Securing e-mails in XML format using colors and Armstrong numbers

Prof. Mr. S. A. Saoji, Nikita B. Agarwal, Mrunal B. Bokil, Ashwini V. Gosavi,

Abstract— Cryptographically signed email has been widely used to provide the end-to-end authentication, integrity and non-repudiation. PGP mail and S/MIME have the significant drawback that the headers are unauthentic. Domain Keys Identified Mail (DKIM) protects specified headers, however, only between the sending server and the receiver. These lead to possible impersonation attacks and profiling of the email communication, and encourage spam and phishing activities. Furthermore, none of the currently available security mechanisms supports signature generation over partial email content by distinct signers, which might be useful in commercial scenarios. In order to handle these problems we suggest a new approach which can be considered as an advanced email security mechanism based on the popular XML technology. Our approach supersedes currently available email security standards in the sense of the higher flexibility and security, and can be transported via Web Services easily.

This paper provides a technique to encrypt the data using a key involving Armstrong numbers and a color as the password. Three set of keys are used to provide secure data transmission with the colors acting as vital security element thereby providing authentication.

Index Terms — HSV(Hue Saturation Value) color model ,RSA(Rivest Shamir Adleman), SHA-1 (Secure Hash Algorithm),RGB(Red Green Blue) color model ,XML(Extensible Markup Language)

1 Introduction

In the present world scenario it is difficult to transmit data from one place to another with security. This is because hackers are becoming more powerful nowadays. To ensure secured data transmission there are several techniques being followed. One among them is cryptography which is the practice and study of hiding information.

Email messages are not protected as they move across the Internet. Often information being transmitted is valuable and sensitive such that effective protection mechanisms are desirable in order to prevent information from being manipulated or to protect confidential information from being revealed by unauthorized parties. A large number of email security mechanisms have been meanwhile developed and standardized, building a solid basis for secure email communication. Based on the analysis, we make it clear that further improvement of the currently existing formats is needed. We describe a novel solution that unifies strengths of the previous approaches and provides additional attractive features for higher flexibility of email communication achieving the desirable security properties.

Emails are a very important form of communication in day to day life. Many transactions and important information transfers as well as simple communications take place through emails. Thus, protecting the data contained in the emails becomes mandatory. By taking into account the extent to which the data contained in the emails can be misused (whether working online or offline) providing security, both to online as well as offline email usage is of prime importance. The MS Outlook works in a similar way. The difference being that MS Outlook does not provide any security to the emails. A textual password is accepted while linking the email account with Outlook. Therefore the data becomes vulnerable and it can be misused.

2 PROPOSED APPROACH

We design a framework to exchange email in XML format,called XML Email, that achieves the advantages of both XML and email: code efficient for processing, archiving and searching, end-to-end security of complete message, simple implementation of clients, readability of the signature and encryption for a natural person, multiple signatures over different contents, and transport via the existing systems.

The user needs to create an account where he is expected to enter all his details and select a color either from RGB or HSV color model to verify the authenticity of the receiver.

- *Author Mr. S. A. Saoji is currently a teaching faculty in information technology engineering in Pune University, India.E-mail: sarang.iiita2008@gmail.com*
- *Co-Author Nikita B. Agarwal is currently pursuing bachelors degree program in information technology engineering in Pune University, India. E-mail: nikita.ba12@gmail.com*
- *Co-Author Ashwini V. Gosavi is currently pursuing bachelors degree program in information technology engineering in Pune University, India. E-mail: ashvini.g@hotmail.com*
- *Co-Author Mrunal B. Bokil is currently pursuing bachelors degree program in information technology engineering in Pune University, India. E-mail: mrunalb.25@gmail.com*

3 WORKING ONLINE WITH THE APPLICATION

3.1. ENCRYPTION:

1. Digital signature of message contents : Digital Signature of the message contents is produced using SHA-1 which is a hashing algorithm.
2. Encrypt message contents using receiver's public key : Using RSA algorithm, the message contents are encrypted. RSA uses public key of the receiver which is already stored on the server.
3. Register sender's color : For the recipient to confirm that the message being received is intended to be received by him, the sender's color should be registered on the server.
4. Add Armstrong code to data contents : Armstrong code is added to the encrypted message in step b.
5. Upload data.

3.2 DECRYPTION:

1. Download message
2. Decrypt Armstrong code : Armstrong code is removed from the encrypted message.
3. Confirm message authenticity : The receiver.
4. Decrypt message :The message is decrypted using the private key of the receiver.
5. Digital Signature verification : The Digital Signature of the decrypted message is produced and compared with the received digital signature to verify the message integrity.

4 WORKING OFFLINE WITH THE APPLICATION

After the e-mail is downloaded on the users' system instead of storing it in the original format the contents are encrypted in the following way and then stored. The user needs to select an Armstrong number when he logs in to the system for the first time and also set a password. Whenever the user wants to access the e-mails , he needs to enter the password.

4.1 ENCRYPTION

Step 1: (Encryption of the actual data)
Let the message to be transmitted be "CRYPTOGRAPHY".
First find the ASCII equivalent of the above characters.

C	R	Y	P	T	O	G	R	A	P	H	Y
67	82	89	80	84	79	71	82	65	80	72	89

Step 2: Suppose the Armstrong number is 153. Now add these numbers with the digits of the Armstrong number as follows

```
     67  82  89  80  84  79  71   82   65  80  72  89
(+)  1   5   3   1   25  9   1    125  27  1   5   3
  -----------------------------------------------------------
     68  87  92  81  109 88  72   207  92  81  77  92
```

Step 3: Convert the above data into a matrix as follows

$$A=\begin{bmatrix}68 & 81 & 72 & 81\\87 & 109 & 207 & 77\\92 & 88 & 92 & 92\end{bmatrix}$$

Step 4: Consider an encoding matrix

$$B=\begin{bmatrix}1 & 5 & 3\\1 & 25 & 9\\1 & 125 & 27\end{bmatrix}$$

Step 5: After multiplying the two matrices (B X A) we get

$$C=\begin{bmatrix}779 & 890 & 1383 & 742\\3071 & 3598 & 6075 & 2834\\13472 & 16082 & 28431 & 12190\end{bmatrix}$$

The encrypted data is...
779, 3071, 13427, 890, 3598, 16082, 1383, 6075, 28431,742, 2834, 12190
The above values represent the encrypted form of the given message. This is stored in the database.

4.2 DECRYPTION

Now , when the user wants to access his e-mails offline, he can view only the sender of the mail and for viewing the contents of the mail he needs to enter the Armstrong number in the reverse order so as to proceed with the decryption of the contents. And only if the reverse Armstrong number is correct the data gets decrypted.

Step 1: (Decryption of the original data)
The inverse of the encoding matrix is

$$D = (-1/240)^{*}\begin{bmatrix}450 & 240 & -30\\18 & 24 & -6\\100 & -120 & 20\end{bmatrix}$$

Step 2: Multiply the decoding matrix with the encrypted data (D X C) we get

$$\begin{bmatrix}68 & 81 & 72 & 81\\87 & 109 & 207 & 77\\92 & 88 & 92 & 92\end{bmatrix}$$

Step 3: Now transform the above result as given below
68 87 92 81 109 88 72 207 92 81 77 92

Step 4: Subtract with the digits of the Armstrong numbers as follows:
68 87 92 81 109 88 72 207 92 81 77 92

```
(-) 1  5    3  1   25    9    1 125   27    1    5    3
----------------------------------------------------------
   67  82  89  80   84  79  71   82   65  80  72  89
```

Step 5: Obtain the characters from the above ASCII equivalent
67 82 89 80 84 79 71 82 65 80 72 89
C R Y P T O G R A P H Y
In this way we get the required contents back in the original form.

5 ADVANTAGES

1. The application uses RSA algorithm which has the biggest advantage that it uses public key.
2. Trapdoor of RSA is in knowing value of n (where n =p*q, p and q are prime numbers) but not knowing the primes that are factors of n.
3. SHA-1 can be applied to a block of any size but still it produces an output of fixed size i.e.160 bits.
4. SHA-1 has strong collision resistance as it is computationally infeasible to find any pair (x, y) such that Hash code of x is exactly same as hash code of y i.e. H(x) = H(y).

6 DISADVANTAGES

1. More space is required on server side because of RSA.
2. The speed of execution is slow because the file size after encryption is 8 times the original size.
3. The only way way to break into this system is by Brute force attack, which also can take up to two or three years.
4. To protect the encryption , the minimum number of bits in n(n in RSA) should be 2048.

7 CONCLUSION AND FUTURE SCOPE

To reduce the file size or to increase the execution speed, we can also include different algorithms like AES, DES etc. A provision is made to replace RSA algorithm by other alternatives by changing the index. Although other algorithms can be used, RSA proves to be the strongest algorithm in terms of security. Another addition in this project can be working on the attachments received in the e-mails.

REFERENCES

[1] S. PavithraDeepa, S. Kannimuthu, V. Keerthika, Security using colors and Armstrong numbers,Proceedings of the National Conference on Innovations in Emerging Technology-2011.

[2] Atul Kahate, Cryptography and Network Security , Tata McGraw Hill Publications

[3] Lijun Liao,Jorg Schwenk, Secure Emails in XML Format Using Web Services, Fifth European Conference on Web Services

[4] Nina Godbole, Information Systems Security, Wiley India Pvt Ltd, ISBN -978-81-265-1692-6

[5] Berouz Forouzan, Cryptography and Network Security, 2 edition, TMH, ISBN :9780070702080

[6] Enterprise Cloud Computing by Gautam Shroff,Cambridge, ISBN :978-0-521-13735-5

[7] RESTfulWeb Services, O'Reilly Media, ISBN : 978-0-596-52926-0