基于 AES 加密算法的数据库二级密钥技术①

苗 杰, 钱 强, 高 尚

(江苏科技大学 计算机科学与工程学院, 镇江 212003)

摘 要: 为了提高用于数据库加密的加密密钥的生成和存储的安全性, 本系统采用一种新的二级密钥体制, 利用 AES 算法对主密钥进行一系列变换, 在变换中产生工作密钥, 而不需要提前设定工作密钥. 为了进一步保证数据库加密密钥的安全和生成速率, 在分析 AES 算法密钥矩阵的构造原理的基础上对其进行改进, 引入在密钥扩展算法中被称之为“单向性”的设计策略, 来减轻各轮子密钥之间的关联性, 为数据库的安全提供了更有力的保障.

关键词: AES 算法; 密钥扩展; 二级密钥; 主密钥; 工作密钥

Technology About Double Encryption Key of Database Security Based on AES Encryption Algorithm

MIAO Jie, QIAN Qiang, GAO Shang

(School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: To ensure the security of the generation and storage of encryption key, this system proposes one new double encryption-key mechanism for database encryption, in which the AES encryption is used to get work key from transform of the main key, without setting work key firstly. In order to further ensure the security of database encryption key and to improve the encryption key generation rate, the construction of AES algorithm key matrix is analyzed and improved. The key expansion algorithm called ‘one-way’ design strategy is introduced to reduce the correlation between each wheel key, which will help to provide a more effective guarantee for the security of database.

Key words: AES encryption algorithm; key expansion; double encryption-key; main key; work key

在当今, 人们与各种各样的信息打交道, 一些重要信息都是以数据库的形式来保存, 这样数据库中的一些机密信息的泄漏, 将会为个人、公司、甚至是国家带来难以预料的损失. 为了确保这些数据不被窃取, 最有效的方法就是对数据库进行加密. 所谓数据库加密[1]就是将数据库中的数据(明文)经过一些加密算法变换成不可直接识别的数据形式(密文).

相对于传统的数据加密, 数据库加密有其自身的特点和要求. 数据库的加密粒度可分为基于表、记录、字段. 通过不同的加密级别的平台实验, 得出字段级数据库加密的综合性能最好. 而且由于数据库的数据量比较大, 为了保证系统性能, 就要求加/解密速度一定要快, 所以数据库的加密一般选用对称加密算法. 另外数据库中的信息是长期存放的, 数据的加密密钥的一些信息也必须存放在系统中. 这样就会造成一些特定用户可以获取密钥, 使系统的安全面临危险. 解决这样的方法之一是采用二级密钥体制, 即主密钥用来完成对工作密钥的加密, 而工作密钥用来完成对数据库数据的加/解密, 这样一来工作密钥的存储和通信又成为了一个数据库安全的新问题.

目前对数据库加密的对称算法主要有 DES、AES, 由于 DES 算法的密钥只有 56 位, 且所有的 S-盒都是固定的, 随着计算机计算能力提升, 只用了 20 年, 使用穷举法就完全破解了 DES 的全部密钥. AES 算法作为新一代的高级数据加密标准, 相对于 DES, AES 的密钥长度更长, 实现起来也更简单, 抗分析攻击能力也得到增强. 基于此, 本文采用 AES 加密算法来实现数据库加密的二级密钥体制[2,3], 并对其进行改进, 利

① 基金项目:“青蓝工程”(苏教师(2010)27 号)

收稿时间:2012-08-28;收到修改稿时间:2012-09-21

用主密钥来产生工作密钥, 从而避免了工作密钥的存储和通信问题.

1 AES 算法原理[4,5]

AES 算法是分组密码体制, 其数据分组长度为 128 位, 密钥长度可为 128 位、192 位或 256 位, 由密钥长度决定的轮变换次数 *Nr* 为 10、12 或 14. AES 算法的数据处理的最小单元是字节, 128 比特的分组信息被分成 16 个字节, 并按顺序被复制到一个 4*4 的矩阵里, 称为状态. AES 的所有变换都是基于状态的变换, AES 对数据的加密是通过把输入的明文和密钥由轮函数经过 *Nr*+1 轮迭代实现. 轮函数的每一轮迭代包含 4 步变换, 分别是字节替换 ByteSub()、行位移变换 ShiftRows()、列混合变换 MixColumns()以及轮密钥的添加变换 AddRoundKey(). 具体操作分别为: AddRoundKey 使用从种子密钥值中生成的轮密钥代替 4 组字节. SubBytes 替换用一个代替表替换单个字节. ShiftRows 通过旋转 4 字节行的 4 组字节进行序列置换. MixColumns 用域加和域乘的组合来替换字节. 不失一般性, 本文选取密钥长度为 128 位, AES 具体的加密与解密过程如图 1 所示:

图 1　AES 的加密与解密流程

10 轮算法的前 9 轮结构相同, 除第 10 轮由字节变换、行位移变换、轮密钥加 3 步组成外, 其它 9 轮都是由字节变换、行位移变换、列混合变换、轮密钥加 4 部组成. 第 1 轮的开始之前首先要进行 1 次密钥加运算.

2 二级密钥的生成与管理

2.1 主密钥 MK

为了保证主密钥 MK 随机性, 采用投币法随机产生用于 AES 加密算法的 128 位原始密钥作为一级主密钥 MK. 这样, 利用主密钥 MK 通过调用 AES 算法密钥扩展可以产生 AES 算法中的 10 组轮密钥 *k*1、*k*2…*k*10.

由于主密钥是进入数据库系统的第一道门户, 其重要性自不待言, 为了保证其安全性, 可以将主密钥经加密存储到安全区域, 使用时由系统自动获取并解密, 也可以将主密钥注入到加密卡中以保证安全.

2.2 工作密钥 WK

首先调用 AES 加密算法的 constructor 函数获取主密钥数组, 通过密钥扩展可得到 10 组轮密钥, 形成一个子密钥向量: $K_i=(k1,k2,...,k10)$. 在此, 将 *k*1 到 *k*10 的顺序分别循环左移一位, 便得到 9 个新子密钥向量 $K_2\cdots K_{10}$, 形成一个如下所示的子密钥向量矩阵.

$$\begin{pmatrix} K_1 \\ K_2 \\ \vdots \\ K_{10} \end{pmatrix}=\begin{bmatrix} k_1 & k_2 & \cdots & k_{10} \\ k_2 & k_3 & \cdots & k_1 \\ \vdots & \vdots & \ddots & \vdots \\ k_{10} & k_1 & \cdots & k_9 \end{bmatrix}$$

然后再次利用主密钥 MK, 作为 128bit 明文, 将其变换为状态, 调用 AES 加密算法, 依次使用子密钥向量 *Ki*$(1\le i\le 10)$对主密钥 MK 加密. 从 AES 算法加密流程图可知, 在加密过程的每一轮变换中, 都要经过字节变换, 行位移变换和列混合变换, 然后从轮密钥中按顺序取出一组轮密钥 k_i 与状态进行异或, 且把每一轮的迭代结果作为下一轮的参数. 这样, 每使用一次子密钥向量 *Ki*, 就会经过 10 轮迭代, 提取每轮的迭代结果作为一个二级工作密钥 WK. 如此使用 10 个子密钥向量 *Ki*, 利用 AES 加密算法对主密钥 MK 进行 10 次变换, 就可以得到 100 个工作密钥 WK.

由于工作密钥是通过主密钥产生的, 故不考虑其存储和通信问题. 在这样的设计下, AES 算法的轮密钥安全性是影响数据库性能的一项关键因素, 故在本文中采用一种新的方法对应用密钥进行扩展, 来保证轮密钥生成的安全性和快速性.

2.2.1 原 AES 密钥扩展方法的缺陷

原 AES 的构造函数被调用时，加密例程获取该密钥数组并用它来生成一个名为 w[]的密钥调度表. 变量 *Nk* 代表以 32 位字为单位的种子密钥的长度，w[]最初的 *Nk*(4) 行被作为种子，剩余行从种子密钥来产生. 具体实现过程为：首先输入的种子密钥被直接复制到要扩展密钥的前 4 个字，然后每次用 4 个字填充扩展密钥数组余下的部分. 在扩展的密钥数组中，w[i]的值依赖于 w[i-1]和 w[i-4]，当 i 为 4 的倍数时，也就是下一轮的第一个 word 字，密钥生成的伪代码如下:

```
KeyExpansion(byte key[16], word w[44])
 { word=temp;
 For(i=0; i<4;i++) w[i]=(key[4* i], key[4* i +1],
 key[4* i +2], key[4* i +3]);
 For(i=4;i<44;i++)
 { temp=w[i -1];
  if (i % 4 == 0)
  temp = SubWord(RotWord(temp)) ⊕ Rcon[i/4];
  w[i] =w[i-4] ⊕ temp;
  }
 }
```

算法分析：AES 采用了直接将把种子密钥扩充，得到子密钥的方法，由于 w_i 与它之前的第一个字 w_{i-1} 和第四个字 w_{i-4} 有关. 假设我们知道了 AES 的某一轮密钥 w_i、w_{i+1}、w_{i+2}、w_{i+3}，那么可以通过 w_i、w_{i+1} 反推知道 w_{i-3}，通过 w_{i+1}、w_{i+2} 反推知道 w_{i-2}，通过 w_{i+2}、w_{i+3} 反推知道 w_{i-1}，再有 w_{i-1}、w_i 反推知道 w_{i-4}，这样便知道了该轮密钥的前一轮的全部密钥，进而获得种子密钥，这样就存在安全性隐患.

2.2.2 一种新的 AES 密钥扩展方法

如果我们能找到一种方法，使得可以保持从前向后推导时所具有的高效性，然而从后向前推导时花费极大，甚至完全无法向前推导，即密钥的生成具有单向性. 基于这种思想，本文提出如下密钥扩展算法.

对任一轮密钥，由 w_{i-4} 和 w_{i-2} 生成 w_i，由 w_{i-3} 和 w_{i-1} 生成 w_{i+1}，接着由 w_i 和 w_{i+1} 生成 w_{i+2}，由 w_{i+1} 和 w_{i+2} 生成 w_{i+3}，这样可得到一轮新密钥 w_i、w_{i+1}、w_{i+2} 和 w_{i+3}. 这样再生成下一轮密钥，以此类推可得到需要加密的所有轮次的轮密钥，算法的伪代码如下:

```
KeyExpansion(byte key[16], word w[44])
 { word=temp;
 For(i=0; i<4;i++) w[i]=(key[4* i], key[4* i +1],
 key[4* i +2], key[4* i +3]);
 For(i=4;i<44;i+=4)
  { w[i]= w[i-4] ⊕ w[i-2];
 w[i]= SubWord(RotWord(w[i])) ⊕ Rcon[i/4];
 w[i+1]= w[i-3] ⊕ w[i-1];
 w[i+2]= w[i] ⊕ w[i+1];
 w[i+3]= w[i+1] ⊕ w[i+2];
 }
}
```

算法分析：根据原来的方法分析，由于 w_{i+2} 与 w_i、w_{i+1} 有关，w_{i+3} 与 w_{i+1}、w_{i+2} 有关，所以不能用它来推导前一轮密钥. 而 w_i 与 w_{i-4}、w_{i-2} 有关，因每个字长为 32 位，则通过穷举法进行暴力破解，则需要 2^{32} 次. 同理，反推 w_{i-3}、w_{i-1} 也需要 2^{32} 次，故推出前一轮密钥需要 2^{64} 次. 此种推断次数在任何一轮均相同. 这样，改进后的算法使得在第一轮密钥扩展后即可与暴力破解相同.

通过在 Pentium®4 处理器、1.00GB 内存、Vc6.0 环境下的实验测试，分析与验证了两种方法的执行效率，并与文献[6]中的相对较好的方法 3 进行比较，结果如下表 1 所示:

表 1 改进方法对比表

方法	穷举攻击次数	程序执行时间(s)
原始算法	0	0.05602956
文献[6]方法 3	2^{128}	0.05293355
改进算法	2^{64}	0.04258412

相比而言，文献[6]方法的密钥生成的抗攻击强度最高，但是如果采用该种算法实现二级密钥系统时，就需要在生成第一轮密钥前，引入另一套与种子密钥无关的新密钥，这样必然带来该密钥在存储和通信上的新安全问题. 改进算法在保持和提升 AES 算法整体效率的同时，又使得生成的子密钥具有高强度的抗击性，使用该方法实现数据库二级密钥系统则不存在上面所述的安全问题，且设计起来简洁.

3 数据库加密系统

3.1 系统的整体设计

本数据库加密系统，采用以字段为数据加密的基本单位，以原有的数据库表存储加密后的信息. 首先本系统设计一个数据加密字典，记录用户对数据库信息具体的加密要求，即记录下哪些表的哪些字段需要

加密[7,10]; 其次建立起加密字段和工作密钥的对应关系建立起来, 为了减少存储的安全问题, 对此, 采取数组对应方式, 在程序设计中动态实现两者的对应关系. 具体系统设计如图 2 所示:

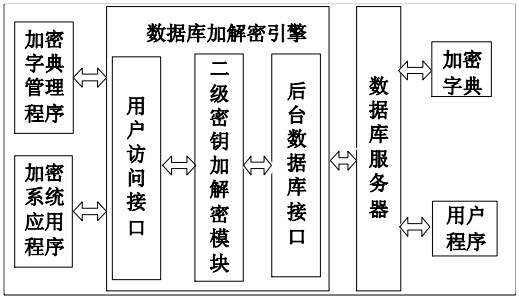

图 2 数据库系统结构图

在系统中, 用户对数据库信息各种加密要求都记录在加密字典表中; 加密字典管理程序由系统管理员使用, 负责完成加密字典表的维护工作, 是变更数据库系统加密要求的工具; 数据库加解密处理由数据库加解密引擎完成, 它是数据库加密系统的核心部分, 由加解密处理、语法分析、数据库接口等模块组成, 是二级密钥实现数据加密的地方, 负责在系统后台完成敏感字段的加解密操作.

3.2 加密解密操作

系统中, 二级工作密钥一共产生了 100 个, 这样就可以用于 100 个数据库字段加解密. 在选取工作密钥时, 为了避免数据库中要加密的字段数 m 过多, 即 $m \geq 100$ 时, 采用 $m \bmod 100$ 的方式来获得字段相应的工作密钥. 具体加解密操作如下:

当应用系统要输入一条记录到一张数据库表时, 系统从加密字典查得该表的加密要求, 如果该条记录中有字段需要加密, 则从建立的工作密钥数组 WK[t] 取出与记录中要加密的字段相对应的工作密钥, 对需要加密的字段数据进行加密, 然后将得到的密文存储到数据库的表中. 如果该表不需要加密, 则直接将数据插入到数据库的表中.

同样, 当应用系统对数据库表进行修改、查询操作时, 系统首先查看加密字典表, 若该表无加密字段, 则直接执查询. 如果该表有加密字段, 先调取相应的工作密钥解密后, 然后再执行相应的修改和查询操作.

4 结语

AES 作为当前普遍使用的数据库加密算法, 与 DES 加密算法相比之下, AES 具有更长的密钥, 不存在弱密钥和半弱密钥, 且由于采用宽轨迹的设计策略, 使得算法的抗击差分密码分析的能力增强. 本文使用 AES 加密算法来实现数据库的二级密钥加密方式, 采用新的密钥扩展方法, 提高轮密钥的安全系和快速性, 能有效的防止能量攻击和渗透攻击, 使数据库的安全性得到进一步提高, 足以满足大部分数据库的安全需求.

参考文献

1 王春爽.数据库加密中的二级密钥体制.电子商务,2008,2(527):228–230.
2 王瑾,刘自伟,黄晓芳.密钥管理在数据库加密中的应用.网络信息技术,2005,24(1):28–32.
3 侯有利.数据库加密中的二级密钥设计.通信技术,2011,44(5):52–56.
4 王先培,张爱菊,熊平.新一代数据加密标准—AES.计算机工程,2003,29(3):69–70.
5 卢正鼎,廖振松.Rijndael 算法的研究.计算机工程与科学,2005,27(6):72–74.
6 杨小东,王毅.AES 密钥扩展新方法.微电子学与计算机,2012,29(1):102–108.
7 朱鲁华,陈荣良.数据库加密系统的设计与实现.计算机工程,2002,28(8):61–63.
8 卢开澄.计算机密码学.第 3 版.北京:清华大学出社,2003.54–61,69–72.
9 贺敏伟,刘睿.高级加密标准 Rijndael 算法的一种改进.微计算机信息,2006,22(6):94–97.
10 刘自伟,罗燕琪,段琦.加密算法与密钥生成技术在数据库加密中的应用.计算机时代,2007,6:1–3.