基于 J2ME 和混沌加密的安全移动商务方案*)

吴中堂　冯久超
(华南理工大学电子与信息学院　广州 510641)

摘　要　本文提出了一个端到端的安全移动商务方案,它是基于 J2ME 并采用密钥动态更新的快速混沌加密技术而实现的。针对混沌序列的特点,本文提出密钥生成与加密过程分离;对软件实现的有限字长效应,采用了两次反馈混沌加密。一个实例证实了本文的方案。
关键词　J2ME,混沌加密,移动商务,混沌序列

A Secure Mobile Commerce Scheme Based on J2ME and Chaos Encryption

WU Zhong-Tang　FENG Jiu-Chao
(School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641)

Abstract　An end-to-end secure scheme for mobile commerce is proposed in this paper. It is designed and realized based on J2ME and fast chaos encryption with dynamical key. According to the character of chaotic sequence, the key is separated from the process of encryption. Because the effect of the limited word-length always exists, the user data is encrypted two times by feedback. An example is demonstrated.
Keywords　J2ME, Chaos encryption, Mobile commerce, Chaotic sequence

1　引言

移动通信业务的普及使得原来基于有线网络的商务逐步向基于无线网络扩展。手机、个人数字助理(PDA)等移动终端的功能越来越强大,已逐步满足人们随时随地进行商务活动的需要。虽然移动终端厂商采用了不同的芯片和操作系统,但绝大多数都支持 Java,故 Java 成为开发移动软件的首选语言。同时,采用 J2ME(Java 2 Micro-Edition)和 J2EE(Java 2 Enterprise Edition)开发工具,可以开发出纯 Java 的移动商务平台。

移动商务对安全性提出了新的挑战,需要保证数据的安全性、完整性、通信双方可辨别性及不可抵赖性[1,2]。本文主要解决从移动终端到网络服务器端的信息安全传输问题。根据混沌序列的特点和有限字长效应,采用了混沌序列的生成与加密过程的分离,实行两次反馈加密,并对密钥进行集中管理和动态更新。

2　系统结构

2.1　系统结构模型

图 1 显示了本文提出的包含客户端与服务器端的系统框图。客户端利用密钥对(key1,key2)对发送的信息进行加密,将信息分成一定的长度,然后与 key1、key2 做反馈型异或运算。即从第二字节始,前一字节的密文与密钥异或后,再与后一字节的明文加密(如图 2)。经过两次加密后得到的密文通过 HTTPS(基于 KSSL 的 HTTP)协议发送到服务器端,服务器利用它预存的客户端初始密码,对接收的信息进行解密并做进一步的处理。同时,服务器也建立用户注册信息,并从密钥生成器中获取新的密钥(key3,key4),存于用户信息表中。然后将经过 key1、key2 加密后的新密钥传送到客户端。客户端获得密钥后,自动更新密钥。服务器用新密钥对客户需要的信息加密后通过 HTTPS 协议传送,最后由客户端解密并获得需求信息。

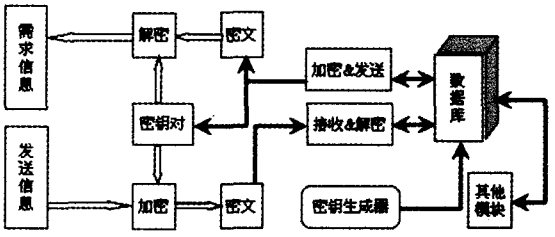

图 1　含客户端和服务器端的交互模型

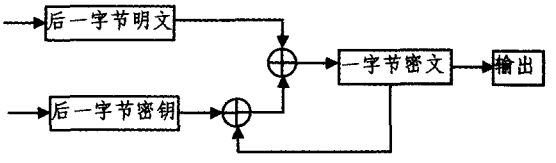

图 2　加密流图

2.2　客户端和服务器端组件描述

以电话会议系统为例,客户端遵守 MIDP2.0 规范,兼容 MIDP1.0。客户端的重要应用类包括 MeettingMIDlet、ChaosEncrypt、RegisterManage、Communicator 以及它的配套类。MainMIDlet,RegisterManage 类实现与用户的交互;ChaosEncrypt 类实现数据信息的加密和解密;Communicator 类实现数据的传送、接收和信息解析。服务器端组件用 J2EE 开发,并用 servlet 与客户端交互信息;其它模块包括系统维护模块、电话拨打处理模块等。服务器端的主要功能是:对客户端信息加密、解密、解析、存储、更改帐户信息、拨打电话和传递信息等。

3　密钥生成、密钥管理

3.1　混沌加密

混沌系统是一种确定性的类随机系统,它对初始条件及

*)国家自然科学基金(60572025)、教育部基金("新世纪优秀人才"基金:NCET-04-0813,重点项目:105137、广东省自然科学基金团队研究项目:04205783)资助。吴中堂　电路与系统专业硕士研究生;冯久超　教授,博士生导师。

系统参数的变化极端敏感[3]。它生成的混沌序列具有非周期性、类随机性并在分布上不符合概率统计学原理，因此混沌系统具有短期不可预测性[4]。混沌系统显示的这些特性与密码学的许多要求，如对密钥的敏感性、对明文的敏感性以及密文的随机性和发散性是一致的[5]。

3.2 密钥的生成

3.2.1 由服务器端生成密钥的优点

在本文中，密钥是指在服务器端生成的用来加密的混沌序列，由 Logistic 混沌系统和一个经过改进的分段混沌映射构成的系统所产生。当赋给混沌系统一定的参数值和初始值，经过大量的迭代后，我们从中选取需要的序列。将它们用在服务器端时有如下优点：

(1) 提高加密效率。由于 MIDP 1.0 不支持浮点型运算，而我们的混沌序列是浮点型数据且运算量大。虽然可以采用一些编程技巧进行浮点数运算，但是冗长的迭代与转换加大了系统的负担，从而延长了加密时间。MIDP2.0 支持浮点型数据运算，但对于一般手机来讲，做大量的浮点型数据运算需要用户在加密数据时等待更长的时间。

(2) 利于密钥生成器更新。有利于采用其它混沌系统生成混沌序列，如 Lorenz 混沌系统[6]，同时生成混沌序列的复杂度与终端加密速度无关。

(3) 实现密钥的动态更新。将密钥在服务器端集中管理，每次链接时服务器发送给客户端新的密钥，加大了第三方破解密文的难度；根据需要，服务器也可以在一次通话中多次更新密钥，这只需要改动服务器端的逻辑层的 servlet 。

3.2.2 两种混沌映射的分析和使用

Logistic 映射是定义在(0,1]上的一维混沌映射，其动力学方程是：

$$x_{n+1}=ax_n(1-x_n) \quad (1)$$

其中 a 是系统参数($a\in(0,4]$)，x_0 是序列的初始值。当 $a\in[3.7,4]$时，该映射是混沌系统。理论上，Logistic 映射生成的混沌序列具有非周期性和对初值的敏感性，但由于受计算机字长的限制，无论采用何种精度的浮点运算，生成序列的状态数目都是有限的。下面是另一个具有良好统计特性的一维分段映射：

$$y_{n+1}=f(y_n)=\begin{cases} y_n/p, & 0\leqslant y_n<p \\ (y_n-p)/(0.5-p), & p\leqslant y_n<0.5 \\ f(1-y_n), & y_n\geqslant 0.5 \end{cases} \quad (2)$$

其中 $y_n\in[0,1]$，$p\in(0,0.5]$。该系统是混沌的且生成的序列具有良好的自相关性和均匀分布特性。但直接利用此映射迭代得到的混沌序列作为密钥是不可取的，因为该映射的参数 p 容易破解，同时迭代可能陷入周期循环[7]。为此，我们采用以下方法解决上述问题。

(1)n 阶 m 序列：对第二个映射加入 n 阶 m 序列，这种随机扰动改变了混沌序列的性能，同时提高了序列的随机性和复杂性[8]。

(2) 两次加密：分别采用上述两个混沌映射生成的序列对数据进行加密。

(3) 反馈加密：通过反馈加密，使得每一分组中的后一字节密文与前面所有的密文相关。

3.2.3 密钥生成器

我们以 Logistic 映射为例，说明混沌序列的获取过程。对该系统，我们取参数 $a=3.999999999$ 和初值 $=0.3333333333$(均取十位有效数字)。将它们代入式(1)并迭代 5000 次后，从生成的序列中随机地选取 16 个双精度小数，在乘以 10^8 并取整后，用 128 取余，由此得到的序列作为加密序列。可以看出，这样获得的序列需要经过大量的浮点数运算，这对于资源受限的手持设备来说是不小的开销。

3.3 密钥管理

客户端的软件系统内置了由密钥生成器产生的密钥对(作为初始密钥)。用户注册后，在服务器上就建立了用户信息表，并能从密钥生成器中获取新的密钥对，用户在每次链接时发送新的密钥。这样，密钥的更新不会给客户端造成任何资源负担，这对像手机 PDA 等手持设备是非常重要的。

4 应用实例——手机电话会议系统

4.1 客户端功能简介

我们在客户端采用了 MVC 设计模式[9]，实现用户注册、帐户信息管理、预定电话会议和召开电话会议等功能。用户注册后，从输入界面输入相关信息(如图 3)，系统自动加密后发送到服务器；当用户开通服务后，客户端会显示服务器发来的用户帐户信息。

4.2 客户端加密、解密

加密过程由 ChoasEncrypt 类完成。用户输入信息后，点击 Encrypt 按钮。系统程序先将输入信息集合成一个字符串，然后进行反馈加密。下面是加密算法的代码简述：

```
public String encrypt(String plainText, byte[ ]keyCode ){
int Ntext = palinText. length (); //明文长度
int N=Ntext/16;
int flag=Ntext %16 ;
if(flag= =0){ //明文长度是密钥长度的整数倍时，分为 N 组
for(i=0;i<16;i++){
if(i= =0){ charString [i]=(char)(str. charAt(i) ∧ keyCode [i]);
}
else //反馈型异或运算
charString [i]=(char)(str. charAt(i) ∧ (charString [i ? 以1] ∧
keyCode [i])); } }
```

当明文的长度不能被密钥的长度整除时，仍然进行上述的类似处理。反馈运算的解密算法如下：

```
charString [i]=(char)(plainText. charAt(i) ∧ (plainText.
    charAt(i − 1) ∧ keyCode [i]));
```

对上述图 3 中输入的数据进行加密后，得到的密文如图 4 所示。

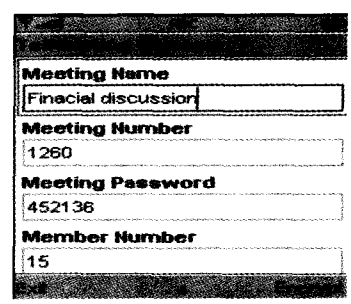

图 3 用户输入示例

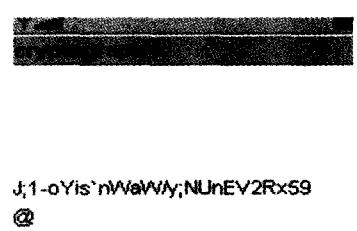

图 4 对应的密文

4.3 服务器端

基于J2EE,我们开发了服务器端的系统。当服务器端的逻辑层servlet接收用户数据后,从数据库中查询出用户的密码,然后用上述的解密算法得到明文。最后,根据标志符进行分类处理,如存储用户的会议密码、人数、旁听密码等等。

5 系统安全性

由于密钥生成器的参数(即公式(1),(2)中的a、x_0、p、y_0)的值在给定的区间内有较大的取值范围,故密钥空间较大;基于KSSL的HTTP协议,保证了数据的完整性。由于采用了反馈加密,使得密文具有良好的扩散性。该方案具有抗穷举攻击能力,这是因为当进行密码分析时,只有获得了上述4个参数才能得到正确的密钥生成器的信息。加之混沌序列的选取是随机的,若用直接的穷举攻击,在理论上是行不通的(因为遍历16个序列值需要验证次数为128^{16},数量级为10^{32})。该方案也具有抗选择明文攻击性:混沌序列中的x_n与x_{n+1},y_n与y_{n+1}之间不是直接的迭代关系,它们是在一个较大的范围内随机地选取的,因而攻击者无法通过简单的运算得出混沌系统的参数[10]。

结论 本文提出了一个将混沌加密与HTTPS协议相结合并应用于移动商务的安全方案。它利用了混沌系统的初值和系统参数的秘密性、序列值选取的随机性,并采用了密钥集中管理、动态更新。算法速度快,加密效果好,能抵抗混沌密码系统的相空间重构攻击、统计分析攻击。该方案适用于客户端/服务器(C/S)模式。

参 考 文 献

1 Park Nam-Je, Song You-Jin. M-Commerce security platform based on WTLS and J2ME. In: Proc. of 2001 IEEE International Symposium on Industrial Electronics, Pusan, Korea, 2001. 1775～1780
2 朱从旭,陈志刚. 一种适于移动计算的快速组合混沌加密方法. 计算机工程, 2005, 31(1): 138～140
3 Lorenz E N. Deterministic nonperiodic flow. J. Atmospheric Sciences, 1963, 20(2): 130～141
4 Eckmann J P, Ruelle D. Ergodic theory of chaos and strange attractors. Rev Modern Phys, 1985, 57: 617～656
5 Jakimoski G, Kocarev L. Chaos and cryptography: block encryption ciphers based on chaostic maps. IEEE Transactions on Circuits and Systems-I, 2001, 48(2): 163～169
6 Lenz H Obradovic D. Global Control of Lorenz Chaos. Proc. of IEEE Conference on Decision and Control, 1977, 2: 1486～1487
7 崔光亮,冯正进,胡国杰,等. 基于参数跳动和扰动的混沌保密系统的理论设计. 上海交通大学学报, 2004, 38(11): 1818～1821
8 Staff M. Introduction of MVC structure in J2ME client (see website: http://www.motorola.com)
9 周红. 有限精度混沌系统的m序列扰动实现. 电子学报, 1997, 25(7): 95～97
10 孙克辉,张泰山. 基于混沌序列的数据加密算法设计与实现. 小型微型计算机系统, 2004, 25(7): 1368～1371

(上接第81页)

相应服务率,然后按服务率分布存储于不同的磁盘(组)和不同的服务器,提高系统的并发处理能力。关于服务率的确定,虽然可以依据(5)式得到,但在不同的应用场合用户可能有不同的偏好,因此对于相应排列我们可以采用基于统计的方式获取。每次重分布的统计我们引入了一个依赖因子λ,并按照依赖因子重新计算所有流节目的服务率及其排列,如(7)式所示:

$$f(i)=\lambda f(i-1)+(1-\lambda)s(i) \quad i=1,2,\cdots,N \qquad (7)$$

其中,i是节目编号,S是本次统计的服务率。依赖因子的引入平衡了每次统计,避免重新分布的过度抖动。

4 测试与分析

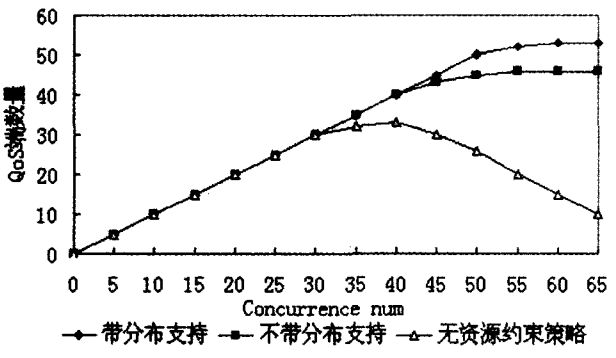

图7 策略控制比较

为了验证上文所述策略的有效性,采用了两台浪潮英信Np350服务器,每台使用promise Ultra133加速阵列分别连接4个IBM80G硬盘,1000个视频流文件为mpeg2格式,DVD质量,平均码流基本相同。其中100个为影视文件,其他为短节目(≤10min)。长节目按(6)式分割,其中t_0=10min,t_1=60min,E取平均值0.4,T=0.6,依赖因子取0.3。按(5)式赋初始服务率值给各个流文件再按概率平均存放在2台服务器8个硬盘,然后按20～80规则,访问相应流节目。实验结果如图7所示。

在无资源约束策略下,系统是尽可能为所有连接提供服务,在连接数增多的情况下,客户端的QoS将不能保证。在有约束策略的支持下,QoS端数量将在达到一定水平后保持稳定,此时多出的并发请求将被拒绝接入并进入队列管理。测试同时也显示使用服务率分布支持系统能负荷的并发数有所提高。实验证明策略对保证客户端表现质量有效,并能较好利用系统资源支持尽量多的并发。

结束语 在大规模的高质流媒体服务系统中,流应用对QoS更为敏感,系统的关键资源也是系统性能的瓶颈,如何更有效地分配和使用这些资源,是保证QoS的关键。本文从资源约束的角度出发,针对系统瓶颈提出一套流媒体服务系统QoS保障的策略,并通过测试验证了策略的有效性。在基于统计的服务率排列计算中,重分布对系统性能也是有一定的影响,这也是下一步的研究方向。

参 考 文 献

1 Li X, Ammar M. Bandwidth Control for Replicated-stream Multicast Video Distribution [C]. In: Proc. of the Fifth IEEE Intl Symposium on High Performance Distributed Computing, Syracuse, NY, 1996, 8: 45～56
2 Wolf J L, Yu P S, Shachnai H. Disk Load Balancing for Video-on-Demand Systems [J]. Multimedia Systems, 1997, 5: 358～370
3 Flynn R, Tetzlaff W. Disk striping and block replication algorithms for video file servers [A]. Multimedia Computing and Systems. In: Proc. 3rd IEEE Int Conf [C]. IEEE Press, 1996. 590～597
4 Wu Song, Jin Hai. Symmetrical pair scheme: a load balancing strategy to solve intra-movie skewness for parallel video servers [A]. In: Parallel and Distributed Proc Symposium, Proc. Int [C], IEEE Press, 2002. 126～132
5 Laoutaris N, Zissimopoulos V, Stavrakakis I. On the optimization of storage capacity allocation for content distribution. [J]. Computer Networks, 2005, 47(2): 409～428
6 李宇辉,李吉桂. 多媒体流间同步参数计算[J]. 计算机科学, 2001, 28(8): 50～52
7 钟玉琢,向哲,沈洪. 流媒体和视频服务器[M]. 清华大学出版社, 2003