers. Why Wasn't It Encrypted?, http://www.forbes.com/sites/brucejapsen/2015/02/06/anthem-didnt-encrypt-personal-data-and-privacy-laws-dont-require-it/
8. Yu, H., He, J., Liu, R., and Ji, D., On the security of data collection and transmission from wireless sensor networks in the context of internet of things. *Int. J. Distrib. Sens. Netw.* 2013:1–13, 2013.
9. TrustedSEC, Explaining security issues with healthcare.gov, https://www.trustedsec.com/january-2014/explaining-security-issues-healthcare-gov/
10. Ogiela, M. R., and Ogiela, U., Linguistic protocols for secure information management and sharing. *Comput. Math. Appl.* 63(2):564–572, 2012.
11. Ogiela, M.R., Ogiela, U. Linguistic extension for secret sharing (m, n)-threshold Schemes, *2008 International Conference on Security Technology*, December 13–15, 2008, Hainan Island, Sanya, China, pp. 125–128, 2008.
12. Ko, H., Chen, N., Marreiros, G., and Ramos, C. Safe high accuracy Context-Aware Matrix (CAM) making based on X.509 proxy certificate, *Lecture Notes in Computer Science*, pp. 829~837, Korea University, Seoul, S. Korea, June 25~27, 2009.
13. Zhu, Z. A., An efficient authentication scheme for telecare medicine information systems. *J. Med. Syst.* 36(6):3833–3838, 2012.
14. Jiang, Q., Ma, J., Ma, Z., and Li, G., A privacy enhanced authentication scheme for telecare medical information systems. *J. Med. Syst.* 37(1):1–8, 2013.
15. Cao, T., Zhai, J. Impr*oved dynamic id-based authentication scheme for telecare medical information systems,* J. Med. Syst. Vol. 37, No. 2, 2013.
16. Chen, H.-M., Lo, J.-W., and Yeh, C.-K., An efficient and secure dynamic ID-based authentication scheme for telecare medical information systems. *J. Med. Syst.* 36(6):3907–3915, 2012.

17. Zhang, L., Zhang, F., Wu, Q., and Domingo-Ferrer, J., Simulatable certificateless two-party authenticated key agreement protocol. *Inf. Sci.* 180(6):1020–1030, 2010.
18. Khana, M. K., Kimb, S.-K., and Alghathbar, K., Cryptanalysis and security enhancement of a more efficient & secure dynamic id-based remote user authentication scheme. *Comput. Commun.* 34(3):305–309, 2010.
19. Xiong, H., Chen, Z., and Li, F. G., Provably secure and efficient certificateless authenticated tripartite key agreement protocol. *Math. Comput. Model.* 55(3):1213–1221, 2012.