RFID Enabled Secure Certificate

Sonali Singh

Department of Computer Science& Engineering, School of Engineering &Technology, Sharda University, Greater Noida, India

Abstract: ***Fake degrees are a menace to the society and threat to integrity of both the certificate holder and educational institution that has awarded the certificate. For this it has been tried to implement a RFID based security system for the certificates that ensures security of information and authenticity of the certificate issued. In this we are trying to encrypt the data at time of transmission using PC-MAC-AES Algorithm for security and used MHRD a 3rd party verifier that collaboration with both university and the organization that need to verify the degree. For this a RFID tag is fabricated with the degree and this will be read by using RFID reader.***

Keywords: RFID, TAG, READER, PC-MAC-AES, MHRD, Authentication, Verification.

1. Introduction

The key concern of using RFID Enabled Secure Certificates is as following
i. Prevent from forgery.
ii. 3rd party verification of degrees.
iii. Global Access of Degrees.
iv. Since there is no such mechanism for verification of degree at an end viz. like if anyone presents his/her degree for job then the organization does not have any method that can automatically verify degree is legitimate or not, instead to wait for manual verification or just accepting that fake degree. In contrast of this using RFID enabled degrees, the organization can verify the degree at the time of presentation to them. In this a RFID Tag is sandwiched between degree and a piece of transparent sheet and been read by RFID Reader.

1.1 RFID System comprises of [1]

i. RFID Tag
ii. RFID Reader
iii. Database

Figure 1 shows the required components which help in communication through RFID [2].

Type of RFID Tag [3]:

a. Passive Tags
i. Do not require power – Draws from Interrogator Field
ii. Lower storage capacities (few bits to 1 KB)
iii. Shorter read ranges (4 inches to 15 feet)
iv. Usually Write-Once-Read-Many/Read-Only tags
v. Cost around 25 cents to few dollars

b. Active Tags
i. Battery powered
ii. Higher storage capacities (512 KB)
iii. Longer read range (300 feet)
iv. Typically can be re-written by RF Interrogators
v. Cost around 50 to 250 dollars.

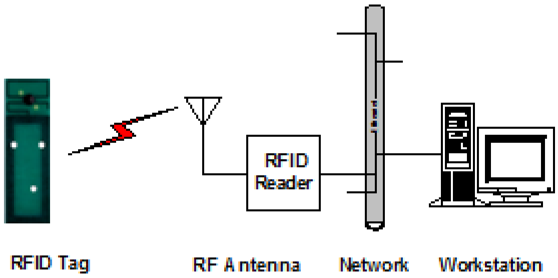


Figure 1: RFID Network Setup

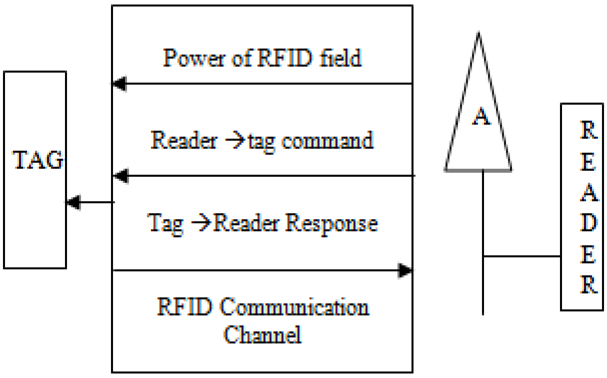


Figure 2: RFID communication

The communication takes place between RFID Tag and Reader through required commands which are sent wirelessly through which communication is established over insecure channel as shown in figure 2 [4].

2. Existing Work

TCS has work in this field and they introduced SMART DEGREE [5]. But it has two main issues that is needed to resolve:
1) To prevent unauthorized person from reading and writing data stored on or transmitted from tags.
2) Encryption must be ensured at all interfaces-medium itself tag-reader & reader-host communication.

For this PC-MAC-AES is used in our proposed work to make the data transmission wirelessly secure. As we have critically analyzed all possible algorithms whether it's symmetric or asymmetric algorithms. As symmetric key algorithms offers speed as in DES, 3DES and AES due to shared key effect [6]. Now, comes to asymmetric algorithm like RSA [7], offers security due to public key cryptography.

3. Proposed Work

There is three main motivations for RFID Enabled Secure Certificates as described in introduction, for this there is three main steps that will be used to make such system

1) Registration
2) Embodiment
3) Verification

These steps are described as below

3.1 Registration

It is primary step, done at the university affiliated colleges at the time of admission of student. At this point of time we can store all required credentials such as student's signature, finger print and the enrollment details using camera, signature pad,and scanner.

3.2 Embodiment

It is final step before delivering the degree to student by the university. At this stage RFID tag is fabricated with the degree. Steps including in this stage is

1) Certificate Initialization
2) Digital Signing
3) Certificate Personalization
4) Certificate Verification

3.3Verification

For this a 3rd party verifier like MHRD (assumed) is included. It will have the collaboration with university server from where it will match the data read by the READER and the data stored at the server. But it has no authority to read the data it will just do 1:1 mapping for verification and validation.

3.4 Security Mechanism

We are going to make a system solution to make degrees verifiable at a global scale and also make the transmission secure from any 3rd party intrusion. That's why we are required to use both private key symmetric encryption algorithm and public key asymmetric, because we require both speed as well as security. For provable security we are using MAC function, to ensure data integrity, authenticity and prevention from any attack and AES algorithm for speed. Hence we are introducing algorithm PC_MAC_AES.

3.4.1 PC_MAC_AES [8]

It is a mode of operation in which we are combining AES algorithm and deterministic MAC function. It uses 4-round version of AES as shown in figure 3. It is 1.4 to 2.5 times faster than usual MAC mode of operation. PC-MAC-AES uses a sub key scheduling for 4-round AES. That means if our algorithm AES is secure then PC-MAC-AES is also secure. In our project we are transferring short messages from sender to verifier and from verifier to receiver of 128-bit length and this mode is beneficial, secured and faster. It is beneficial due to its finalization step in which it removes duplicate invocations caused due to un-optimized message padding, it is done with the second 128-bit key. Total Key length of PC-MAC-AES is 256 bits i.e. 128-bit key of AES (K) and an independent 128-bit key (L). It also consists of two parameters first, tag length (π), its value lie between 1≤ π≤ 128 but we recommend π≥64 for security and second, a positive integer called order (d), it should lie between1 to 5 for secure implementation. The function of 4-round AES is denoted by G_u where $U= U^{(1)}||U^{(2)}||U^{(3)}$is the key. After calculating Key U now we have 4 rounds through which are data will travel and at the end of 4th round will have output. Its working is done like, in first round bit length of a bit string (x) is given to G_u it omits first round sub key (all 0's) addition and performs Sub bytes, Shift rows, Mix columns. Then, this procedure is done three times more. For second round sub key is $U^{(1)}$, for third round sub key is $U^{(2)}$ and for fourth round sub key is $U^{(3)}$.

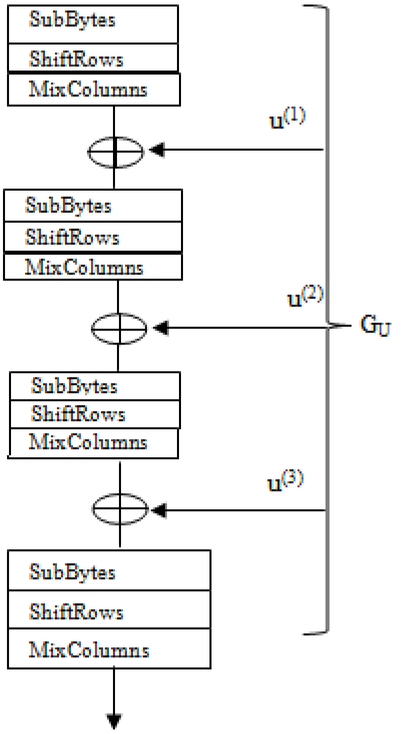


Figure 3: 4 rounds of AES with G_U

PC-MAC-AES $_{(\pi,\ d)}$ needs three procedure to ensure the message or a block of message that is being transmitted over an insecure network is authentic or not. The three procedures are as follows

1) Key Schedule
2) Tag Generation
3) Tag Verification

These procedures are described as below

a. Key Schedule

It is a preprocessing step used to generate authentication tag called KEYSCH using AES. After this, PC-MAC-AES can process a message or a block of message to generate authentication tag.

b. Tag Generation

It is called as TAGGEN. It is a procedure which generates π bit tag (T) and accepts message (M). Then pair of these two (M, T) will be transmitted to the receiver.

c. Tag Verification

It is called as TAGVER. Checking of authenticity of transmitted message to the receiver is verified through this verification procedure. It takes as inputs four parameters that are (M, T), π, and d, and after applying TAGVER procedure it produces binary output. If binary output is 1 that means the transmitted message is authentic but if binary output is 0 that means the transmitted message is not authentic. Now, we describe each procedure in detail.

Key Schedule

KEYSCH produces a string of 384.d+128. (d-1) bits using AES, which is the required authentication tag for a block of message. It works as follows

Firstly, produce d keys of 4-round AES function G_u. For that we need to compute E_k(L [0])|| E_k(L [1])||……..||E_k(L [3d-1]) and divide it into three blocks i.e. for i= 1,2,3…..d we use U_i = (E_k(L [3(i-1)])), (E_k(L [3(i-1)+1])),(E_k(L [3(i-1)+2])) for block 1, block 2,block 3 respectively. Basically partitioned into 384-bit sequences.

Secondly, produce d-1 128-bit supplementary keys. K_1^{XOR}=E_k(L⊕[3d]),……….., K_{d-1}^{XOR}=E_k(L⊕[4d-2]).
If d=1 then we are not required to calculate a supplementary key and KEYSCH produces 384-bit key, U_1.

```
Algorithm   KEYSCH (d, K, L)
fori← 1 to d
do {
$U_i^1$←$E_k$ (L⊕ [3(i-1)])
$U_i^2$ ←$E_k$ (L ⊕[3(i-1) +1])
$U_i^3$ ←$E_k$ (L ⊕[3(i-1) +2])
$U_i$←$U_i^1$||$U_i^2$ ||$U_i^3$
}
for j ←1 to d-1
do {
$K_j^{XOR}$ ←$E_k$ (L⊕ [3d +j-1])
Return( $U_1$, $U_2$,………,$U_d$,$K_1^{XOR}$ ,$K_2^{XOR}$ ……., $K_{d-1}^{XOR}$ )
```

Tag Generation

It generates π-bit tag T for any message of bit length M as shown in figure 4. Its input parameters are d, π, K, L, M, U_1…U_d,K_1^{XOR} …..K_{d-1}^{XOR} Core components of tag generation are AES encryption function that are E_k,G_{u1} and $G_{Ui}^{\oplus}$ and for i=2,……d and defined as
$G_{Ui}^{\oplus}$ (x)= Gui(K_{i-1}^{XOR} ⊕ x)
Firstly, we need to divide M into 128-bit strings M= M_1||M_2||…..M_{m-1}||M_m for i=1…..m-1. We have |M_i| =128, 1≤|M_m|≤128. Now, we are defining another intermediate function Ch, defined as Ch [F_1…..F_{m-1}](M)=pad(M_m) ⊕ F_{m-1}(M_{m-1}⊕F_{m-2}(………F_2(M_2⊕F_1(M_1))……)) where F_i is a keyed function of 128-bit inputs and outputs, defined as
$F_{(d+1)(i-1)+1}$=E_k, $F_{(d+1)(i-1)+2}$=G_{u1}, $F_{(d+1)(i-1)+3}$=G_{u2},….. , $F_{(d+1)(i-1)+d+1}$=G_{ud}, for i=1…..d.
Secondly, we need to produce 128-bit string h as h= Ch [F_1 …F_{m-1}] (M_1||M_2||…..||M_{m-1}||pad (M_m))
Thirdly, we will calculate π-bit tag T as

$$T=\begin{cases} cut_\pi (E_k (mul2 (L) \oplus h)) & \text{if } |x_m|=128 \\ cut_\pi (E_k (mul2 (mul2 (L)) \oplus h)) & \text{if} |x_m| <128 \end{cases}$$

Algorithm TAGGEN (d, π, K, L, M)

```
Partition M into $M_1$,$M_2$………$M_m$
If {
        m=1
        then h ←pad(M1)
    }
Else {
        S←$0^{128}$
        for  i← 1 to m-1
        do{
                w←(i-1)mod(d+1)
                if w=0
                then s ←Ek(s⊕ Mi)
                elseif w=1
                then s ←Gu1(s ⊕ Mi)
                else s←Guw(s ⊕ $K_{w-1}^{XOR}$ ⊕ Mi)
        }
        H←s ⊕ pad ($M_m$)
}
If |$M_m$| mod 128=0
        then h ←mul2(L) ⊕ h
        else h ←mul2(mul2(L) ⊕ h)
T ←$cut_\pi$ (Ek (h))
 Return (T)
```

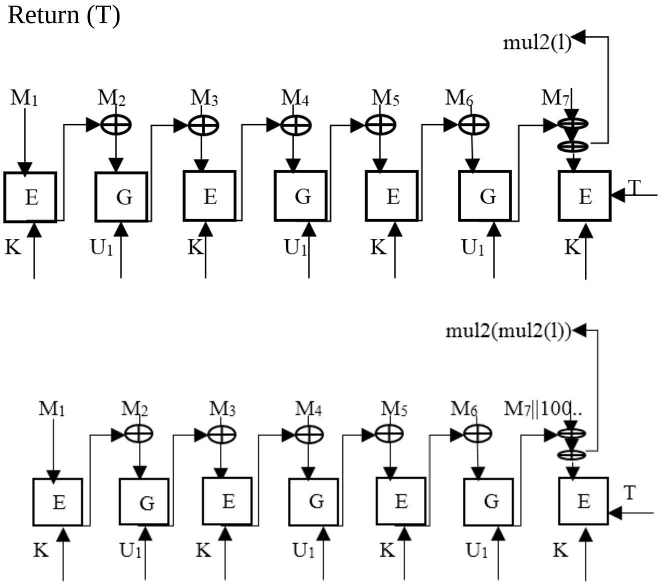


Figure 4: Tag GenerationofPC-MAC-AES (128, 1) (Top:the case|$M7$|=128,bottom:the case|$M7$|<128)

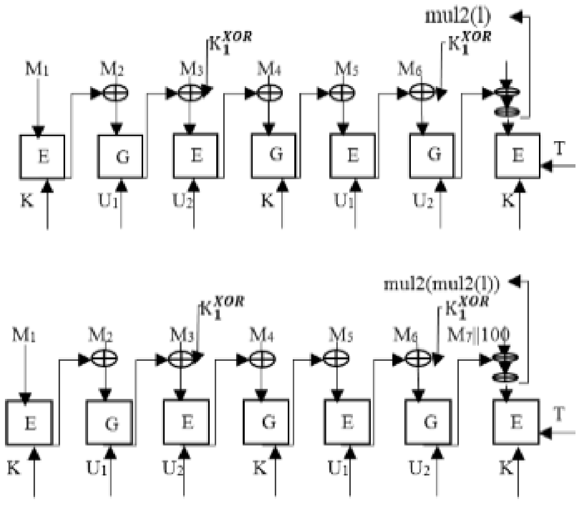


Figure 5: Tag GenerationofPC-MAC-AES(128,2)(Top :the case|$M7$|=128,bottom:the case|$M7$|<128)

Tag Verification

It is a standard deterministic MAC function. It invokes KEYSCH, receives a message pair (M`,T`), gives it to TAGGEN with parameters π and d and compares the output of TAGGEN with new generated T`. If they are same then message M` is authentic it gives output 1 and accepts the original message otherwise M` is not authentic it gives output 0 and rejects the original message.

```
Algorithm TAGVER (d, π, K, L, M, T)
        T`  TAGGEN (d, π, K, L, M)
        if T=T`
        then return (1)
        else return (0)
```

4. Implementation and Result

4.1Hardware Setup

As systems hardware consists of three elements these are, RFID tag, RFID reader and microcontroller as shown in figure 5. Tag which has data stored on it, reader which reads data stored on the tag andmicrocontroller with LED which is representing the data inside the tag.

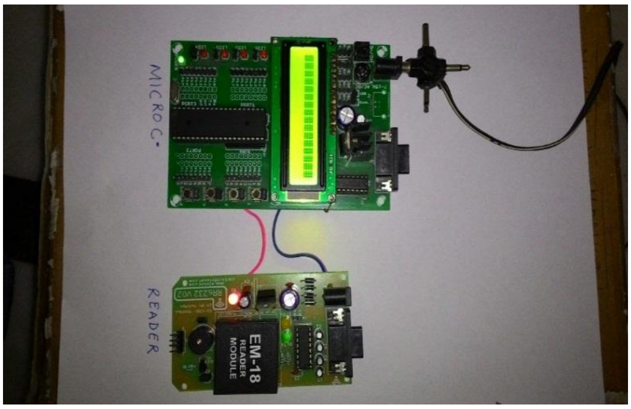
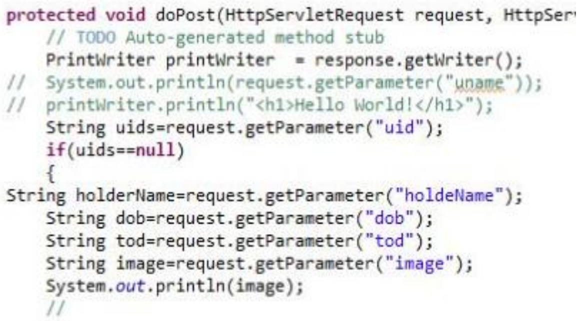


Figure 6: Microcontroller and Reader

4.2 Sever Setup

For any application to be run, first we need to setup server to start and synchronized with the local host to run the java application on it as sh.

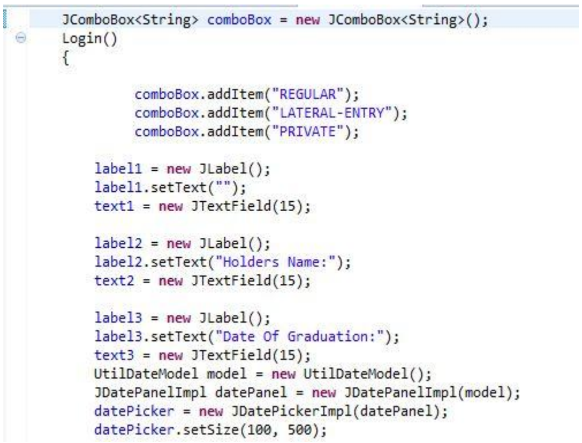


```
protected void doPost(HttpServletRequest request, HttpSer
    // TODO Auto-generated method stub
    PrintWriter printWriter  = response.getWriter();
//  System.out.println(request.getParameter("uname"));
//  printWriter.println("<h1>Hello World!</h1>");
    String uids=request.getParameter("uid");
    if(uids==null)
    {
String holderName=request.getParameter("holdeName");
    String dob=request.getParameter("dob");
    String tod=request.getParameter("tod");
    String image=request.getParameter("image");
    System.out.println(image);
    //
```

Figure 7: Coding of Server setup

4.3Enrollment

It is primary process of collecting data from the student who has taken the information. Basic information like name, date of birth, signature and retina scan is done in this and it is filled in the form manually. As it has shown in Figure 8. It shows the various data those are needed for creating secured certificates are asked in the form. It shows the backend code of enrollment form for submission of user's data. On a user level it is the very first step that takes place for development of the document it.

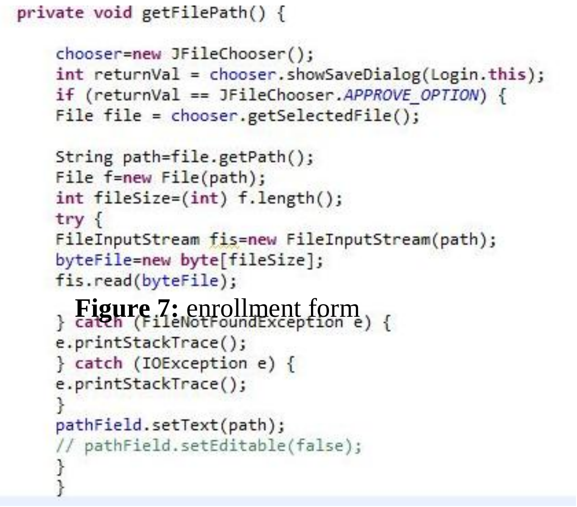


```
    JComboBox<String> comboBox = new JComboBox<String>();
    Login()
    {

            comboBox.addItem("REGULAR");
            comboBox.addItem("LATERAL-ENTRY");
            comboBox.addItem("PRIVATE");

        label1 = new JLabel();
        label1.setText("");
        text1 = new JTextField(15);

        label2 = new JLabel();
        label2.setText("Holders Name:");
        text2 = new JTextField(15);

        label3 = new JLabel();
        label3.setText("Date Of Graduation:");
        text3 = new JTextField(15);
        UtilDateModel model = new UtilDateModel();
        JDatePanelImpl datePanel = new JDatePanelImpl(model);
        datePicker = new JDatePickerImpl(datePanel);
        datePicker.setSize(100, 500);
```

Figure 8: Coding behind Enrollment Form

4.4UID Setup

When student gets enrolled in the required institute then corresponding to that student information one unique id generated randomly. That UID can be taken as primary key by which we can clearly differentiate or we can uniquely identify each and every student. This ID going to help us in authorizing the student at each level, no forgeries, and no duplicity will be entertained. This will surely help in finding out if there is any gap in the studies, which may student can hide.



```
private void getFilePath() {

    chooser=new JFileChooser();
    int returnVal = chooser.showSaveDialog(Login.this);
    if (returnVal == JFileChooser.APPROVE_OPTION) {
    File file = chooser.getSelectedFile();

    String path=file.getPath();
    File f=new File(path);
    int fileSize=(int) f.length();
    try {
    FileInputStream fis=new FileInputStream(path);
    byteFile=new byte[fileSize];
    fis.read(byteFile);
    } catch (FileNotFoundException e) {
    e.printStackTrace();
    } catch (IOException e) {
    e.printStackTrace();
    }
    pathField.setText(path);
    // pathField.setEditable(false);
    }
    }
```

Figure 7: enrollment form

Figure 9: UID generation

4.5Authentication

Here, we are offering 1:1 matching as let say student is going for a job he or she having a degree with RFID chip inserted into it, reader will going to read that certificate only if that certificate chip is made compatible with that reader (through software or public key of university) then it will going to be read, here it is providing security. Now, authentication is done by third party, upon accepting a job student will present his or her degree to the employer, employer will see its name, course, date of graduation, and type of degree. Then employer send this data to the third party (merchant) third party will match this data corresponding to the required UID, if it matches and the required student face matched with certificatepresented student and fingerprint with its demographic information stored inside the database matches with the live fingerprint then our system in return will provide 'yes' in authentication form, if it doesn't matches then it will return 'no' that meansstudent is not verified, and is not authentic. After that we can cancel its candidature.

4.6RFID Processing

RFID module working in figure 9 and figure 10, as we have shown that firstly it scans the tag, reader reads the data it emits

radio waves and that waves makes tag activate and transfer the data and in figure 10 the data is being printed on that LED of micro controller.

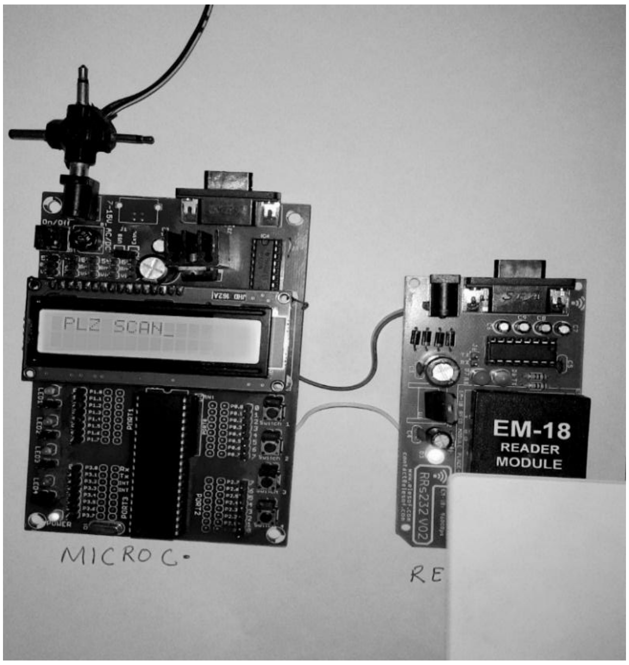


Figure 10: Scanning of RFID tag

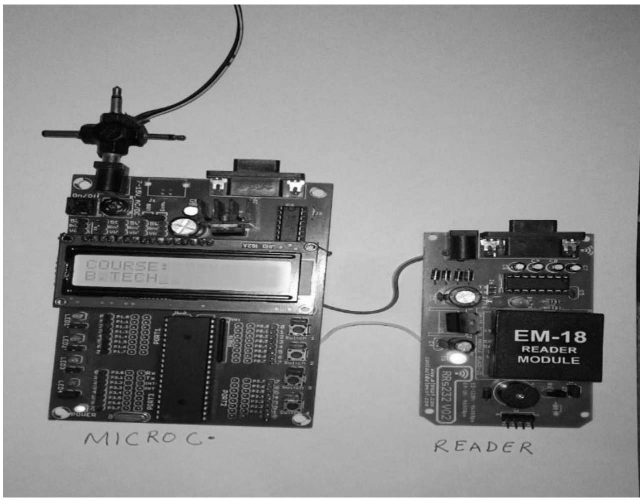


Figure 11: Data read by reader&Certificate Verification

Table 1: Result Analysis

	System with RSA	System with AES	System with PC_MAC_AES
Encryption per iteration (in one round)	Bit by bit	128 all together	128 all together
Key length	Generates key length according to the plain text	128, 192, 256	128
Rounds	Depends on generation of key	10	4
Function	Provides security add on with digital signature	Provides speed in calculating	Provides both security and speed
Time latency	0.18 ms	0.1118 ms	0.0016s

RFID in combination with algorithm PC_MAC_AES offers best possible result for this system. Its read rate is high, due to this system offers high throughput, we can scan multiple certificates simultaneously instead of one at a time. It gives better latency as compared to other. This seems to be like we have electronic passports [9] and identity cards [10] through which we offer people authenticity. RFID enabled secure certificate is another method for proving authenticity like these but an innovative method.

5. Conclusion

In this study, we have implemented a digital security system contains RFID enabled secure certificate using passive RFID tag. A centralized system is being deployed for controlling and transaction operations. The RFID enabled secure certificate will work in real time as when the user put its RFID tag containing certificate in the contact of reader, the user's information is been authenticated through the system and it shows that it does not belong to a false person, the information is been checked through the data stored in central server in an encrypted form along with authentication is been done through third party which validates the data. We utilize RFID technology to provide solution for secure access of information of user (student) while not allowing any discrepancies.

References

[1] Stephen B.Miles, Sanjay E.Sarma, John R.Williams, "RFID technology & applications", Massachusetts Institute of Technology, Cambridge University Press, 2011.

[2] Stephen A.Weis "Radio Frequency Identification: Principles and Applications", MIT CS AIL, 2008.

[3] Arun A. Nambier, "RFID Technology: A review of its applications", world congress on Engineering and Computer Science, Vol 2, 2009.

[4] Roy Want, "An Introduction to RFID Technology," IEEE CS and IEEE ComSoc, Vol. 5, No. 1, Santa Clara, 2006, PP. 25-33.

[5] Chandrasekhar Mudraganam, "Smart Degree from TCS to combat certificate malpractices", 2009.[6] AtulKahate "Cryptography and Network Security", Tata McGraw-Hill Companies, 2008.

[6] William "Network Security Essentials (Applications and Standards)", Pearson Education, 2004.

[7] Self Evaluation Report of PC_MAC_AES, NEC Corporation, 2010.

[8] Karl koscher, VjekoslavBrajkovic&TAdayoshi Kohno, “EPC RFID tag security weaknesses & Defenses: Passport cards, Enhanced Driver Licenses &Beyond”, University “Smart card based identity cards and security white paper”, Jan Kremer Consulting Services.

Author Profile

Sonali Singh is M.Tech Student at Sharda University in CSE- NW. Graduated in CSE from BIT Meerut UP in year 2013.