对称密码算法专用描述语言的函数设计

谢绒娜[1] 史国振[1] 李凤华[1] 胡小敏[2]

(北京电子科技学院 北京 100010)[1] (西安电子科技大学 西安 710071)[2]

摘 要 对称密码算法专用描述语言(SDLSCA)以自然的类似专业语言的思维方式描述算法的设计思想。在密码算法设计中,对于已经证明安全性高的、性能良好的密码构件会重复采用,比如S变换、P置换等。在分析大量公开密码算法设计特点和规律的基础上,在SDLSCA语法中增加函数机制。详细给出了SDLSCA中函数的定义、调用以及实现的关键技术。并以AES为例,说明函数在算法描述的应用。实践表明利用SDLSCA中函数的功能,可以大大简化算法描述,提高算法描述的利用率和效率。

关键词 对称密码算法专用描述语言,函数,语法规则,AES算法

Design of Functions of the Special Description Language for the Symmetric Cryptographic Algorithm

XIE Rong-na[1] SHI Guo-zheng[1] LI Feng-hua[1] HU Xiao-min[2]

(Beijing Electronic Science and Technology Institute, Beijing 100010, China)[1]

(Xidian University, Xi'an 710071, China)[2]

Abstract Symmetric cryptographic algorithm description language(SDLSCA) describes the design ideas of algorithm in a natural way of thinking similar to the professional language. In the design of cryptographic algorithms, the components of algorithm which has safe and good performanc will be repeatly used, such as Substitute, Permutation. In this paper, based on the analysis of designing rules and features of a large number of public cryptographic algorithms, the function scheme was added to SDLSCA. Key techniques of the SDLSCA function were proposed in detail such as definition, calling and implementation. AES was described to illustrate the application of the function. The practice indicates that by making proper use of functions in SDLSCA, algorithm description can be simplified, and the efficiency and utilization rate are improved.

Keywords Special description language for the symmetric cryptographic algorithm(SDLSCA), Functions, Grammar rules, AES

1 引言

随着计算机网络的广泛应用和信息技术的飞速发展,信息技术已经渗透到人们日常生活的各个角落,确保信息安全的重要性不言而喻。密码技术是保护信息安全的关键技术,对整个信息系统的安全起着至关重要的作用。在密码算法研究与设计期间,密码学专家更多关心算法的设计思想与原理。文献[1,2]分别设计了分组密码算法专用描述语言PLBCA和对称密码算法专用描述语言SDLSCA。PLBCA和SDLSCA以自然的类似专业语言的思维方式描述算法的设计思想,其代码是对密码算法的直观描述,可以直接反映出算法的设计细节。利用SDLSCA,设计者可以方便地与其他设计人员交流算法的设计思想,借助SDLSCA解释器直接得到所描述算法的运行结果,为密码算法的测评和分析提供了一种强有力的辅助工具。

在密码算法设计过程中,对于一些常见已经证明的安全性高、性能良好的密码组件,或根据自己风格设计的一些安全性高的密码组件,密码学专家会重复使用[3],比如AES中的S变换,DES中的S变换、P置换,线性反馈移位寄存器、布尔函数等[4-7]。为了使算法描述更加方便、简洁、高效,同时考虑到模块化的思想,本文在SDLSCA中设计了函数的功能。SDLSCA中函数包括两大类,一类为库函数,另一类为用户自定义函数。

本文在研究、分析大量公开密码算法的基础上,将密码算法中常用的基础模块设计成库函数的形式,比如序列密码里面常用到的线性反馈移位寄存器;分组密码中常用到的P置换、S变换等;有限域上的乘和求逆;矩阵的运算等[4,5,8,9]。这样算法描述人员可以很方便地利用库函数进行算法描述,而不用考虑其具体实现过程。随着SDLSCA的进一步开发,库函数的功能不断扩充和完善。对于密码学专家自己设计的密码组件可以采用用户自定义函数的形式进行描述。

SDLSCA中用户自定义函数的定义以/function作为开

本文受国家自然科学基金(60633020,60702059),教育部重点项目(209156),北京市自然科学基金(4102056),新闻出版重大科技工程项目-数字版权保护技术研发工程(GXTC-CZ-1015004/05),北京电子科技学院工程基金(YGCJ1003)资助。

谢绒娜(1976-),女,硕士生,讲师,主要研究方向为密码理论及应用,E-mail:xierongna@besti.edu.cn;史国振(1974-),男,博士,副教授,主要研究方向为网络安全;李凤华(1966-),男,教授,博士生导师,主要研究方向为网络安全与可信计算。

始标志，以\function 作为结束标志，其结构如下所示：

```
/function fun_name(n1)(para1,para2,…)
操作语句 1;
……
操作语句 n;
\function
```

函数中的变量支持密码学中常用的任意长比特串类型的变量，操作语句支持 SDLSCA 中定义的所有的符号、操作符。同时函数支持嵌套的功能，并且在用户自定义函数中可以调用库函数。通过应用函数的功能，使算法的描述更加简洁明了，大大减少了算法描述的代码，提高了算法描述的效率，尤其在大型的复杂密码算法描述中，更能体现函数的优势。

2 函数的设计与实现

2.1 函数的定义及调用

SDLSCA 中函数描述的格式如上所示，其中 fun_name 表示函数的标识，即函数名。$n1$ 表示返回变量的大小（比特串长度），如果 $n1=0$，表示该函数不返回值。（para1，para2，…）为形参的定义，参数个数任意，其中每个参数都可以定义为传实参或传地址的形式。

下面是在 main 函数中调用上述定义的函数的形式。

```
/main
var1=3;
var2=5;
function var3=fun_name(var1,var2);/*var1、var2 分别和 para1、
            para2 的类型相同，fun_name 函数的返回值同 var3
            类型相同*/
function fun_name(var1,var2);
\main
```

2.2 函数的实现

SDLSCA 中库函数和用户自定义函数的实现采用相同的方法，整个函数的实现包含了函数语法规则的定义、函数的存储结构和函数调用时的参数传递。下面将详细介绍函数实现方法。

2.2.1 函数语法规则

根据 SDLSCA 函数定义和调用方式，本文借助 ANTLR 分别定义了函数定义和调用的语法规则[10,11]。函数定义的语法规则如下：

functionstate → SLASH FUNCTION(BLANK)+ stringname OPEN ID CLOSE OPEN(|('&')var(COMMA('&')var)*) CLOSE (NEWLINE|BLANK)*(sentencestatement(NEWLINE|BLANK)*)* BACKSLASH FUNCTION

由上述语法规则可知，函数名定义为规则“stringname”，即以字母或下划线开头的字符串，其中“SLASH”和“FUNCTION”等大写字符的标识符表示 SDLSCA 语法规则中的关键字。函数语法规则图如图 1 所示。

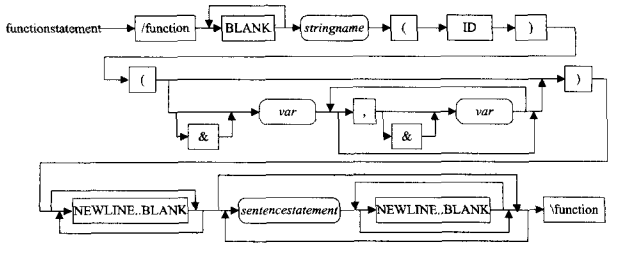

图 1 函数定义的语法规则图

函数定义规定了函数名、返回值类型、形参类型和个数、参数传递的方式、函数体。“var”描述了函数的参数。语法规则中的“sentencestatement”描述了函数体中的可执行语句。函数调用语法规则如下：

functioncall → FUNCTION BLANK+ (|var '=') stringname OPEN(|(('&')exp_1)(COMMA ('&')exp_1)*) CLOSE SEMICOLON

函数调用的语法规则图如图 2 所示。

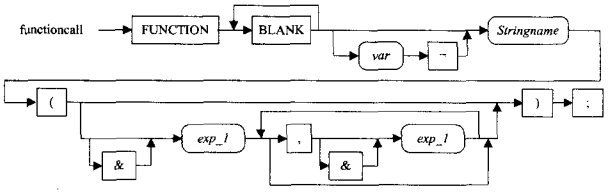

图 2 函数调用的语法规则图

2.2.2 函数的存储结构

为了存储函数的相关信息，本文设计了如下的数据结构：

```
typedef sturct function
{
  char *Name_Ptr;   //指向函数名，函数的唯一标识
  int Para_Count;   //形参的个数
  Var_Str *ParaTrasPtr;   //指向形参和临时变量的交界处
  Var_Str *reval;   //函数返回值，如果返回类型为 void 则 reval=
                      NULL
  Var_Str *FunVarHeadPtr;   //函数中变量链表的头指针
  excute_Sen *FuncSenHeadPtr;   //子函数体执行语句的链表头
                                  指针
  struct function *Next_Ptr;   //指向下一个函数结点
} Func;
```

上面定义的数据结构中，FunVarHeadPtr 为函数中的局部变量链表头指针，该链表的第一个节点存储函数返回的变量，然后依次存储函数定义的形式参数、局部变量。定义这样的结构解决了上层函数和调用函数中变量名重叠的问题。数据存储结构如图 3 所示。

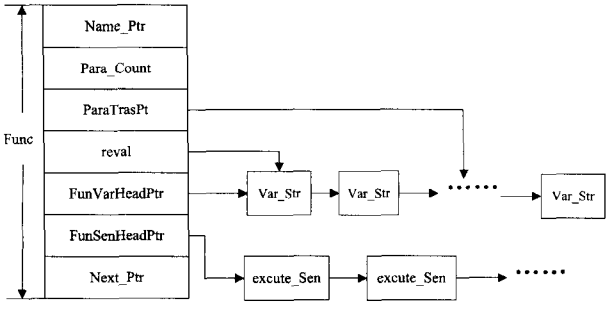

图 3 函数的存储结构

FunSenHeadPtr 指向的句子链表保存了函数中所有的可执行语句。在函数定义时，将函数中描述的语句解析成单个的可执行句子存储到 FunSenHeadPtr 指向的句子链表中。在函数调用时，首先，将外部实参传递给形参，之后逐句执行 FunSenHeadPtr 句子链表，最后将返回值赋给 reval。函数执行过程中遇到的所有变量都保存在局部变量链表 FunVarHeadPtr 中，执行完成后释放该链表。如果函数在执行的过程中又调用了其他函数，被调用函数要在创建自己的局部变

量链表前先保存调用者的变量链表，被调用函数执行完成后要恢复调用者的变量链表。

3 基于 SDLSCA 的 AES 算法的描述

下面以 AES 算法为例，给出库函数、用户自定义函数在算法描述中的灵活性和简洁性。根据 AES 算法结构特点，在算法描述中首先用户自定义了字节代替、行移位和轮密钥加等函数，在描述上述用户自定义函数时，调用了 S 变换等库函数。在随后算法加解密主过程中，直接调用这些用户自定义函数和系统自带的库函数。下面仅以字节代替函数为例说明函数在算法描述中的地位和作用。

字节代替函数的定义：

```
/function SubstituteBytes(0)(&State(128))
  (s00(8) || s10(8) || s20(8) || s30(8) || s01(8) || s11(8) || s21(8) || s31(8) || s02(8) || s12(8) || s22(8) || s32(8) || s03(8) || s13(8) || s23(8) || s33(8))=State(128);
  s00(8)=substitute(8,8){S}(s00(8));//S 变换，S 为 AES 算法中的 S 盒，substitute 为库函数
  ……
  s33(8)=substitute(8,8){S}(s33(8));
  State(128)=(s00(8) || s10(8) || s20(8) || s30(8) || s01(8) || s11(8) || s21(8) || s31(8) || s02(8) || s12(8) || s22(8) || s32(8) || s03(8) || s13(8) || s23(8) || s33(8));
\function
```

在 AES 加密主过程中直接调用字节代替函数。

```
//AES 加密主过程循环部分
/loop(s=1;step=1)=9
  RoundKey=Rkey[s * 4] || Rkey[s * 4+1] || Rkey[s * 4+2] || Rkey[s * 4+3];
  function SubstituteBytes(&State);//字节代替
  function ShiftRow(&State);//行移位
  function MixColumn(&State);//列混淆，该函数为库函数
  function AddRoundKey(&State,RoundKey);//轮密钥加，该函数为用户自定义函数
\loop
```

应用函数的功能描述 AES 算法仅需 111 行代码，而在 AES 征集时提交的 C 语言代码则需要 300 多行。同时应用函数的功能描述其他密码算法，与其公开的 C 语言代码进行比较，代码的行数都大大减少。实践证明，运用函数的功能可大大减少算法描述的代码行数，提高算法描述的准确性。

结束语 本文在分析密码算法设计特点的基础上，运用模块化的思想，设计并实现了 SDLSCA 函数的功能。最后以 AES 为例，利用库函数和用户自定义函数进行算法描述，把加密结果和公开的测试向量进行对比，进而验证密码算法描述的正确性。同时与利用函数功能之前描述的算法进行了对比，结果表明，应用函数功能可以大大简化算法的描述，提高算法描述的效率和利用率，为 SDLSCA 推广应用打下了坚实的理论和实际基础。

参 考 文 献

[1] 李凤华，阎军智，谢绒娜，等. 面向分组密码算法的程序. 设计语言[J]. 电子学报，2009，37(12)：2705-2710

[2] Li Feng-hua, Yan Jun-zhi, Xie Rong-na, et al. Research on the Programming Language forSymmetric Cryptographic Algorithms[J]. Chinese Journal of Electronics, 2010, 19(2): 303-306

[3] 吴文玲，冯登国，张文涛. 分组密码的设计与分析[M]. 北京：清华大学出版社，2009：8-13

[4] Knudsen L R. Block Ciphers-a survey [J]. LNCS, 1998, 1582: 18-48

[5] Knudsen L R. Block ciphers-analysis, design and applications [D]. DAIMI PB 485, Aarhus University, Denmark, 1994

[6] Adams C, Tavares S. The structured design of cryptographically good s-boxes [J]. Journal of Cryptology, 1990, 3(1): 27-41

[7] 金晨辉，孙莹. AES 密码算法 S 盒的线性冗余研究[J]. 电子学报，2004，32(4)：639-641

[8] NIST. Advanced Encryption Standard (AES), FIPS-Pub 197 [S]. National Bureau of Standards, 2001

[9] Biryukov A. Block Ciphers and Stream Ciphers: The State of the Art [EB/OL]. http://eprint.iacr.org/2004/094.pdf, 2004

[10] Terence P. An introduction to ANTLR [EB/OL]. http://www.cs.usfca.edu/~parrt/course/652/lectures/ANTLR.html

[11] Rod C, Paul H. Create Domain Specific Languages with ANTLR [EB/OL]. http://www.devx.com /semantic/Articel/35973?trk=DXRSS_JAVA, 2007

(上接第 8 页)

[3] Carlet C, Feng K. An Infinite Class of Balanced Functions with Optimal Algebraic Immunity, Good Immunity to Fast Algebraic Attacks and Good Nonlinearity[C]// Advances in Cryptology-ASIACRYPT 2008, LNCS 5350. Springer, Heidelberg, 2008: 425-440

[4] Dalai D K, Maitra S, Sarkar S. Basic theory in construction of Boolean functions with maximum possible annihilator immunity [J]. Des. Codes, Cryptography, 2006, 40(1): 41-58

[5] Li N, Qi W. Construction and analysis of Boolean functions of $2t+1$ variables with maximum algebraic immunity[C]// Advances in Cryptology-Asiacrypt 2006, LNCS 4284. Springer, Heidelberg, 2006: 84-98

[6] Tu Z R, Deng Y P. A conjecture about binary strings and its applications on constructing Boolean functions with optimal algebraic immunity[J]. Des. Codes Cryptogr., 2011, 60: 1-14

[7] Cusick T W, Li Yuan, Stanica P. On a combinatoric conjecture. Cryptology ePrint Archive, Report 2009/554[EB/OL]. http://eprint.iacr.org/, 2009

[8] Tang X H, Tang D, Zeng X, et al. Balanced Boolean Functions with (Almost) Optimal Algebraic Immunity and Very High Nonlinearity, Cryptology ePrint Archive[R]. Report 2010/443, 2010. http://eprint.iacr.org/

[9] Dillon J F. Elementary Hadamard Difference Sets [D]. University of Maryland, 1974

[10] Du Yu-song, Zhang Fang-guo. A note on the Tu-Deng function [C]// Cchinacrypt2011. 2011: 59-63

[11] Cohen G, Flori J-P. On a generalized combinatorial conjecture involving addition mod 2^k-1[OL]. http://eprint.iacr.org/2011/400.pdf