ENCIPHERING AND DECIPHERING THE COOKIE DATA USING RECTANGULAR ARRAY

Akkimsetti Mohana Sai Chandra

Student, Dept of Electronics and Communication Engineering, School of SEEE, SASTRA University, Thanjavur, Tamil Nadu, INDIA

---***---

Abstract- Cookies contain sensitive information and also, they are being stored and transferred to web servers through public channels, there is a high risk of stealing and manipulating cookies. So, in this paper a new way of encrypting cookies and decrypting of cookies is analyzed. Since, all cookies are in the form of text, encryption and decryption is made easier by this new algorithm/new way. This algorithm uses the concepts of mathematics and in particular about matrices and their properties. This paper describes an algorithm which encrypts the cookie data(text) into a matrix and decrypts the encrypted matrix back into text.

Key Words**: Cookie data encryption, Cookie data decryption, Key generation algorithm, Matrices**

1. INTRODUCTION

Now a days surfing the internet is the most common thing everywhere. So, intentionally or unintentionally we may open new websites or apps every day. Most of the websites uses cookies for better user experience and better security measures, Joon S. Park and Ravi Sindhu [3]. Generally, cookies are used to authenticate the user and monitor the user. Different websites have different cookies, means that one website cookie can't be accessed or used by another website [5]. Cookie is a text file which is stored on our device hard disk. Cookie transfer occurs every time the website has been used between the user and server. Since cookies being transferred through public channels or servers there is a high risk of hijacking and manipulating cookie data [7]. Generally, cookies contain information like user credentials, how much time has the user is on that website, User preferences, wish lists and payment methods and preferences, credit card numbers etc. This indicates that the cookie data must be secured. The cookie size is restricted to 4 kilobytes and a maximum of 50 cookies are allowed for a particular website and the cookies are different for different devices.[9]. There is need to encrypt the cookies [4]. Encryption of cookies is somewhat reliable because the size of a cookie is restricted to 4KB, which allows a maximum of 4093 characters [9].

The proposed algorithm ensures the safe propagation of cookies from user to server without revealing the original data to public. Since COOKIE is text file, we can encrypt the cookie with text encryptions. Hill Cipher algorithm encrypts the data using linear algebra. Hill cipher algorithm is a polygraphic substitution cipher, which was developed by Hill cipher in 1929 [23] in which each letter of the text is substituted with a number in Z26 [11]. Polyalphabetic Substitution ciphers which were developed by Leon Battista Alberti,1467 also are used for encryption in which the text is partitioned and then substituted with the key related numbers [12]. This polyalphabetic cipher includes methods like Vigenere Cipher, One-time pad etc. [13]. Large sized data will be at risk by these Substitution ciphers as the substitution occurs only related to Z26.Ancient Greeks developed transposition cipher which encrypts the data by shifting the text bits based on the key [14]. The general cookie flown is shown in Fig 1.

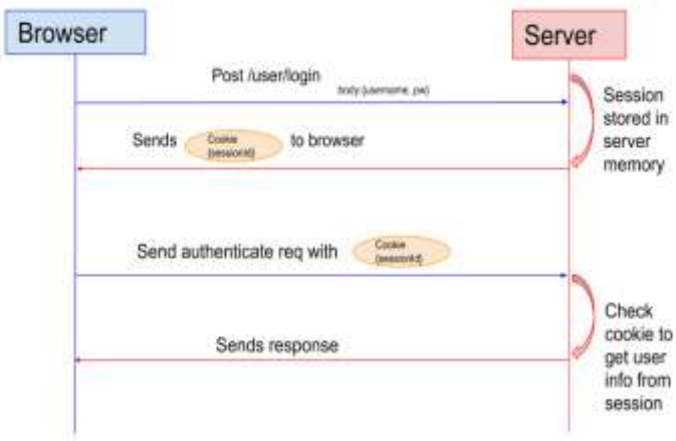


Fig 1. A typical Cookie transmission and reception through server

Source: Adapted from [1]

Apart from these the most popular one is the Data Encryption Standard (DES) which is a Block cipher developed by IBM in 1970.[15], and encrypts the data in 16 Feistel rounds using permutation and combinations. It was the most used encryption algorithm and also accepted by the US government. But the DES uses 56-bit key and was used only to encrypt 64-bit data, which was converted into blocks. The successor of DES is Triple DES which encrypts the 64-bit data using 3 different 56-bit keys by IBM ,1977[16]. Later on, Rivet, Shamir and Adleman developed an encryption algorithm which uses different keys which initiated the Asymmetric key Cryptography [17]. Rivest-Shamir-Adleman (RSA) algorithm is a very powerful algorithm at that time when it came into existence. RSA algorithm involves Algebraic functions and modulo operators [18]. Another revolution is made in cryptography is Hashing which was developed by Ron Rivest in 1989. Hashing is different from Encryption and decryption. In encryption and decryption, the original data is obtained back whereas in hashing the original data cannot be obtained back. Hashing uses hash algorithm which produces a unique hash value for each data [19]. But in cookie encryption the data should be obtained at the website server or client to review the user data. Hashing plays an important role in block chains and exclusively in payment related operations in today's world [20]. Hashing algorithm is very fast compared to encryption and decryption algorithms. In 1990, a new cryptographic technique called Quantum Cryptography has seen a rise in encryption cryptography field which was the work by Stephen Wiesner and Gilles Brassad,1970 [21]. This particular concept is inspired from quantum mechanics and it is extended to secure messages. The present known Quantum encryption method is Quantum key distribution (QKD) where the information encoded on single photons [24].

2. NEED OF COOKIE ENCRYPTION:

Since our sensitive information is being stored in a cookie, it's an important thing to protect the cookie data. In today's modern world the data or information is most important thing. For years many of the developers worked on the concept called SECURE COOKIES [22] but recently the cookies faced MIM attacks (Man-in-the-middle attack), Session hijacking [2] and also faced attacks called as Cookie Poisoning [22]. Cross-site Scripting (XSS attack) is the most common method to steal cookies. Also, cookies are prone to attacks like brute force attack, replicating authentication cookie [22]. So, the data in the cookie should be protected from being exposed to the attackers. Some of the cookies may also contain our credit card numbers which poses major concern about the security. So, clearly there is a need of encrypting our data which is in the form of cookie [8]. As mentioned above in today's world many algorithms are available for encryption, but the thing is for any encryption we have to declare a key (either private/public key) in order to encrypt/decrypt the data. But when it comes to cookies it is not possible to give a separate key for each and every cookie and memorizing in order to decrypt them. So, the cookie needs to be encrypted by itself and also decrypted automatically (Encryption and decryption should be done automatically) without allowing any attacker to decrypt them if they even hijacked.

3. ALGORITHM

Mathematics is the field where cryptography is born and the principles of mathematics can be used very effectively in building a powerful encryption and decryption algorithm [6]. Cryptography using arrays is a unique approach in the field of cryptography [6]. This algorithm is purely based on mathematics, highly on matrices concepts. Every letter or element is encrypted into a matrix element. The order of the matrix is determined by the size of the input data, and also the transformation makes use of the American Standard Code for Information Interchange (ASCII) value in computer language in order to transform alphabets to numbers. We know that there are 256 ASCII values. So, every value can be encrypted and decrypted using this algorithm. For encryption and decryption of data the basic property of matrices is used here. Since the data is transformed into ASCII values this can be referred as Transformative Cryptography.

$$[A]^{-1} = \frac{1}{|A|} * Adj(A)$$

So, this can be written as,

$$|A| * I = [A] * Adj(A)$$

Where, *I* is the Identity matrix

|A| is determinant of matrix

[A] is a matrix of order n

Adj(A) is Adjoint matrix of A

We also know that,

$$(B * A)^{-1} = B^{-1} * A^{-1}$$

Encryption Algorithm,

$$[Coded\ Matrix] = [Message\ ASCII\ Matrix] * \{|Key| * I\}$$

In this Encryption algorithm the ASCII equivalent matrix is multiplied with the identity matrix which was multiplied with the determinant value. So, the resultant matrix is an encrypted form of our data (string). Here, the key used is the identity matrix which is multiplied with the determinant value.

Decryption Algorithm,

$$[Message\ ASCII\ Matrix] = [Coded\ Matrix] * [Key\ Matrix]^{-1} * [Adj(Key\ Matrix)]^{-1}$$

For decryption we have to multiply the coded data (encrypted matrix) with the inverse of the key matrix and its adjoint matrix (Adj(key)). Here the key is the key matrix not the identity matrix. Message ASCII Matrix is the equivalent ASCII elements of each alphabet in the data. Note that the key used for encryption is different the key used for decryption. So, this can be classified as Public-key encryption. This algorithm uses ASCII values for transformation of text to decimal values. The equivalent ASCII values of characters can be determined from Fig 2.

The order of the matrix is fixed by the length of the data (input string). By taking square root of the size of the string and adding 1 to the result will give us the order.

$$Order\ of\ the\ Matrix = \sqrt{size\ of\ the\ stirng} + 1$$

Fig 2. ASCII Chart [10]

Source: Adapted from [10]

Example:

Let us encrypt and decrypt a data using this algorithm. Consider the word COOKIE. It is of the length 6 so the nearest square matrix is of order 3. The code given below choses automatically the order of the matrix based upon the size of the data. The ASCII Matrix of the word COOKIE is

$$Message\ Matrix = \begin{matrix} 67 & 79 & 79 \\ 75 & 73 & 69 \\ 32 & 32 & 32 \end{matrix}$$

The remaining elements are filled by the default number "32" because **ASCII 32 represent a "Blank Space".** Let the key matrix be

$$Key\ matrix = \begin{matrix} 2 & 3 & 0 \\ 5 & 2 & 1 \\ 2 & 1 & 2 \end{matrix}$$

$$Det\ of\ Key\ Matrix = \ -18$$

The Encrypted Matrix is,

$$Coded\ Matrix = \begin{matrix} 67 & 79 & 79 \\ 75 & 73 & 69 \\ 32 & 32 & 32 \end{matrix}$$

$$* \begin{matrix} -18 & 0 & 0 \\ 0 & -18 & 0 \\ 0 & 0 & -18 \end{matrix}$$

$$Coded\ Matrix = \begin{matrix} -1206 & -1422 & -1422 \\ -1350 & -1314 & -1242 \\ -576 & -576 & -576 \end{matrix}$$

We didn't use the Key Matrix directly for encryption, we have used only the determinant value of the key matrix. So, to get the Message matrix back

$$Adj(Key) = \begin{matrix} 3 & -6 & 3 \\ -8 & 4 & -2 \\ 1 & 4 & -11 \end{matrix}$$

$$(Key)^{-1} = \begin{matrix} -0.1667 & 0.3333 & -0.1667 \\ 0.4444 & -0.2222 & 0.1111 \\ -0.0556 & -0.2222 & 0.6111 \end{matrix}$$

$$(Adj(Key))^{-1} = \begin{matrix} -0.1111 & -0.1667 & -0.0000 \\ -0.2778 & -0.1111 & -0.0556 \\ -0.1111 & -0.0556 & -0.1111 \end{matrix}$$

$$Message\ Matrix = [Coded\ Matrix] * [Key]^{-1} * [Adj(ke$$

$$= \begin{matrix} 67 & 79 & 79 \\ 75 & 73 & 69 \\ 32 & 32 & 32 \end{matrix}$$

$$= COOKIE$$

But as mentioned above the key should be automatically generated and the process of encryption and decryption of data should be done automatically. For that purpose, we have to build an algorithm which produces a matrix of a given determinant value

The KEY MATRIX GENERATION ALGORITHM: The General Structure of the matrix is

Table 1: Structure of Key Matrix

2	1	D	D	D	D	-----	D
2	2	1	D	D	D	-----	D
2	2	2	1	D	D	-----	D
2	2	2	2	1	D	-----	D
2	2	2	2	2	1	-----	D
|	|	|	|	|	|	|	|
x	x	x	x	x	X	x	x+1
x+3	x+3	x+3	x+3	x+3	x+3	x+3	x+5

D-Don't care (Any number can be inserted)

$$x = Determinent + 3$$

For a given determinant value this algorithm produces a Matrix of equal determinant valueSince the cookie maximum size allowed is 4KB (4093 bytes) the maximum possible matrix order is 64 which can be processed easily by modern processors or computers.

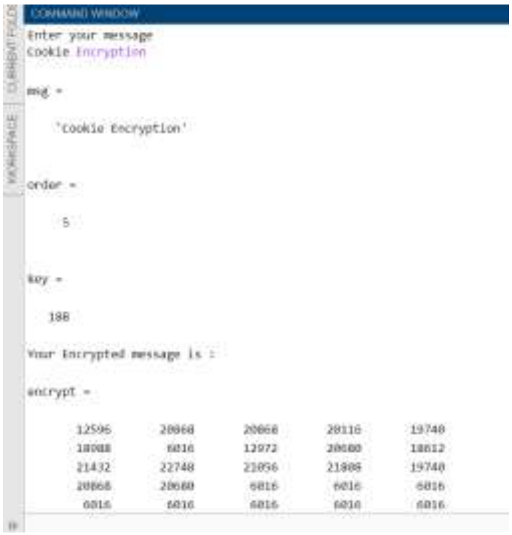
THE MATLAB CODE FOR THE ABOVE ALGORITHM (KEY GENERATION INCLUDED):

%ENCRYPTION

```
msg=input ('Enter your message','s')
                              %Taking Input

l=strlength(msg);
```

```
x = l;
n=floor(sqrt(x))+1;
message=0;
for i=1:n,j=1:n;
   message(i,j)=32;
end
 k=1;
for i=1:n
   for j=1:n
     if(k<=l)
       message(i,j)=double(msg(k)); %Formation of ASCII Matrix
     end
    k=k+1;


   end
end
order=n
key=randi(230) %Random Determinant value
encrypt=0;
disp("Your Encrypted message is :")
encrypt=message*key*eye(n,n) %Encryption Algorithm
%%DECRYPTION
dec=input('Do you want to decrypt [press d]','s') %Asking for decryption
            if(dec=='d')
   disp('Your msg is decrypting')
dett=key
order=n
if(order==2)
   m=dett+3;
   matt=[m m+1;m+3 m+5]
elseif(order==3)
   m=dett+3;
      matt=[2 1 6;m m m+1; m+3 m+3 m+5]
elseif(order==4)
   m=dett+3;
      matt=[2 1 6 9;2 2 1 5;m m m m+1;m+3 m+3 m+3 m+5]
elseif(order==5)
   m=dett+3;
      matt=[2 1 6 9 2;2 2 1 5 4;2 2 2 1 15;m m m m m+1;m+3 m+3 m+3 m+3 m+5]
elseif(order==6)
   m=dett+3;
       matt=[2 1 6 9 2 17;2 2 1 5 4 23;2 2 2 1 15 12;2 2 2 2 1 19;m m m m m m+1;m+3 m+3 m+3 m+3 m+3 m+5]
elseif(order==7)
   m=dett+3;
       matt=[2 1 6 9 2 17 26;2 2 1 5 4 23 57;2 2 2 1 15 12 38;2 2 2 2 1 19 41;2 2 2 2 2 1 62;m m m m m m m+1;m+3 m+3 m+3 m+3 m+3 m+3 m+5]
end
keyver=matt;
      adjkeyver=det(keyver)*inv(keyver);
      decrypt=0;
      decrypt=encrypt*inv(keyver)*inv(adjkeyver) %Decryption Algorithm
      MSG=1:l;
      t=1;
      for i=1:n
         for j=1:n
            if(t<=l)
             MSG(t)=decrypt(i,j);
             t=t+1;
            end
         end
      end
    c=blanks(l);

    d=blanks(l);

    u=1;

    my_string = native2unicode(MSG,'ASCII')

else

   disp('Thank you,your message is encrypted')
           %Message String

end
```

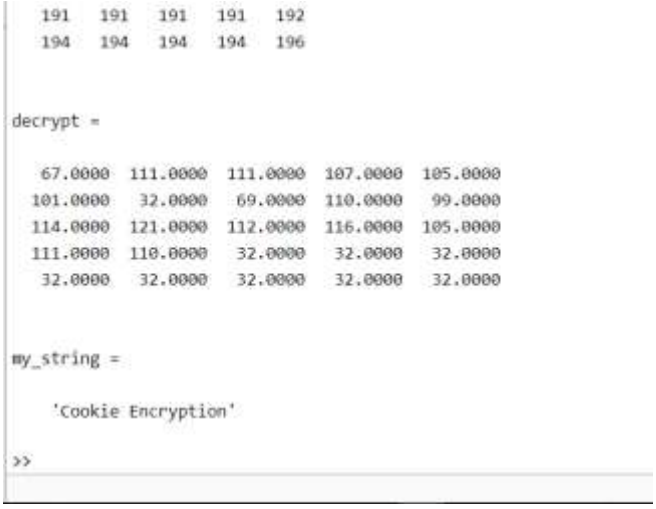
The above code is compiled using MATLAB R2019b Online Version and the following are the outputs



```
Enter your message
Cookie Encryption

msg =

    'Cookie Encryption'


order =

     5


key =

   188

Your Encrypted message is :

encrypt =

       12596        20868        20868        20116        19740
       18988         6016        12972        20680        18612
       21432        22748        21056        21808        19740
       20868        20680         6016         6016         6016
        6016         6016         6016         6016         6016
```

Fig 3. ENCRYPTION

“encrypt” represents the encrypted matrix in the above figure.



```
   191    191    191    191    192
   194    194    194    194    196


decrypt =

   67.0000  111.0000  111.0000  107.0000  105.0000
  101.0000   32.0000   69.0000  110.0000   99.0000
  114.0000  121.0000  112.0000  116.0000  105.0000
  111.0000  110.0000   32.0000   32.0000   32.0000
   32.0000   32.0000   32.0000   32.0000   32.0000


my_string =

    'Cookie Encryption'

>>
```

Fig 4. DECRYPTION

“my string” represents the decrypted string or data and “decrypt” represents the equivalent ASCII matrix of the give data (“Cookie Encryption”)

4. RESULTS:

By using this algorithm, we have encrypted the data into a nxn order matrix. The result of the encryption is a matrix whose order is based upon the size of the text or given input. The resultant matrix of the encryption contains the ASCII character values of the data multiplied with the key elements. For decryption of data, encrypted matrix is converted into text based upon the decryption algorithm which uses a key related to the encryption key. For key generation this algorithm uses a general format to produce required order of the matrix. The encrypted matrix can’t be decoded without the function of encryption key. The key generation function in encryption uses a random value generator function which cannot be predicted by anyone.

DISCUSSIONS: As mentioned above cookie encryption should be done automatically without the user interference, and also the cookies should get encrypted very quickly to avoid time delay for accessing website. Since so many advanced algorithms exists, the main thing in them is that they are that much complex which takes time for the whole process i.e., encryption and decryption to complete. The resultant ciphertext should also be of a maximum size of 4kB, which fails for some methods or algorithms as they produce higher sized ciphertext than plaintext. This algorithm is fast enough because it produces the ciphertext or encrypted matrix size based on the data, and the maximum order of the matrix can be up to 64 (since 4096 bytes) which can be handled easily by toady’s modern-day processors, and also the key used here is also data size dependent and the key size varies from data to data. Substitutions encryptions and transposition encryptions fails in Case-Sensitive text as they based on Z26, whereas this method can handle even Case-Sensitive text because it is based on ASCII values as ASCII values are valid for all characters (256 characters). In addition to this, this algorithm produces both encryption and decryption key by itself and doesn’t depend on the user for manual input. This also enables the server or client to review the data of the user which improves their services to user. This gives an additional layer of protection in website security.

5. CONCLUSION

A new algorithm or a new way to encrypt and decrypt the data has been developed and analyzed in this paper, and observed that it could be used effectively and efficiently for encryption and decryption of cookie data .Since there is a possibility of cookies being hacked and get manipulated, this algorithm is better in avoiding those things as it involved different keys and matrix properties which makes hard to decrypt for an attacker. It involves matrix multiplication process it become very hard to decrypt. By encrypting and decrypting the cookie data provides additional layer of protection against attackers and can be helpful for secure browsing.

V.REFERENCES

[1].Fig1.Imagelink:https://micro.medium.com/max/1530/1 *Hg1gUTXN5E3Nrk u0jWCRow, png

[2]. "Detecting Third-part User Trackers with Cookie Files" in proceedings of Valery Dudykevych, Vitalii Nechypor in October 4-6,2016 IEEE Conference.

[3]. R. Tirtea, C. Castelluccia and Ikonomou, Bittersweet cookies. Some security and privacy considerations Greece: European Network and Information Security Agency,2011

[4]. "Analysis and Compliance Evaluation of Cookies-Setting Websites with Privacy Protection Laws" by Adeyemi Aladeokin, Pavol Zavarsky, Neelam Memon in proceedings of IEEE,2017

[5]. "Using Browser Cookies for Event Monitoring and User Verification of an account" in proceedings of IEEE,2018

[6]. "A Study on Data Encryption and Decryption using HILL CIPHR Algorithm" by Maheshwari, A. Kaushika, A. Jenifer in International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES).

[7]. "Secure Cookies on the Web" by *Joon S. Park and Ravi Sandhu* in proceeding of IEEE,2000

[8]. Securing HTTP cookies by Camilo Reyes in jscrambler blog, August 07,2017

[9]. Browser cookie limits by squawky,net

[10]. Fig 2. Image link: https://theasciicode,com/ar

[11]. Practical Cryptography, practicalcryptography.com, [Online]Available: http://practicalcryptography.com/ciphers/hill-cipher/

[12]. Polyalphabetic Substitution Ciphers, Crypto corner Available at: https://crypto.interactive-maths.com/polyalphabetic-substitution-ciphers.html

[13]. Cryptography and Network Security by *Prakash C.Gupta,* Dept of Information Technology, Maharashtra Institute of Technology, Pune

[14]. Transposition cipher, En.wikipedia.org, [Online] Available: https://en.wikipedia.org/wiki/Transposition_cipher

[15]. DataEncryptionAlgorithm, Umsl.edu. [Online] Availble: http;//www.umsl.edu/~siegelj/information_theory/projects/des.netau.net/Dataencryptionalgorithm.html

[16]. TripleDES, Tutorialspoint.Available: https://www.tutorialspoint.com/cryptography/triple_des.htm

[17]. RSA (cryptosystem), Available: https://en.wikipedia.org/wiki/RSA_(cryptosystem)

[18]. Eddie Woo,25 January 2020, The RSA Encryption Algorithm (1 of 2: Computing an Example).Available:

https://youtu.be/4zahvcJ9glg

[19]. Cryptographic Hashing, Hackernoon.com. [Online]. Available: https://hackernoon.com/cryptographic-hashing-c25da23609c3

[20]. Introduction to Cryptography in Block chain technology, crushcrypto.com. Available: https://crushcrypto.com/cryptography-in-blockchain/

[21]. David Cardinal, Quantum Cryptography Demystified: How It Works in Plain Language,[Online] Available:https://www.extremetech.com/extreme/287094-quantum-cryptography

[22]. What are Cookie poising attacks, venafi.com. Available: https://www.venafi.com/blog/what-are-cookie-poisoing-attacks-0

[23]. Cryptology: Polygraphic Substitution, Staff.uni-mainz.de, Available: https://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/1_Monoalph/Polygraph.html

[24]. Quantum Key Distribution (QKD) - Quantum Technology,Available: https://qt.eu/understand/underlying-principles/quantum-key-distribution-qkd/