Architecture for Portable and Secure Patient Smart Card

Amjad Gamlo, Omar Batarfi

Computer Sciences Department, King Abdul-Aziz University
Jeddah, Saudi Arabia

***Abstract*— In this paper, the patient smart card, its uses and its importance have been defined. Patient smart card projects have been reviewed in order to evaluate their provided security levels. Then, a patient smart card system has been proposed to overcome the disadvantages of reviewed projects. Architecture and algorithm which provide higher security for the card have been suggested and tested. Testing methods have been described. Finally, results proved that the security services- confidence, authentication, integrity and non-repudiation- are realized in the proposed system.**

Keywords*— Patient Smart Card, Security Services, Confidentially, Authentication, Integrity, Non-Repudiation, Digital Signature, Private Key, Public Key, Encryption, and Decryption.

I. INTRODUCTION

Establishing accurate identification of each person receiving healthcare services is at the heart of improving healthcare information management. By validating identity and linking this with a patient's medical information; accuracy of medical care increases and the potential of medical fraud is significantly reduced. And, if the patient can carry his medical record with him/her anywhere and anytime and present it to any doctor at the time of consultation i.e. the patient have a portable medical identity; this feature can improve the healthcare systems.

The appropriate media for previous purposes should be cheap, easy to use, carry and update with new information and should not get damaged easily. "Smart card" appears as the most suitable medium to be used in healthcare information systems. Smart cards can be described as credit card look-liked, small microprocessor embedded in to store and process data. And it has a widespread usage especially in telecommunication systems.

Smart card technology should be a secure media to be used in healthcare information management. Using the public key infrastructure (PKI) with the smart card technology is providing high levels of confidence and portability. Public key infrastructure is a cryptography method that achieves privacy through using public/private key pairs to facilitate third party examination of and vouching for user identities. The PKI provides four security services which are confidentiality, integrity, authentication and non-Repudiation.

In the next sections, Background about the PKI and the patient smart card is first presented. Then, related projects are reviewed. Then, the proposed architecture and the testing methods are described. Finally, Results are discussed to conclude that the security services- confidence, authentication, integrity and non-repudiation- are realized in the proposed system.

II. BACKGROUND

A. Public Key Infrastructure (PKI)

PKI is an arrangement in cryptography that provides four basic services: confidentiality, integrity, authentication and non-Repudiation.

An increasing number of applications are using public key infrastructure (PKI) to provide high levels of confidence when exchanging information, especially over the internet [4].

B. Patient Smart Card

Patient smart card is being introduced in many countries. It contains a microprocessor, memory and an interface to the outside world.

There are three distinct approaches to store and access patient health history on smart cards. All information about patient may be stored on the card and no information stored in a central database, or the card can include a key to access the information that is on a central database, each of previous two methods have disadvantages. In the first method, there is no available data for research purposes. The second approach disallows off-line services. But the approach that avoids these disadvantages is that the patient's medical history stored on a central database, and some important information stored on the card with a key to access the DB patient's record.

However, Information on the card classified in four types: secret and cannot be accessed such as keys and encryption algorithms, read only and should be password protected such as arithmetic transactions, read/write and should be password protected such as last medical diagnosis and prescriptions, and public and it should be readable without a password and it is used in emergency situations such as names and addresses and allergy information [1].

There are several types of smart cards, such as:

- A digital signature smart card which performs the basic smart card functions plus hashing, digital signature generation, and digital signature verification. It provides more protection for the data it carries.
- A cryptographically protected smart card which can encrypts its data for storage, and decrypts them for use. It may encrypt or decrypt data being exchanged with an application program to protect it while in

transit between the smart card and some trusted workstation.

These cards will only be enabled to do their previous functions after the cardholder has entered the Personal Identification Number (PIN) for more security and information privacy [3]. Smart cards that use digital certificates offer greater security and portability for Internet-based business than other security solutions. Placing the digital certificate and key pair on the smart card provides more protection against theft, and requiring a PIN to access the user's credentials on the smart card provides an added layer of protection if the smart card itself is lost or stolen [5].

III. LITERATURE REVIEW

There are many projects implemented and applied in the patient smart cards area. The Health Insurance Institute of Slovenia (HIIS) started a project of nation-wide introduction of smart cards in 1995, called the Health Insurance Card (HIC) project.

They merge the internet technology with the smart card technology while smart cards link via internet to a central server where the database (DB) is run. Smart cards used in this project are using symmetric key cryptography, not intended for digital signature operations, and not PIN protected. They proposed the use of public key technologies and authentication methods for future work [2].

A system called smart card healthcare system (SCHS) is developed in Turkey in 2005. Both patients and doctors have smart cards in SCHS. Doctors use their cards to be authenticated in the system. Patient cards, which include owner's general health information encrypted with owner's DES (digital encryption standard) symmetric key, can be accessed without any database connection.

When a doctor's smart card or a patient's smart card is inserted in card acceptance devices (CAD), authentication is assured between card and host computer software by key exchange. Then, card access PIN is requested from card owner and the entered PIN is checked by the smart card itself. Then, a smart card session is started.

After examination, the doctor updates inspection and prescription information on the patient smart card with new data. Updated data are again encrypted using patient encryption key, signed with doctor's private digital signature and sent to the server. On the server side: signature is verified with doctor's public key, then the patient data is decrypted again using the patient DES key. The required DES and DSA (digital signature algorithm) keys are obtained from the hospital central database [6].

The first project is portable over the nation, but offers low security levels. The second project is more secure through the application of digital signatures generation and verification, but it is not portable over many hospitals.

In the next sections, patient smart card architecture, which preserving the card security while maintaining portability, will be proposed and described.

IV. PROPOSED ARCHITECTURE

The architecture shown in Figure 1 was proposed to establish a patient smart card system with high level of security and wide-nation portability.

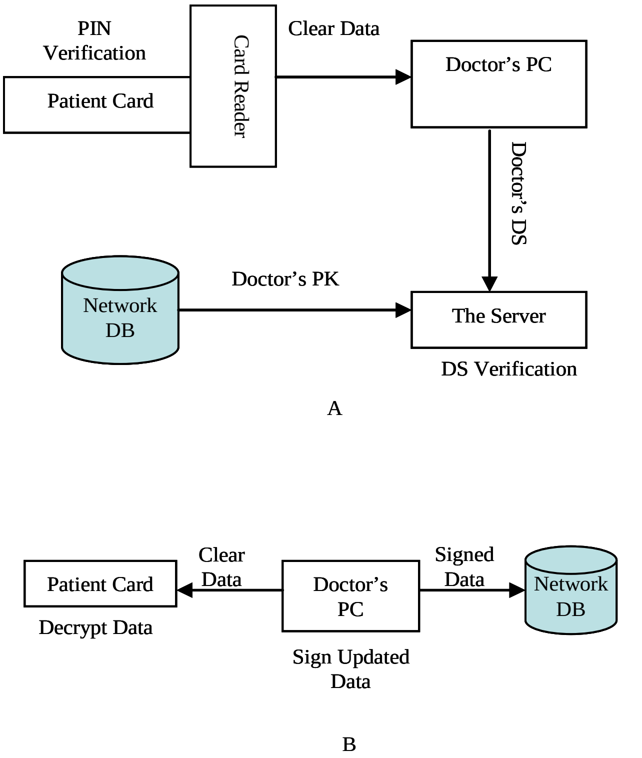


Fig. 1 The patient smart card proposed architecture

A. The Architecture Entities

The architecture entities are described below:

- Network DB which stores the doctor's public keys:
  - The DB located in the internet, so it can be accessed from any hospital.
  - Public keys of all the doctors in the country are registered in this DB.
- The patient smart card which stores the patient's private key and public key and the database network address, it also have the ability to decrypt and encrypt its data.
- The card reader which allows the connection between the doctor's PC and the patient card after the PIN is verified.
- The doctor's PC which stores the doctor's private key and creates his digital signature.
- Server which in verification process is done.

B. The Algorithm Steps

As shown in Figure 1, the system will proceed as the following:

```
Step1: The Card Entered into the Reader
Input ← The Patient's PIN
Process:
    The Card Checks the PIN
        If verified
            Decrypt Data using the private key
            Send the information
        Else
            Send Error message
Output → Card Clear Information

Step2: The Doctor's PC which is connected to the Reader
Input ← Card Clear Information
Process:
    Display the information
    Retrieve the DB address from the card
    The Doctor Send his Digital Signature (DS) to the server.
    Output → The Doctor Digital signature.

Step3: The DB Server which is connected to the PC
Input ← The Doctor Digital signature
    Process:
        The Server Retrieves the Doctors Public Key (PK) from the DB,
        Then verify the Digital Signature
            If verified
                Accepts the digitally signed new Information from the Doctor,
                Then Store it on the DB.
            Else
            Send Error message
Output → Authentication and Integrity are done.
```

Figure 1-A shows how that the data is transferred from the card to the PC. And then how the doctor verification is done.

Step 4: The PC sends the Information to the card.
Step 5: The card encrypt the data with the patient public key.

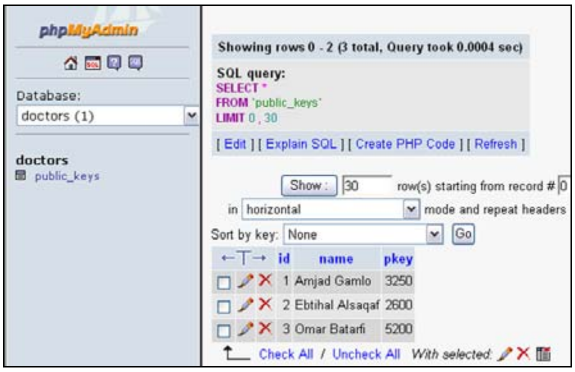
Figure 1 B shows how the doctor sends the data in two directions: first, in clear form to the card and then the card encrypt it; second, digitally signed to the database.

V. Test Method

Step1 is providing two security layers. First, the data encrypted using the patient private key, and then the information would not be decrypted without the right patient PIN. C++ Program was written to show how this mechanism works. In the program, the private and public keys are supposed to be small numbers, and their relationship is based on the ASCII code; this relation is just supposed to facilitate and speedup the testing process.

To test the doctor authentication process in *Step3*, PHP program was written on apache local server with MySQL database. The database that is shown in Figure 2, stores the doctor's public keys. The program creates the doctor signature with his private key, and then retrieves the public key from the DB to verify the signature. If the private key was wrong, the verification will fail, and if the doctor information is not stored in the DB, the authentication will also fail. Figure 3 shows an example of a successful verification.

The testing source codes are in the appendix.


phpMyAdmin
Database:
doctors (1)
doctors
public_keys
[Edit] [Explain SQL] [Create PHP Code] [Refresh]
Show :
Go
Check All / Uncheck All With selected:

Showing rows 0 - 2 (3 total, Query took 0.0004 sec)

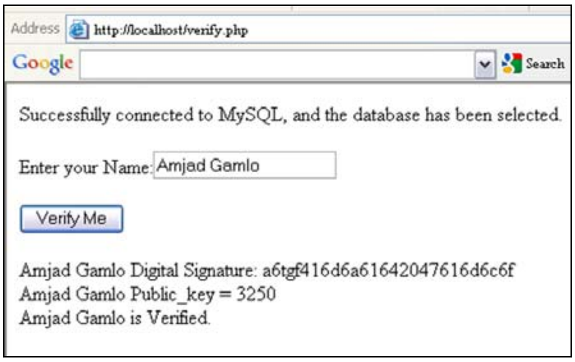
SQL query:

```
SELECT *
FROM `public_keys`
LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers
Sort by key: None

	id	name	pkey
□	1	Amjad Gamlo	3250
□	2	Ebtihal Alsaqaf	2600
□	3	Omar Batarfi	5200

Fig. 2 The testing database shot screen



Address: http://localhost/verify.php

Successfully connected to MySQL, and the database has been selected.

Enter your Name: Amjad Gamlo

Amjad Gamlo Digital Signature: a6tgf416d6a61642047616d6c6f
Amjad Gamlo Public_key = 3250
Amjad Gamlo is Verified.

Fig. 3 The testing of doctor verification shot screen

VI. Results and Discussion

The system's scenario is analysed bellow:

- After the patient enters his PIN in the reader, the card is allowed to decrypt the data using the patient's private key.
    - This step enhances the data confidentiality and authenticity by using asymmetric keys and PIN as another security level.
- Doctor's PC displays the clear patient data and retrieves the DB network address from the card, then the doctor's PC sends the doctor's DS to the server.
    - This step authenticates the doctor.
- Server retrieves the doctor's public key from the DB and then verifies the signature. If the Doctor has been authenticated, he is allowed to send the updated data to the server encrypted with his digital signature, and send the information decrypted to the card.
    - This step realizes the integrity.
- Server stores the data encrypted by the doctor's DS on the DB.
    - This step prevents repudiation problems.
- The card accepts the new information, decrypts them with the patient's public key then store them.
    - This step affords the portability.

Note that the data encrypted with the doctor's digital signature will not be stored on the card to allow portability; this was a lack in some previous projects. It is used to prevent the doctor's repudiation. So, if a doctor writes a wrong medical history on the card, the doctor cannot repudiate his updating because any updated information is also stored in the network DB with the doctor's digital signature.

The proposed system scenario is realizing the security services of PKI without ignoring the card portability feature.

VII. Conclusions

Previous sections discussed the ability of the proposed architecture to provide the patient smart card with the PKI security services and to allow the patients' mobility at the same time.

The architecture uses the patient's key pair to provide patient's confidentiality and the PIN to get patient's authenticity. The doctor's digital signature is used to provide doctor's authentication and non-repudiation. Doctor Authentication provides the integrity to the patient's data.
Storing the digitally signed data out of the card and also storing the doctor's public keys in the internet allows the patient to present and update his medical record anywhere and at any time.

References

[1] E. Smith and J.H.P. Eloff, "Security in health-care information systems," *International Journal of Medical Informatics*, vol. 54, pp. 39–54, 1999.

[2] D. Trcek, R. Novak, G. Kandus, and M. Suselj, "Slovene smart card and IP based health-care information system infrastructure," *International Journal of Medical Informatics*, vol. 61, pp. 33–43, 2001.

[3] S. Bing, L. Liang, F. Xunli, "Security technology of smart cards applied in an information system," *Journal of Materials Processing Technology*, Vol. 139, pp. 243–246, 2003.

[4] D. Storch, "Smart cards and public key infrastructure," *Card Technology Today*, pp. 10-12, 2003.

[5] *Extending Managed PKI Services to Smart Cards*, A VeriSign Guide, 2004.

[6] G. Kardas and E. T. Tunali, "Design and implementation of a smart card based healthcare information system," *Computer Methods and Programs in Biomedicine*, vol. 81, pp. 66–78, 2006.

Appendix

The PHP code for doctor verification (*Verify.php*):

```
<?php

function d_sign($name ,
$private_key)//In the PC.
{
   //The signature is the concatination
of the private key with the name ascii.
  $sign = $private_key.bin2hex($name);

  return($sign);
}
function verify($sign ,
$public_key)//In the Server.
{
    //Find the length of the private
key.
    $public_key = $public_key / 650;
    //Remove the private key.
    $sign = substr($sign,$public_key);
    //Reconvert the ascii to its
letters.
    if($sign)
    $name = pack('H*', $sign);
    else
    $name = "";

    return $name;
}

//Reporting errors.
ini_set ('display_errors', 1);
error_reporting (E_ALL & ~E_NOTICE);

// Attempt to connect to MySQL and
print out messages.
if ($dbc = @mysql_connect ('localhost',
'root'))
{
print '<p>Successfully connected to
MySQL, ';

if (@mysql_select_db ('doctors')) {
print 'and the database has been
selected.</p>';
}
else {
die ('<p>Could not select the database
because: <b>' . mysql_error() .
'</b></p>');
}
}
else {
die ('<p>Could not connect to MySQL
because: <b>' . mysql_error() .
'</b></p>');
}

print '<form method="post"
action="verify.php">';
print 'Enter your Name:<input
type="text" name="name"/><br/><br
/><input type="submit" value="Verify
Me" name="submit"/>';

//Ask the doctor about his name.
$name = $_POST['name'];

//Retrieve his private key (suppose
they are retreived from the PC).
if($name == "Amjad Gamlo")
$private_key = "a6tgf";
else if($name == "Ebtihal Alsaqaf")
$private_key = "a9s3";
else if($name == "Omar Batarfi")
$private_key = "a66f9ngf";
else
$private_key = "";

if (isset ($_POST['submit'])) { //
Handle the form.
//Create the Doctor Digital Signature
(assumed to be in the doctor PC).
$sign = d_sign($name,$private_key);
print "<br /><br />$name Digital
Signature: $sign";
//Retrive the public key from the DB
(the server).
$r = mysql_query ("SELECT * FROM
public_keys WHERE name = '". $name
."'");

// Run the query.
if($row = mysql_fetch_array ($r)){
$public_key = $row['pkey'];

print "<br />$name Public_key =
$public_key";

//Verify the doctor by verifying his
digital signature.
$ver = verify($sign ,$public_key);

if($ver == $name)
print "<br />$name is Verified.";
else
print "<br />$name is NOT Verified.";
}
else
print "<br />$name is not Registerd
Doctor.";
}
//Close the connection.
mysql_close();
?>
```

The C++ code for PIN verification (*Encrypt and Decrypt.cpp*):

```
#include <iostream.h>
#include <conio.h>

// prototypes of functions used in the
code.
void encrypt( char [ ] , char []);void
decrypt( char [ ] , int );

//The main Function.
int main( )
{
clrscr();


}

//encrypt data using the public key.
void encrypt (char e[] , char pub[])
{
// convert the public chars to intgers.
int pub_1 = (int)pub[0];
int pub_2 = (int)pub[1];
// find sum of the two converted chars.
int pub_3 = pub_1 + pub_2;
// add the result to each letter in the
string.
```

```
int PIN = 1234 , PIN_2; //Patient PIN
stored on the card.

int private_key = 103 ; // Patient
private key stored on the card.
char public_key[] = "34" ; // Patient
public key, suppose that this value is
stored in the DB.

// The following string is encrypted
using the public key and stored on the
card.
char string[ ] = "Patient Amjad had a
flu.";
encrypt( string , public_key );

cout << "Data on the card is encrypted
by the patient public key:\n\n";
cout << string << endl ;

//Step 1:
cout << "\nOn the Reader: Enter the
Patient PIN:" ;
cin >> PIN_2;

if(PIN_2 == PIN){
// call to the function decrypt( )
using the private key.
decrypt( string , private_key );
cout << "\n\nData is decrypted.\n\n"; }

else {
cout << "\n\nPIN not valid" ;
//End the program.
getch();
return 0;}

//Decrypted data will display on the
doctor PC.
cout << "---------PC-SCREEN----------
\n\n" ;
cout << "Decrypted Patient Data
is:\n\n" << string << endl;
cout << "\n---------------------------
-" ;

getch();
return 0;
for( int i=0; e[i] != '\0'; ++i )
e[i] += pub_3;
}

//decrypt data using the private key.
void decrypt( char e[] , int pri) {
// sutract the private key from each
letter in the string.
for( int i=0 ; e[i] != '\0'; ++i )
e[i] -= pri ;
}
```