文章编号: 1671-5896(2005)03-0306-05

* 基于 Diffie-Hellman密钥交换的 Web安全传输

魏 达, 刘衍珩, 李晓东
(吉林大学 计算机科学与技术学院, 长春 130012)

摘要: 采用 SSL (Secure Sockets Layer) 安全传输协议实现 Web方式下的安全传输需要对整个传输通道的数据都进行加密处理, 因此其运算速度较慢, 对 Web服务器的性能有很大影响。为此提出了一种利用 DH (Diffie-Hellman) 密钥交换所需的参数并利用 DES (Data Encryption Standard) 分组密钥算法进行加密的安全传输过程。该方法以明文方式传送 DH密钥交换的参数, 客户端和服务器端利用这些参数计算共知的密钥, 并用该密钥对希望加密的数据进行 DES加密。不但实现了 Web方式下数据的安全传输, 而且因 DH和 DES算法有较快的速度以及只对希望加密的数据信息进行加密, 所以对 Web服务器的性能影响比采用 SSL下降了 70%以上。该方法已经在“JL金融保卫 MIS (Management Information System) 系统”中得到较好的应用效果。

关键词: 迪福 海尔曼; Web; 安全; 数据加密标准
中图分类号: TP309.7 **文献标识码**: A

Secure Web Transfer Base on Diffie-Hellman Key Exchange Algorithm

WEI Da, LIU Yan-heng, LI Xiao-dong
(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: The communicating information on Web must be encrypted by use of SSL (Secure Sockets Layer) protocol which is the common method of the secure transmission. This method is slow and deteriorates the performance of the Web server. A new method for securing transmission by use of DH (Diffie-Hellman) key exchange algorithm and DES (Data Encryption Standard) algorithm to transfer the DH parameters directly is presented. Web server and client compute the shared key and encrypt the essential communicating information, the secure transmission is insured and the influence of the new method compared with SSL on Web server performance come down 70%. The better effect is obtained by this method in Finance Security Management Information System.

Key words: Diffie-Hellman; Web; security; data encryption standard (DES)

引 言

随着 WWW应用领域的扩大, Web的安全问题日益受到重视。由于最初的 HTTP协议在设计时注重方便的交流, 并没有考虑安全问题, 基于 Web方式的传输都是以明文方式传送而没有任何加密手段。恶意攻击者可以利用软件或硬件设施很容易地得到数据包, 并从中提取明文方式的有用信息。目前对于 Web方式下的安全传输较多的解决办法是采用 SSL (Secure Sockets Layer)。SSL一旦启动, 则在 TCP (Transmission Control Protocol) 之上建立了一个加密通道。具有 SSL功能的浏览器与 Web服务器之间通信时, 利用数字证书确认身份并产生公共密钥进行通信。在这个通信过程中, 通过该层的所有数据都经过了加密[1]。在“JL金融保卫 MIS (Management Information System) 系统”的开发中, 由于存在部分

* 收稿日期: 2004-11-18
基金项目: 长春市科技局振兴东北老工业基地基金项目 (04-02GG158)
作者简介: 魏达 (1964—), 男, 吉林舒兰人, 吉林大学博士研究生, 副教授, 主要从事计算机网络管理和网络安全研究, (Tel) 86-431-5690186 (E-mail) weida@email.jlu.edu.cn; 刘衍珩 (1958—), 男, 吉林松原人, 吉林大学教授, 博士生导师, 主要从事计算机通信与网络研究, (Tel) 86-431-5168355 (E-mail) yhliu@email.jlu.edu.cn。

© 1994-2008 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

文件需要基于 Web方式上传和下发，这种上传下发过程以及登录过程需要一定的安全措施，并且不需要对所有的信息都进行加密，只需要加密部分敏感信息。对于该系统来说，全面部署 SSL的代价过高，笔者将 DH (Diffie-Hellman) 密钥交换过程与 DES (Data Encryption Standard) 加密方法相结合，提出了一种在不改变 Web传输基本架构的基础上对用户交互信息灵活加密的方法。该方法具有较高的安全性，对 Web服务器的性能影响极小。

1　Diffie-Hellman算法和 DES算法

Diffie-Hellman算法[2]是第 1个公开密钥算法，其安全性源于在有限域上计算离散对数，比计算指数更为困难。该算法可以使两个用户之间安全地交换一个密钥，但不能用于加密或解密信息[3]。首先，通信双方 A和 B协商一个大的素数 n和 g，g是模 n的本原元，这两个数不必是秘密的。故 A和 B可以通过不安全的途径协商它们，它们也可以在一组用户中公用。

在 Diffie-Hellman算法中，n和 g的选择对安全性有着极大的影响，最重要的是 n应该很大，$(n-1)/2$也应该是一个素数。这是因为系统的安全性取决于与 n同样长度的数的因子分解的难度。g则不必是素数，但它必须能产生一个大的 mod n的乘法组子群，并且任何模为 n的本原元 g都可以被选择。因此，可以选择满足条件的 g中最小者，以减少运算量，且不影响安全性。

1.1　协议过程

1) A选取一个大的随机整数 x并计算 $X = g^x \bmod n$，将 X发送给 B。

2) B也选取一个大的随机整数 y并计算机 $Y = g^y \bmod n$，将 Y发送给 A。

3) A计算 $k = Y^x \bmod n$。

4) B计算 $k' = X^y \bmod n$。

经过该过程的处理 k和 k都等于 $g^{xy} \bmod n$。由于线路上的窃听者所能知道的数据只有 n、g、X和 Y，所以除非他们能计算离散对数，以恢复 x、y，否则无法计算机出 k和 k的值。因此，k和 k可以作为通信双方各自独立计算，又相互知道的秘密密钥，其过程如图 1所示。

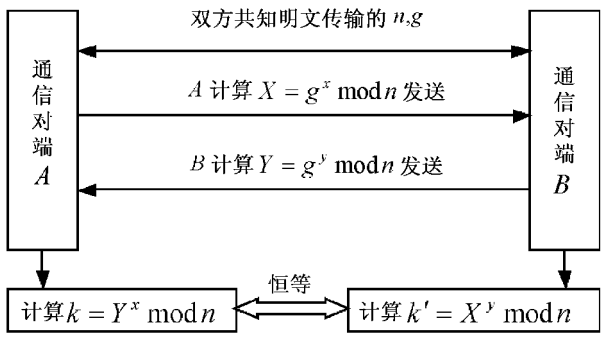

图 1　Diffie-Hellman密钥交换过程

Fig. 1 Diffie-Hellman key exchange

DES算法[4]是一个分组加密算法，它以 64位为分组对数据进行加密。DES算法的入口参数有 3个：Key, Data, Mode。其中 Key为 8个字节共 64位，是 DES算法的工作密钥；Data也为 8个字节 64位，是要被加密或被解密的数据；Mode为 DES的工作方式 (有两种：加密或解密)。

1.2　DES算法工作方式

1) 如 Mode为加密，则用 Key对数据 Data进行加密，生成 Data的密码形式 (64位)，作为 DES的输出结果。

2) 如 Mode为解密，则用 Key对密码形式的数据 Data解密，还原为 Data的明码形式 (64位)，作为 DES的输出结果。

在通信双方约定一致的 Key后，通过 DES加密、解密，保证了核心数据的安全性和可靠性。

由于 DES算法密钥长度的问题，虽然存在使用硬件设备对 DES算法进行穷举搜索攻击的可能性，但对于普通 Web应用中的口令和用户信息的保护，DES算法已经足够强壮了，具有较高的安全性。

2　用户交互信息的保护原理

随着基于 Web方式的应用逐渐增多，基于 Web方式的用户注册、登录等服务器端和客户行为的安全性日益得到人们的重视。Web服务以客户机 服务器模式运行，用户通过客户端的 Web浏览器向 Web服务器发出查询请求；Web服务器根据客户端请求的内容做出响应，并将某个页面发送给客户端浏览

© 1994-2008 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

器；客户端浏览器也可以提交表单给 Web服务器，Web服务器可以通过服务端脚本语言如 JSP (Java Server pages)，ASP (Active Server Pages)，ASP.NET或 CGI (Common Gateway Interface) 公共网关接口程序获得客户提交的信息并进行处理。无论服务器后台使用什么样的服务器端脚本，客户端浏览器得到的总是静态的 HTML (HyperTextMarkup Language) 语言格式的 Web页面[5]。因此，根据 Diffie-Hellman算法需要协商的参数，在客户得到 Web页时，同时也得到了服务器端的 Diffie-Hellman算法所需要的参数，这些参数的传输是以明文传输的 (见图 2)。

用户在客户键入用户信息之后，根据 Diffie-Hellman密钥交换算法，客户端脚本程序产生一个大的随机数字 x，并根据已经得到的 n，g，Y (其中 $Y = g^y \bmod n$由服务器端计算) 计算 $X = g^x \bmod n$和 $k = Y^x \bmod n$。然后利用秘密密钥 $k = Y^x \bmod n$，使用 DES算法对交互信息进行加密，并连同 $X = g^x \bmod n$一起提交给 Web服务器。其交互信息的加密过程如图 3所示。

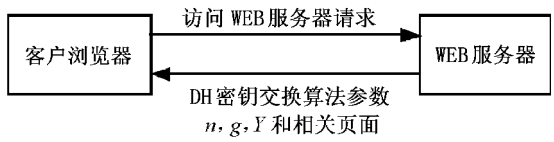

图 2 使用 Diffie-Hellman密钥交换的登陆过程

Fig. 2 Log in precessing of Diffie-Hellman key exchange

Web服务器得到 X和经过 DES加密的密文后，计算 $k' = X^y \bmod n$，并用 k对用户提交的经过 DES加密的信息进行解密。在该过程中 n，g，X和 Y是明文传输的，但其他任何用户交互信息都是经过以 k (k) 为密钥的 DES算法加密过的密文。

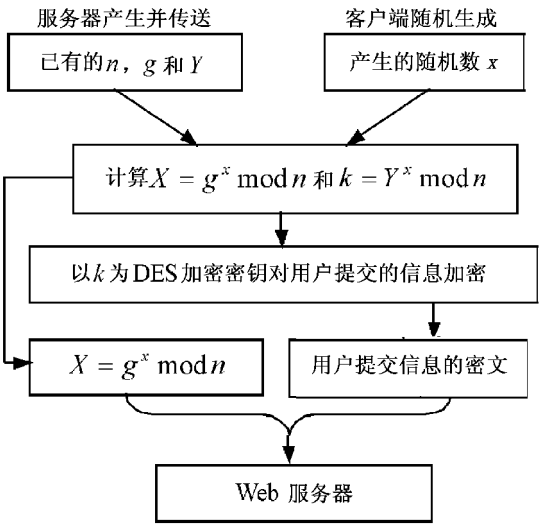

图 3 用户提交信息加密过程

Fig. 3 Ecrepting of the user information

3 密钥交换及加解密的实现

“JL金融保卫 MIS系统”是在 .NET架构下开发的 B/S (Browser/Server) 结构软件。在实现密钥交换及加解密过程中，密钥交换的参数可以选择合适的 n，g并存储在 Application对象[6]里，而不必每次进入页面的时候都重复生成。同时，给每个用户生成密钥交换的参数 Y并存放在 Session对象里。

利用 Javascript和 ASP.NET来实现基于 Diffie-Hellman密钥交换及 DES对称密钥的 Web信息的安全传输。客户端的 DES加密和客户端密钥交换参数 X的产生是通过用 Javascript语言写的 DES.js和 BigInt.js两个 js文件进行处理。DES.js是用来实现标准 DES加密的 js文件，BigInt.js是用来进行大数 (512至 1 024位) 运算的 js文件，并包含计算客户端公钥 $X = g^x \bmod n$的函数 create_X(var_g, var_n) 和计算公共密钥 $k = Y^x \bmod n$的函数 create_key(var_X, var_Y, var_n)。下面以用户登录过程为例，说明该加密过程的实现。页面名为 DHLogin.aspx，后置代码为 DHLogin.aspx.cs。

DHLogin.aspx页面部分：在页面部分放置输入用户名和密钥的 TextBox控件以及用来存放密钥交换参数的隐藏控件，同时在该页面包含 DES.js和 BigInt.js，并用 javascript语言写出相关加密函数，实现客户端的加密和 DH密钥交换的参数。

```
〈script language = "JavaScript"〉〉
var num_g = 〈% = Application["numG"].ToString() %〉
var num_n = 〈% = Application["numN"].ToString() %〉
var num_Y = 〈% = Session["numY"].ToString() %〉
funtion DHencrypt()
{
var num_X = create_X(num_g, num_n);
  //该函数通过密钥交换参数 n, g以及产生的随机数 x,
    计算参数 X = g^x mod n,并赋值给变量 num_X
var key = create_key(num_X, num_Y, num_n);
  //该函数通过密钥交换参数 X, Y, n,计算双方共知密钥
    k = Y^x mod n
```

© 1994-2008 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

```
var temp = document. all. username. value;
document. all. username. value = des_ escape( temp, key);
    //函数 des_ escape()使用 key对数据块进行 DES加密
temp = document all pwd value;
document all pwd value = des_ escape( temp, key);
document all numX. value = num_ X;
}
〈/script〉
```

DHLogin. aspx的后置代码 DHLogin. aspx cs部分:首先编写 Diffie-Hellman算法和 DES算法的 C#实现代码,然后在 DHLongin. aspx cs中直接调用。先定义全局变量 DiffieHellman diff;　DHParameters dhp;

```
private void Page_ Load(object sender, System. EventArgs e)
{
    //在此处放置用户代码以初始化页面
if (! Page. IsPostBack)
{
    //密钥交换参数的生成并存储过程
    diff = new DiffieHellmanManaged();
    dhp = diff. ExportParameters(false);
    byte[ ] key_ Y = diff. CreateKeyExchange();
    if(Application [ "numG" ]! = null) Application. Add
    ("numG", dhp. G);
    if(Application [ "numN" ]! = null) Application. Add
    ("numN", dhp. N);
    if(Session[ "numN" ]! = null) Session. Add( "numY",
    numY);
    //为 Web控件 submit按钮注册点击时进行处理的
      JavaScript函数 submit Attributes. Add( "onclick",
      " DHencrypt()");
    }
}
    //当 submit控件点击后服务器端的处理函数
private void submit_ Click(object sender, System. EventArgs e)
{
    //得到客户端和服务器端共知的密钥
    byte[ ] key = dhp. DecryptKeyExchange(Convert. ToByte
    (numX. Text));
    //通过上述密钥进行 DES解密得到客户端加密的用户
      信息
    string username = des_ unescape(username. Text, Convert. To-
    String(key));
    string password = des_ unescape(pwd. Text, Convert. ToString
    (key));
}
```

在本实现过程中的 Diffie-Hellman算法和 DES算法都分别用 JavaScript和 C#实现,并分别用于客户端和服务器端的加密解密过程。

4　性能分析

笔者提出的这种安全传输方案仅对任何欲加密的信息进行加密处理,因此,除了加密页面的请求和下载的传输需要服务器的额外计算外,其他任何页面的处理都不增加服务器的负担,并且不需要公钥证书的存在。而 SSL方式则对每个 Web请求都需要进行通信信息的加解密过程,尽管初始认证和密钥生成对于用户是透明的,但对于 Web服务器来说,它们远非透明,因而给服务器 CPU造成了沉重负担并产生了严重的性能瓶颈,大大增加了 Web服务器的负担。标准的 Web服务器在处理 SSL相关任务时,性能比只处理 HTTP1.0连接时的速度慢 50倍。虽然通过安装 SSL加速器和卸载器可以减少 SSL处理中的时延,但价格昂贵[7]。

SSL安全传输过程的握手协议包括多种密钥组合,现将笔者提出的方案与 SSL方式下的不同密钥算法组合在性能上作如下分析。

1) 当 SSL采用 RSA算法进行加解密过程并且在进行握手过程时已经取得了公钥证书,则在 10 MB以太网、Pentium 90客户机和 SUN-2SPARC10服务器的测试环境下,得到的 RSA算法与 DH算法握手时间如表 1所示。

若采用 RSA算法进行握手前未取得公钥证书,则需要更长的时间。因此,尽管在安全方面上 DH比 RSA密钥交换脆弱,但该算法简洁,运算速度快,适用于需要经常建立连接的通信业务[8]。

表 1　不同算法握手时间

Tab. 1 Handshake time of different algorithm　　s

DH算法 * *	DH算法 *	RSA算法 *
9.6	3.0	8.9

注: * 已经取得公钥证书;
* * 无公钥证书,实时生成公钥证书

2) 当 SSL采用 DH算法进行握手过程并实时生成证书时,同样在 Pentium90客户机和 SUN-

© 1994-2008 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

2SPARC10服务器环境下,使用 Diffie-Hellman密钥交换协议,服务器和客户实时生成公钥参数 g^x, g^y 和 g^{xy},握手时间为 9.6 s[9]。而采用本方案进行密钥交换并加密的过程并不需要证书的参与,其处理时间还是 3 s。

因此,这种通过 Diffie-Hellman算法进行密钥交换然后使用 DES加密的处理过程快速,灵活,且具有较好的安全性。

5 结 语

此外,在本方案中由于客户端和服务器端的密码在每次交互中都是随机生成的,因此也很好地解决了重放攻击。笔者提出的这种基于 Diffie-Hellman密钥交换的 Web安全传输,解决了“JL金融保卫 MIS系统”中安全方面的要求,并且在实践中得到较好的应用效果。由于 Web服务是在一个公共平台上提供服务的,所以面临着更多的安全问题。目前,针对 Web服务器的攻击方式层出不穷,Web安全的研究还有很多工作要做。

参考文献:

[1] RELER, KARLTON KOCHER. The SSL Protocol Version 3.0 (draft-ietf-tls-ssl-version3-00. txt) [EB/OL]. http://www.ietf.org, 1996-10.

[2] DIFFIE W, HELLMAN M. New Directions in Cryptography [J]. IEEE Transaction on Information Theory, 1976, IT-22 (6): 644—654.

[3] 孔晖,郑志华,徐秋亮 (KONG Hui, ZHENG Zhi-hua, XU Qiu-liang). 几种典型的认证 Diffei-Hellman型密码共识协议的分析与比较 (Analysis to Several Typical Authenticated Diffie-Heiiman Key Agreement Protocols) [J]. 计算机工程与应用 (Computer Engineering and Applications) 2001, 37 (18): 72—74.

[4] (USA) BRUCE SCHNEIER. 应用密码学:协议、算法与 C源程序 (Applied Cryptography: Protocols, Algorithms and Source Cord in C) [M]. 北京:机械工业出版社 (Beijing: China Machine Press), 2000.

[5] 杨千里 (YANG Qian-li). 电子商务技术与应用 (Electronic Commerce Technology and Applications) [M]. 北京:电子工业出版社 (Beijing: China Publishing House of Electronics Industry), 1999.

[6] 王超 (WANG Chao). ASP NET XML深入编程技术 (Advanced Programming Technology of ASP. NET XML) [M] 北京:北京希望电子出版社 (Beijing: Beijing Hope Electronic Press), 2002.

[7] 徐炳康 (XU Bing-kang). 用 SSL安全协议实现 Web服务器的安全性 (Implementation of Secure Web Server with SSL) [J/OL]. 计算机世界 (China Computerworld), http://news chinabyte com/ServerIndex/77132944006709248/20040508/1794846. shtml, 2004-05-10.

[8] 韦卫,王行刚 (WEI Wei, WANG Xing-gang). 密钥交换理论与算法研究 (Researches on Key Agreement Theory and Algorithms) [J]. 通信学报 (Journal of China Institute of Communication), 1999, 20 (7): 64—68.

[9] 韦卫,王德杰,张英,等 (WEI Wei, WANG De-jie, ZHANG Ying, et al). 基于 SSL的安全 WWW系统的研究与实现 (Study and Implementation of A Secure World Wide Web System Based on Security Socket Layer) [J]. 计算机研究与发展 (Journal of Computer Research and Development), 1999, 36 (5): 619—624.

(Ed.: H, T)

© 1994-2008 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net