AN IMPROVED DATA CONFIDENTIALITY PROTOCOL BASED ON TIMESTAMP

Amer Kais Obaid
Received: 11/5/2011
Accepted: 25/8/2011

Abstract

Security and data confidentiality has become a critical issue for companies and individuals. In a classical client-server network, the access control management is performed on the server, relying on the assumption that the server is a trusted party. In this paper, we focus on how to strengthen the data confidentiality in client-server networks. We address the improving data confidentiality protocol by using symmetric encryption based on different secret keys for each session derived from timestamp. This key is generated in a local machine and without needing to exchange it between the client and the server for encryption decryption process. We have to use symmetric encryption without exchanging the secret key to ensure the security of information from attacks for eavesdropping or modification.

***Key words**: Network security, Data confidentiality, Cryptography, and Timestamp.*

الخلاصة

أمن وسرية البيانات أصبحت مسألة حاسمة بالنسبة للشركات والأفراد. في شبكات خدمة العملاء المعتادة، يتم إدارة التحكم في الوصول للبيانات الموجودة لدى الخادم (Server) ، والاعتماد على افتراض أن الخادم (Server) هو طرف موثوق .

في هذا البحث سوف نركز على كيفية تعزيز سرية البيانات في شبكات (client-server). حيث سنعالج سرية البيانات باستخدام التشفير المعتمد على خوارزميات التشفير المتناظر(symmetric encryption) التي تعتمد على المفاتيح السرية المختلفة التي تتولد من (Timestamp). هذا المفتاح يتغير في كل جلسة اتصال التي ترسل إلى (Client) حيث تشفير البيانات. ان الفكرة الاساسية بأن المفتاح يتم خلقه (تكوينه) لدى كلٍ من الطرفين فلن يكون هنالك سبب لنقله بين (Client) و(Server).أستخدام التشفير المتناظر (symmetric encryption) ، دون الحاجة الى تبادل المفتاح السري (secret key) سيؤدي الى ضمان تأمين البيانات من الهجمات اوالتغير أو التعديل عليها خلال ارسالها عبر الانترنت.

*Control & Systems Eng. Dept., University of Technology

1. Introduction

Data passes between a client and a web server, sometimes through one or more intermediaries. Messages may also be kept in repositories. Some of the data within the messages is considered to be sensitive and may be attacked by adversaries. There is a risk that an attacker can gain access to sensitive data by eavesdropping on the network, and modify or use it in malicious activities [1, 2].

A confidentiality service provides protection to the sensitive data against unauthorized access or disclosure [3]. In this paper, we propose an improved data confidentiality protocol for exchanging the sensitive data between client and server using encryption based on timestamp to generate a secret key on local machines that will exchange only some information which is useful to generate the secret key.

2. Data Confidentiality

Data confidentiality is the protection of transmitted data from passive attacks, such as eavesdropping [4]. Threats to confidentiality include the direct release of sensitive data values, approximate disclosures, and leaks resulting from inferences and outside knowledge. One way to provide confidentiality is through cryptography [5].

3. Symmetric Cryptography

In modern security models, cryptography plays a fundamental role in protecting data integrity and confidentiality in information systems. There are two basic techniques in cryptography; symmetric and asymmetric cryptography [6, 7].

Symmetric key systems require both the sender and the recipient to have the same key. This key is used by the sender to encrypt the data, and again by the recipient to decrypt the data. Symmetric cryptography algorithms are typically fast and are suitable for processing large streams of data [8].

Cryptography using the same encrypting decrypting keys is called symmetric cryptography. This is illustrated in Figure 1.

4. Hash Function Cryptography

A hash function is a function that takes some message of any length as input and transforms it into a fixed-length output called a hash value, a message digest, a checksum, or a digital fingerprint [9].

4.1 MD5 Hash Algorithm

MD5 is one of the most widely used cryptographic hash functions nowadays. It takes an arbitrary message as input and generates a 128-bit output digest [10].

4.2 MD5 Hash Properties

The MD5 hash consists of a small amount of binary data, typically no more than 128 bits. The length of the hash value is determined by the type of the used algorithm, and its length does not depend on the size of the file. Every pair of non-identical files will translate into a completely different hash value. Each time a particular file is hashed using the same algorithm; the exact same hash value will be produced [11].

All hashing algorithms are one-way. Given a checksum value, it is infeasible to discover the password. In fact, none of the properties of the original message can be determined given the checksum value alone.

5. Related Works

Wenjing Lou, et.al. [4]: proposed a new idea to transform a secret message into multiple shares by secret sharing schemes and then deliver the shares via multiple independent paths to the destination so that even if a small number of nodes that are used to relay the message shares are compromised, the secret message as a whole is not compromised.

Dahui Hu and Zhiguo Du [7]: proposed the idea of applying fast RSA algorithm to improve Kerberos so as to meet the basic requirements, and then analyze the security and efficiency of the improved Kerberos.

6. The Proposed Protocol

We propose a new approach to improve of data confidentiality protocol using encrypted secret data with secret key derived from timestamp, this key is used to encrypt and decrypt the secret data. Because the symmetric cryptography presumes that two parties have agreed on a key and been able to exchange that key in a secure manner prior to communication. This is a significant challenge. We propose that that both client and server agree on a protocol to generate the secret key on local's machines with exchange only some information is useful to generate the secret key, as illustrated in section 7.

7. Implementation of proposed protocol

This section presents an improved protocol for data confidentiality.

7.1 Login and authentication phases

Figure 2: illustrates the login and authentication phase of this protocol.

$$C \rightarrow S: m_1 = \mathbf{MD5}[UN, PWD]. \quad (1)$$

After the client C requests a login to server's services, the later sends the authentication web-page. The client inserts his identification information (Username (UN) and Password (PWD)).

7.2 Encryption and Decryption phases

This information [UN, PWD] is encrypted with MD5 hash function to produce m_1, then sent to S.

$$S \rightarrow C : m_2 = [\, m_1, T_S] \quad (2)$$

After receiving message $m1$ from C, and verifying its identify, S generates m_2 derived from $m1$ and the timestamp T_S of client's login.

$$m_2 = m_1 + T_S.$$

Figure3: shows process of generating m_2.

Steps to generate m_2

1. Convert m_1 to ASCII code, then to Binary code.
2. Convert T_S to Binary code.
3. Add both binary values, neglecting the carry.
4. The result value is m_2.

$$C \rightarrow S : C_{txt} = [m_3, K_S]. \quad (3)$$

After receiving message m_2,the *Client* generates K_S (secret key) by encrypting m_2 *with* ***MD5*** *Hash Function.*

Where:

$$K_S = MD5\ [m_2]$$

This secret information m_3 is encrypted with K_S, to produce *Cipher text* C_{txt}, then sent to *Server*, without sending the secret key. Figure 4: Shows the process of generating K_S, and C_{txt}.

After receiving message $\boldsymbol{C_{txt}}$ from a client, the server decrypts it using Ks derived from m_2 which is stored in secret server's database, as the following:

$$K_S = MD5\ [m_2]$$

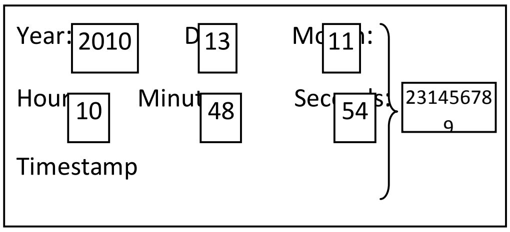
Figure 5 shows process of generating K_S, and C_{txt} .

7.3 Key Generation based on the Timestamp

A process is needed to generate a different set of random keys for each transmitter. The key is different for each client, and stored in temporary database.

The algorithm is to use a key generation based on timestamp as shown in the following procedure:

1. Take current time/date from client's computer.
2. Convert it to Timestamp.



3. As shown in figure 3, m_2 is generated from m_1 and T_S.
4. Take ***MD5 Hash Function*** for m_2 to produce K_S.

$K_S = MD5\ [m_2]$

1. $C\rightarrow S : m_1 = \boldsymbol{MD5}[UN , PWD]$:
   8081495505fc811ec36065c41c5c62ed

2. $S \rightarrow C : m_2 =[\ m_1, T_S]$

   Convert m_1 in Binary Code :

   1000000000010100100101010101000000101111111001000000100011110110000110110000000110010111000100000111000101110001100010111011011101 0.

   T_S: 231456789

   MD5 [T_S] :
   574b1a57efdfd71764e763800a6441de

   Convert T_S in Binary Code :

   0101011101001011000110100101011111101111110111010111000101110110010011100111011000111000000000000101001100100010000011101111011001

   $m_2 = m_1 + T_S$

   110101110100101100110100010010101101110011101010101011111000110101010111101010101110010010101111111011110110100011010101010101111

3. $K_S = MD5\ [m_2]$:
   3d37801a3d022842c0f9b3fa42afa886

8. Conclusion

Encryption data with symmetric cryptography is much faster than asymmetric cryptography, but secret key exchange is difficult because the exchange itself must be secure with no intervening compromise of the key. So this problem affects the confidentiality of the data. In this paper, we present an improvement of the data confidentiality protocol. The core idea of our improved protocol is based on encrypting the sensitive data using symmetric encryption without exchanging a secret (symmetric) key; where generating it in local machines, this reduces attacks of secret key. Thus, it would be difficult to detect data sent by the attackers. In future, we will implement the improved protocol practically and validate our conclusion. We believe that our improved protocol increases the network security as well as data confidentiality and integrity.

References

[1] Erik Tews, “Attacks on the WEP protocol”, Fachbereich Informatik TU Darmstadt, 2007.

[2] William Stallings, Network Security Essentials: Application and Standards, Prentice Hall, 1999.

[3] B. Schneier, Applied Cryptography, John Wiley & Sons, 1996.

[4] Wenjing Lou, Wei Liu, Yuguang Fang, "SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks", IEEE INFOCOM, 2004.

[5] Gerome Miklau, "Confidentiality and Integrity in Distributed Data Exchange", PhD Thesis, University of Washington, 2005.

[6] Huy Hoang Ngo, "Dynamic Key Cryptography and Applications", International Journal of Network Security, Vol.10, No.3, May 2010.

[7] Dahui Hu, Zhiguo Du, "An Improved Kerberos Protocol Based on Fast RSA Algorithm", IEEE, 2010.

[8] Amin D. Malayeri, Jalal Abdollahi, "Modern Symmetric Cryptography methodologies and its applications", Department of Computer Engineering, Malayer Azad University, Iran, 2009

[9] Joseph S. G. "Hash Function in Cryptography", MSc. Thesis, University Bregensis, 2008.

[10] Xiaoyun W. and Hongbo Y. "How to Break MD5 and Other Hash Functions",Shandong University, China, EUROCRYPT 2005, LNCS 3494, pp. 19–35, 2005.

[11] Ultimate MD5 Reverse, LuckySoft Company Ltd, *August 13, 2011.* http://md5decrypter.co/other.php.

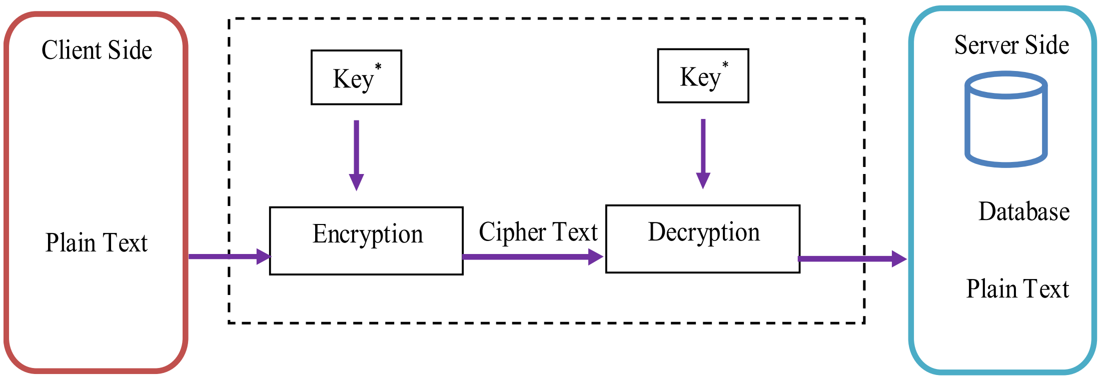


Figure 1: Symmetric cryptography

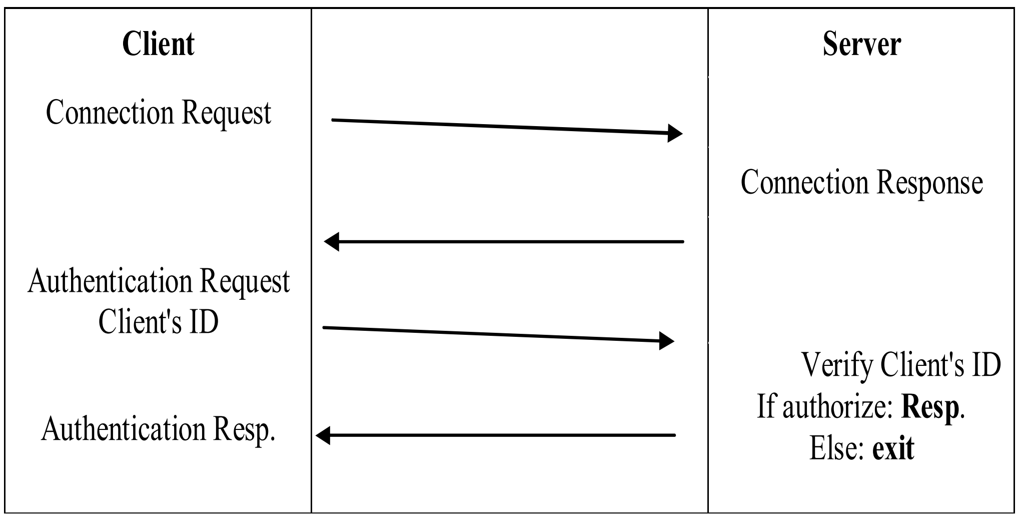


Figure 2: Login and Authentication Phases

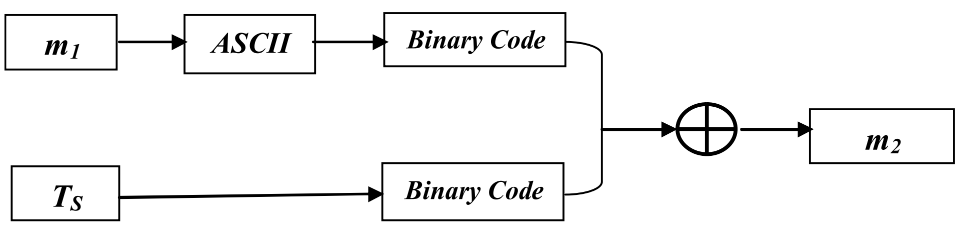


Figure 3: The process of generating m_2.

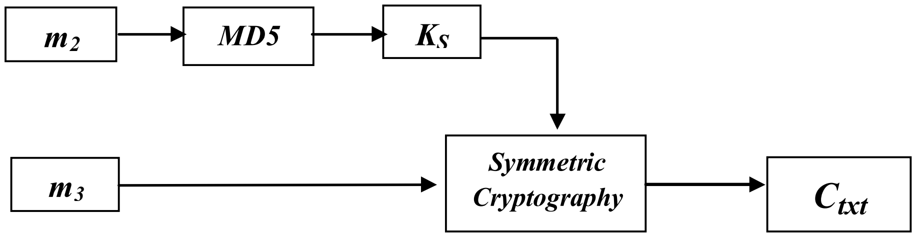


Figure 4: Process of generating K_S, and C_{txt} .

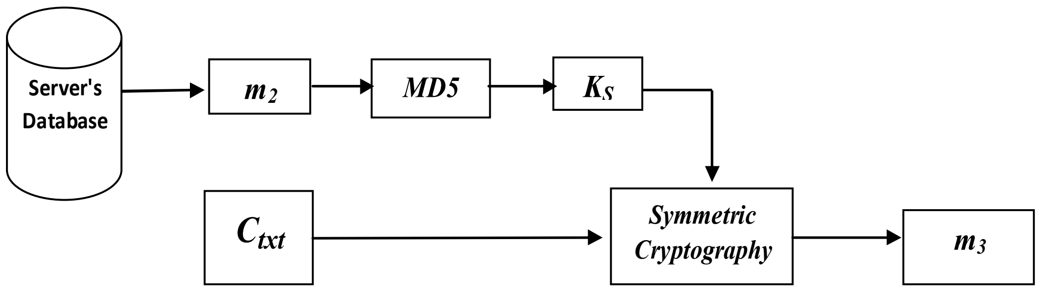


Figure 5: Decrypting Ciphertext to plaintext.