RESEARCH ARTICLE OPEN ACCESS

Data Security in Message Passing using Armstrong Number

Ajay Bansode[1], Amit Joshi[2], Awanish Singh[3], Kiran Gosavi[4],
Prasad S.Halgaonkar[5], Vijay M.Wadhai[6]
Department of Computer Science and Engineering,
MIT College of Engineering,
Pune-India

ABSTRACT
Data security while transferring data from one place to other is major issue in today's world. Data security mainly refers to protection of data from unintended user. This technique uses encryption and decryption at sender's and receiver side respectively. Especially this technique makes use of Armstrong number while encrypting and decrypting the data. This technique also makes use of Diffie-Hellman key exchange algorithm for exchanging key between sender and receiver. The proposed Algorithm is simple, flexible and making both hardware and software implementation easier.
Keywords- Armstrong numbers, data security, authentication, cryptography, cipher text.

I. INTRODUCTION

In today's world transferring data through unsecure network is major concern. To ensure the security of data while transferring through unsecure network, there are various kinds of technique used. One of the popular techniques used worldwide is cryptography. Cryptography involves converting plain text to some unreadable form. This unreadable form of data is then transmitted over the unsecured network. Cryptography mainly consist of encryption and decryption of the data. Encryption-Decryption is one of the techniques which is quite popular. But, the complexity which is involved in this technique doesn't allow its users to apply it in a simpler way. Now, if we look into the detailed context of this technique then we may observe that there are number of ways which allows the user to encrypt the private files and information. [1][2]

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. 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. [3]

In this paper, Encryption and Decryption process applies to both data as well as its key. So that two way security is provided to the application. After successful authentication, data is encrypted by random Armstrong number and at the same time Armstrong number gets encrypted. Now for both these encrypted data and key, current system timestamp is attached. So whenever receiver gets both the data he can easily recognize which key is for which data. Then encrypted key is decrypted by sender's public key and that resulted Armstrong number is used to decrypt actual data.

So it is difficult to hack the data and steal it. Once hacker steals the data, then he must have key by which that data is encrypted with its timestamp. If hackers get both data and key then he must know the decryption algorithm to retrieve both key and data which is very difficult.

II. CRYPTOGRAPHY

Cryptography is mainly a technique to keep communication private. Cryptography protects data from theft or alteration and also can be used for user authentication. Its main purpose is to ensure privacy by keeping the information hidden from anyone for whom it is not intended.

There are two main steps involved in cryptography such as encryption and decryption .The purpose of encryption is to ensure privacy by keeping the information hidden from anyone for whom it is not intended. Encryption is the transformation of plain text into some unreadable form. Decryption is the reverse of encryption, it is the transformation of encrypted data back into some readable form.

The data to be encrypted is called as plain text. The encrypted data obtained as a result of encryption process is called as cipher text.[4]

A. Types of Cryptographic Algorithms

There are many ways of classifying cryptographic algorithms. Generally they are classified on the basis of the number of keys that are used for encryption and decryption. The three types of algorithms as follows:

1. Secret Key Cryptography (SKC)

This kind of algorithm uses a single key for both encryption and decryption. The most common algorithms in use include Data Encryption Standard (DES), Advanced Encryption Standard (AES).

2. Public Key Cryptography (PKC)

This kind of algorithm uses one key for encryption and another for decryption. RSA (Rivest, Shamir, Adleman) algorithm is an example.

3. Hash Function

This kind of algorithm uses a mathematical transformation to irreversibly "encrypt" information. MD (Message Digest) Algorithm is an example.

III. DIFFIE-HELLMAN ALGORITHM

Diffie-Hellman key exchange algorithm is a cryptographic that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communication channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

For this discussion we will use Alice and Bob, to demonstrate the DH key exchange. The goal of this process is for Alice and Bob to be able to agree upon a shared secret that an intruder will not be able to determine. This shared secret is used by Alice and Bob to independently generate keys for symmetric encryption algorithms that will be used to encrypt the data stream between them. The "key" aspect is that neither the shared secret nor the encryption key do not ever travel over the network.[5]

TABLE I

Alice and Bob agree on two numbers "**p**" and "**g**"	"**p**" is a large prime number "**g**" is called the base or generator
Alice picks a secret number "**a**"	Alice's secret number = **a**
Bob picks a secret number "**b**"	Bob's secret number = **b**
Alice computes her public number **x** = g^a(mod p)	Alice's public number = **x**
Bob computes his public number **y** = g^b (mod p)	Bob's public number = **y**
Alice and Bob exchange their public numbers	Alice knows p, g, a, x, y Bob knows p, g, b, x, y
Alice computes **ka**= g^{ab}**(mod p)**	**ka** (g^b**mod p**)**a** mod **p** **ka**= (g^b)**a** mod **p** **ka**= g^{ab}**(mod p)**
Bob computes kb = g^{ab}**(mod p)**	**kb** = (g^a**mod p**)**b** mod **p** **kb** = (g^a)**b** mod **p** **kb** = g^{ab}**(mod p)**
Fortunately for Alice and Bob, by the laws of algebra, Alice's "**ka**" is the same as Bob's "**kb**", or **ka**= **kb** = **k**	Alice and Bob both know the secret value "**k**"

IV. SERVER ARCHITECTURE

Very general definition of a Server is a computer or device on a network that manages network resources. For example, a file server is a computer and storage device dedicated to storing files. Any user on the network can store files on the server. However on multiprocessing operating systems, a single computer can execute several programs at once. A server in this case could refer to the program that is to manage resources rather than the entire computer.[4]

A. What is Server Platform?

A server platform is basically a platform consist of underlying hardware or software for a system and thus is the engine that drives the server.

B. Types of server

1] FTP-Servers: One of the oldest of the Internet services, File Transfer Protocol makes it possible to move one or more files securely between computers while providing file security and organization as well as transfer control.

2] Mail-Servers: Almost as ubiquitous and crucial as Web servers, mail server move and store mail over corporate networks via LANs and WANs and across the Internet.

3] ***Print-server:*** It is a computer that manages one or more printers and a network server is a computer that manages network traffic. There are so many servers according to requirement like Audio/video, Chat, Fax, News, Proxy, Web servers etc.

V. PROPOSED SYSTEM

A. Introduction

In proposed system instead of keeping sender and receiver database as color and key value as that is used in earlier systems we keep unique number and name on single common server.

Before sending, data authentication is done between sender and receiver. Then after successful authentication we carry our encryption process of data and send that data to receiver. And encryption key as Armstrong number is send via server to receiver. After getting that key from server, receiver decrypt that key using sender's key and get original key. Using this key receiver decrypt encrypted data. [6]

Task of enhancing security include construction of formula for both data encryption and also for hiding pattern. Server should not process any fake request hence concept of "Diffie-Hellman key exchange algorithm is introduced. Implementation of such a security constraints in banking sector is widely helpful.

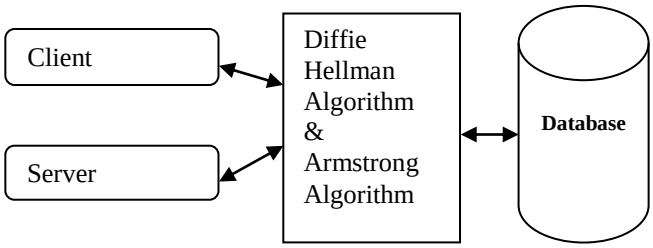

Fig. 1 Client-Server Architecture

B. Illustration

1. Encryption Using Armstrong Numbers**:**

Let the Armstrong number used for data encryption be 153.

Step 1: (Creating password)

User generates the key using Diffie-Hellman key exchange algorithm

Step 2: (Encryption of the actual data begins here)

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 3:

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 4:

Convert the above data into a matrix as follows

$$A=\begin{matrix}68 & 81 & 72 & 81\\ 87 & 109 & 207 & 77\\ 92 & 82 & 92 & 92\end{matrix}$$

Step 5:

Consider an encoding matrix

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

Step 6:

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

$$C=\begin{matrix}779 & 890 & 1383 & 742\\ 3071 & 3598 & 6075 & 2834\\ 13427 & 16082 & 28431 & 12190\end{matrix}$$

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.

2. Decryption Using Armstrong Numbers: Decryption involves the process of getting back the original data using decryption key.

Step 1: (Authenticating the receiver) only when the keys from sender and receiver match, the following steps could be performed to decrypt the original data.

Step 2: (Decryption of the original data begins here)

The inverse of the encoding matrix is

$$D=(1/240)\ *\begin{matrix}-450 & 240 & -30\\ -18 & 24 & -6\\ 100 & -120 & 2\end{matrix}$$

Step3:

Multiply the decoding matrix with the encrypted data (DXC) we get

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

Step 4:

Now transform the above result as given below

68 87 92 81 109 88 72 207 92 81 77 92

Step 5:

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 6:

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

C. Advantages

This minimum key length reduces the efforts taken to encrypt the data. The key length can be increased if needed, with increase in character length.

Tracing process becomes difficult with this technique, because the Armstrong number is used differently in each step. The key can be hacked only if the entire step involved in the encoding process is known earlier.

Simple encryption and decryption techniques may just involve encoding and decoding the actual data, but in this proposed technique the password itself is encoded for providing more security to the access of original data.

D. Disadvantages

Diffie-Hellman key exchange algorithm involves expensive exponential operations. The only way to break into this system is by Brute force attack, which also can take up to two or three years.

The speed of execution is slow because the file size after encryption is much larger than original file.

VI. CONCLUSION

The above combination of Diffie-hellman key exchange algorithm and Armstrong number proved to be the more efficient and reliable technique for data exchange between two parties. The combination of Diffie-hellman key exchange algorithm and encryption using Armstrong number provides two way securities. This technique provides more security with increase in key length of the Armstrong numbers. In this algorithm we use digital signature hence this algorithm defend against man-in-middle attack and provide more security.

ACKNOWLEDGMENT

We would like to thank Dr. P.Joeg, HOD (Computer Department) and would also like to thank Mr. Prasad Halgaonkar for his valuable support throughout the creation of this research paper.

REFERENCES

[1] S.Belose, M.Malekar, S.Dhamal, G.Dharmawat & N.J.Kulkarni, "Data Security Using Armstrong Numbers," Undergraduate Academic Research Journal (UARJ),ISSN : 2278 – 1129, Volume-1, Issue-1, 2012.

[2] Ajmal K.A, "Security using Colors, Figures and Images", International Conference on Emerging Technology Trends on Advanced Engineering Research (ICETT'12) Proceedings published by International Journal of Computer Applications (IJCA) 2012.

[3] S. A. Saoji, Nikita B. Agarwal, Mrunal B. Bokil, Ashwini V. Gosavi, "Securing e-mails in XML format using colors and Armstrong numbers" International Journal of Scientific & Engineering Research, ISSN 2229-5518, Volume 4, Issue 7, July-2013.

[4] Chavan Satish, Lokhande Yogesh, Shinde Pravin, Yewale Sandeep, Sardeshpande S.A, "Secure Email Using Colors and Armstrong Numbers over Web Services", International journal of research in computer engineering and information technology(IJRCEIT) , volume 1 no.2, 2013.

[5] RFC 2631 "Diffie–Hellman Key Agreement Method", E. Rescorla June 1999.

[6] S. Belose, M. Malekar, G. Dharmawat, "Data Security Using Armstrong Numbers" International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 4, April 2012.