Computer Science and Application 计算机科学与应用, 2020, 10(12), 2306-2314
Published Online December 2020 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2020.1012243

Hans汉斯

一种基于SM4密码算法的DSP程序文件加密保护技术研究

李艳军，赵东升，李　思

北京京航计算通讯研究所，北京
Email: mailzds@126.com

收稿日期：2020年11月26日；录用日期：2020年12月21日；发布日期：2020年12月28日

摘　要

针对专用加密芯片进行DSP程序文件加密保护的成本高、需改变系统硬件架构、开发调试周期较长等现存问题，提出一种基于SM4密码算法的DSP程序文件加密保护方法。该方法是在DSP程序文件编译生成后，利用SM4密码算法对其进行加密处理，并将生成的密文程序文件存储于Flash中，只有在DSP上电运行时从Flash中读取该密文程序文件并结合匹配的密钥完成解密加载后才能正常运行。该方法以软件的方式实现了DSP程序文件的加密保护，可在不改变DSP硬件系统架构的情况下对DSP程序文件进行有效的加密保护，从而降低了硬件成本、提高了系统的安全性。

关键词

SM4，密码算法，DSP，程序文件，加密保护

Research on the Encryption and Protection Technology of DSP Program File Based on SM4 Cipher Algorithm

Yanjun Li, Dongsheng Zhao, Si Li

Beijing Jinghang Research Institute of Computing and Communications, Beijing
Email: mailzds@126.com

Received: Nov. 26th, 2020; accepted: Dec. 21st, 2020; published: Dec. 28th, 2020

Abstract

In view of the high cost, the need to change the system hardware architecture, and the long development and debugging cycle of DSP program file encryption with special encryption chip, this pa-

文章引用: 李艳军, 赵东升, 李思. 一种基于SM4密码算法的DSP程序文件加密保护技术研究[J]. 计算机科学与应用, 2020, 10(12): 2306-2314. DOI: 10.12677/csa.2020.1012243

per proposes a method of DSP program file encryption and protection based on SM4 encryption algorithm. In this method, after DSP program file is compiled and generated, SM4 cipher algorithm is used to encrypt it, and the generated ciphertext program file is stored in Flash. Only after DSP power on runs, the ciphertext program file is read from Flash and decrypted and loaded with the matching key, can it run normally. This method realizes the encryption and protection of DSP program files in the form of software, which can effectively encrypt and protect the program files loaded with DSP without changing the architecture of DSP hardware system, thus reducing the hardware cost and improving the security of the system.

Keywords

SM4, Cipher Algorithm, DSP, Program File, Encryption Protection

1. 引言

数字信号处理器(DSP)目前在航空、国防、机器视觉等领域都有广泛应用，然而随着破解技术的不断发展，DSP 脆弱的安全体制与强大的攻击方法之间形成巨大反差。为了保护 DSP 产品的经济价值和知识产权，需要对程序文件进行加密存储。

一般情况下，DSP 的用户程序往往以明文的方式存储于外部的存储器中，例如 FLASH 或 EEPROM。非法用户可以通过读取存储器中数据，或通过监听系统总线获取源程序或敏感数据，并对其进行反编译或篡改。因此有必要对用户程序进行加密处理和完整性检测。

DSP 程序文件加密保护目前可以采用 DSP 芯片加密、专用加密芯片、软件加扰混淆、FPGA 加密等方法。目前已有部分 DSP 芯片(例如 TI 公司的 28335 芯片[1])具备加密功能，但是由于芯片是国外厂商，无法确定其是否留有后门，一般不采用此方法。专用加密芯片或者 FPGA 加密需要修改硬件架构，对于硬件成熟度较高的产品并不适用[2]。软件加扰即向原有源程序中添加干扰代码，在增加阅读难度的同时也增大了 DSP 程序文件占用内存的大小，有可能会影响软件正常运行。基于上述情况，本文提出了一种以软件的方式实现了 DSP 程序文件的加密保护，可在不改变 DSP 硬件系统架构的情况下对 DSP 程序文件进行有效的加密保护。

2. DSP 程序文件结构分析

在对 DSP 程序文件进行加密之前，需要对其文件结构进行分析，确保加密前后程序文件的运行结果能够完全一致。目前主流的 DSP 芯片主要有 TI 公司的 TMS320 系列和 AD 公司 ADSP21xx 系列，因此需对这两种芯片程序文件格式分别进行分析。TMS320 系列 DSP 芯片采用 CCS 开发工具进行编译，生成的程序文件为OUT格式，需要对其进行转换为BIN格式烧录到Flash中；ADSP21xx 系列芯片采用 ADSP++ 开发工具进行编译，生成的程序文件为 LDR 格式。

2.1. OUT 格式和 BIN 格式分析

使用 CCS 开发后的程序文件为 OUT 格式，文件包含文件头、扩展头、段落头、段落数据、重定位表、行号表、符号表、字符串表，文件结构如图 1 所示。文件头用于保存 OUT 文件的基本信息，如文件

标识、各个表的位置等；根据文件头的基本信息，决定扩展头是否存在，扩展头用于描述文件头没有描述的信息；段落头用于描述段落信息，每个段落都有一个段落头描述，段落的数目在文件头中会指出；段落数据通常是 OUT 文件中最大的数据段，每个段落的真实数据就保存在这个位置；重定位表用于描述 OUT 文件中符号的重定位信息；行号表用于描述 OUT 文件行号信息；符号表用于描述文件中所用到的所有符号的信息；字符串表用于保存程序中所用到的字符串。

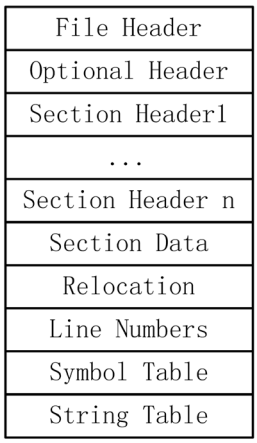

Figure 1. OUT file structure
图 1. OUT 文件结构

与程序加密有关的主要有文件头、扩展头、段头和段数据，具体数据结构如图 2 所示：可以从文件头中获取到区块个数，扩展头大小；从扩展头获取到代码段和数据段偏移地址、段长度、程序入口点等信息。后续加密操作可以基于此结构对代码段和数据段进行选择性加解密。

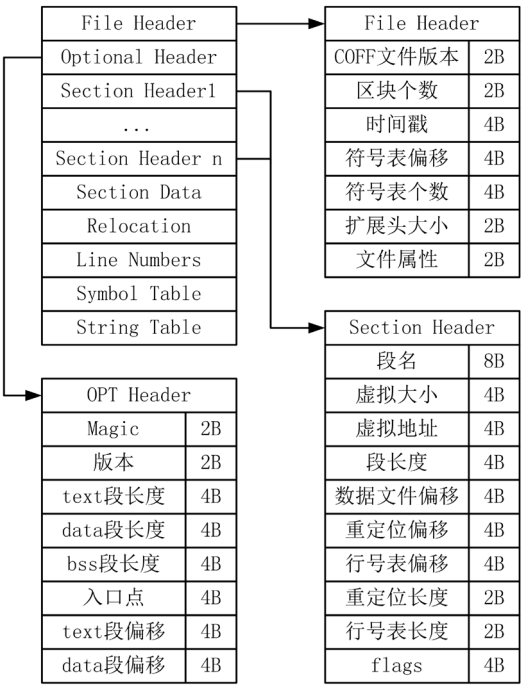

Figure 2. Data structure of OUT file
图 2. OUT 文件数据结构

在研制过程中开发人员需将 OUT 格式文件转换为 BIN 格式进行烧录，一般的转换方法为先将 OUT 文件通过 TI 工具 hex6x_33.exe 依据开发人员编写的 cmd 文件，转换为 hex 文件和 map 文件，再通过 hex2bin.exe 生成 BIN 格式文件，流程如图 3 所示。

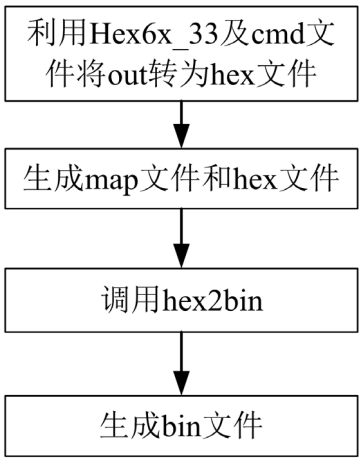

Figure 3. BIN file generation process
图 3. BIN 文件生成流程

转换后的 BIN 格式文件如图 4 所示：文件头为 4 字节_c_int00 的地址，软件从该地址开始初始化并运行；接着为段 n 的长度、地址和段数据，最后为文件结尾。通过分析与测试验证可知，该文件是连续加载到内存中，因此可以对 BIN 文件全部数据进行加解密处理。

_c_int00地址	4B
Sec1长度len1	4B
Sec1基址add1	4B
Sec1数据data1	len1 B
Sec2长度len2	4B
Sec2基址add2	4B
Sec2数据data2	len2 B
…	
SecN长度lenN	4B
SecN基址addN	4B
SecN数据dataN	lenN B
结尾0x00 0x00 0x00 0x00	4B

Figure 4. BIN file structure
图 4. BIN 文件结构

2.2. LDR 格式分析

针对 AD 公司的 DSP 芯片开发人员需要使用 Visual DSP++工具进行开发，最终生成的程序文件为 LDR 格式文件。某工程项目的 LDR 文件示例如图 5 所示。该文件中数据由多行组成，每行的数据格式

为：起始为 1 个“：”，1 个字节的数据长度 n，6 字节内存映射地址，n 字节数据，1 字节校验，具体的 LDR 文件数据结构如图 6 所示。由于该格式文件的内存映射地址往往是不连续的，无法对其直接进行统一处理，因此，针对 LDR 格式文件，需首先从中分别过滤出地址部分呵数据部分，然后才能进行加解密处理。

```
:2003C000000000000000000000000000000000000000000000000000000000001D
:2003E000000000000000000000000000000000000000000000000000000000000FD
:10040000020004090D00800FFFFFFFF0000000046
:10041000020000408ED00800FFFFFFFF0000000038
:0C042000010000408DD00800000000002A
:20042C00160B00400010000353F4B04700800B2FFFF4D08800000B20D311CA8343F4C0417
:20044C00700800B200804F0800FFFFB20F391CA800805008000040B2D00119A8140060B152
:20046C000C0C4C8804805288B24400A81E0C51881E0A058C080C4B8830FF8871057B9805A0
:20048C00057BB895343F4400700800B20680468886140028000C0B3363F450470080032B1
:2004AC000000C0330000C0B3280008F1333F4400700800B20080460800FFFFB286141CA8ED
:2004CC000080470800008 0B2A70019A8100060B1343F8501700800B2353F840170080032C0
:2004EC000000C0330000C0B3F0FE8871057B9805057BB895068044883 63F440470080032 00
```

Figure 5. LDR file example
图 5. LDR 文件示例

:	数据长度n	内存映射地址	数据	校验和
	1B	6B	nB	1B

Figure 6. LDR file structure
图 6. LDR 文件结构

3. SM4 密码算法

1) 密码算法描述

加密算法主要有对称加密、非对称加密、Hash 算法。对称加密算法在加解密过程中使用相同的密钥，应用简单快捷，可以加密大量数据。非对称加密算法采用公钥和私钥两种不同的密钥进行加解密，密钥成对出现，安全性高，但是算法强度复杂。Hash 算法时一种单向算法，通过该算法可以对目标数据生成一段特定长度且唯一的 Hash 值，却不能通过 Hash 值重新获得目标数据，通常用于数字摘要，完整型校验等。在 DSP 程序文件加密过程中，主要采用对称加密算法进行数据加密，本文选用 SM4 对称加密算法进行数据加密。

SM4 算法是由国家密码管理局公布的一种分组密码算法，在使用上表现出了安全高效的特点，与其他分组密码算法相比较有以下的优势[3]。

- 首先是算法资源利用率高，表现为在密钥扩展算法与加密算法可共用；
- 其次加密算法流程和解密算法流程一样，只是轮密钥顺序相反，因此无论是软件实现还是硬件实现都非常方便；
- 算法中包含异或运算、数据的输入输出、线性置换等模块，这些模块都是按 8 位进行运算的，现有的处理器完全能处理。

2) 算法整体结构

SM4 分组密码算法主要包括加密算法、解密算法以及密钥扩展算法三部分。算法分组长度为 128bit，密钥长度为 128bit。加密算法与密钥扩展算法均采用 32 轮非线性迭代结构，以 32 位为单位进行加密运算，每一次迭代运算均为一轮变换函数 F。SM4 算法解密过程与加密一致，只是使用加密轮密钥的逆序进行解密，整体结构如图 7 所示[4]。

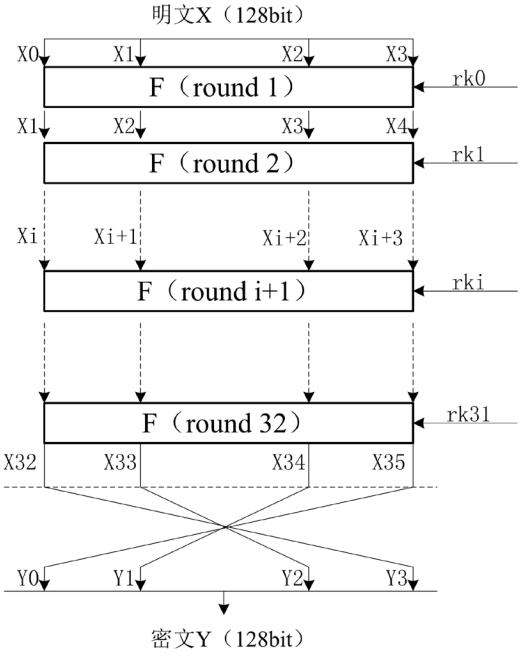

Figure 7. SM4 algorithm structure
图 7. SM4 算法结构

整体的加密函数为：

$$X_{i+4}=F\left(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i\right)=X_i\oplus T\left(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i\right)$$

其中 X_0、X_1、X_2、X_3 为 128bit 明文分组后的结果，轮密钥表示为 rk_i。T 为一个合成置换，由非线性变换 τ 和线性变换 L 复合而成，即 $T(\cdot)=L(\tau(\cdot))$。

非线性变换 τ 由 4 个平行的 S 盒构成，S 盒的数据均采用 16 进制。设输入为 $A=\left(a_0,a_1,a_2,a_3\right)\in\left(Z_2^8\right)^4$，输出为 $B=\left(b_0,b_1,b_2,b_3\right)\in\left(Z_2^8\right)^4$，则 $B=\left(b_0,b_1,b_2,b_3\right)=\tau(A)=\left(Sbox\left(a_0\right),Sbox\left(a_1\right),Sbox\left(a_2\right),Sbox\left(a_3\right)\right)$。

线性变换 L 公式如下，其中 B 为非线性变换得到的字：

$$C=L(B)=B\oplus(B\ll 2)\oplus(B\ll 10)\oplus(B\ll 18)\oplus(B\ll 24)$$

3) 密钥扩展

已知加密密钥 $MK=\left(MK_0,MK_1,MK_2,MK_3\right)$，系统参数 $FK=\left(FK_0,FK_1,FK_2,FK_3\right)$，采用 16 进制表示，具体为：$FK_0=\left(A2B1BAC6\right)$，$FK_1=\left(56AA3350\right)$，$FK_2=\left(677D9197\right)$，$FK_3=\left(B27022DC\right)$；固定参数 $CK=\left(CK_0,CK_1,\cdots,CK_{31}\right)\in\left(Z_2^8\right)^4$。$rk_i$ 为轮密钥，轮密钥由加密密钥生成。

首先 $\left(K_0,K_1,K_2,K_3\right)=\left(MK_0\oplus FK_0,MK_1\oplus FK_1,MK_2\oplus FK_2,MK_3\oplus FK_3\right)$；然后对 $i=0,1,2,\cdots 31:rk_i=K_{i+4}=K_i\oplus T'\left(K_{i+1}\oplus K_{i+2}\oplus K_{i+3}\oplus CK_i\right)$，该变换与加密中的 T 变换基本相同，只是将其中的线性变换为 $L'(B)=B\oplus(B\ll 13)\oplus(B\ll 23)$。

4) 加/解密算法

定义反序变换 R 为：$R\left(A_0,A_1,A_2,A_3\right)=\left(A_3,A_2,A_1,A_0\right),Ai\in Z_2^{32},i=0,1,2,3$

设明文输入为 $\left(X_0,X_1,X_2,X_3\right)\in\left(Z_2^{32}\right)^4$，密文输出为 $\left(Y_0,Y_1,Y_2,Y_3\right)\in\left(Z_2^{32}\right)^4$，轮密钥为 $rk_i\in Z_2^{32},i=0,1,2,\cdots,31$。则本算法的加密变换为：

$$X_{i+4}=F\left(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i\right)=X_i\oplus T\left(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i\right),i=0,1,\cdots,31$$

$$\left(Y_0,Y_1,Y_2,Y_3\right)=R\left(X_{32},X_{33},X_{34},X_{35}\right)=\left(X_{35},X_{34},X_{33},X_{32}\right)$$

本算法的解密变换与加密变换结构相同，不同的仅是轮密钥的使用顺序。

加密时轮密钥的使用顺序为：$(rk_0, rk_1, \cdots, rk_{31})$。

解密时轮密钥的使用顺序为：$(rk_{31}, rk_{30}, \cdots, rk_0)$。

4. 基于 SM4 密码算法的 DSP 程序文件加密处理方法

4.1. 程序文件加密处理

根据文件解析和 SM4 密码算法的实现结果，需要设计对程序数据的加密，一般程序加密过程在上位机软件中运行，加密过程如图 8 所示。

1) 首先打开上位机软件，将程序文件读入内存中，根据选择的文件格式，按照分析结果对其进行解析，提取出需要加密的数据，通常我们选择对代码段(.text)和数据段(.data)进行加密；

2) 然后进行密钥初始化：根据 SM4 密码算法密钥扩展过程，对系统参数，加密密钥，固定参数进行密钥扩展，生成轮密钥，其中加密密钥一般由用户输入或采用随机数生成器生成随机数；

3) 对(1)中提取出的待加密数据进行摘要生成，此部分采用摘要算法，例如 MD5、SM3，具体算法原理参考文章[5]；

4) 对(1)中提取出的待加密数据按照密码算法要求(128bit)进行分组；

5) 将分组后的数据输入到密码算法的加密函数中进行加密，生成密文数据后，写入到密文文件中；将(4)中的摘要数据也写入到密文文件中，最终得到加密后的程序文件。

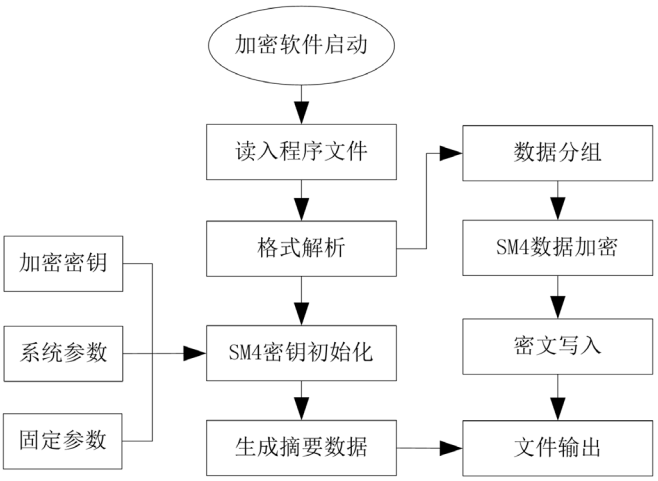

Figure 8. Encryption process
图 8. 加密流程

4.2. 自动解密设计和完整性验证

DSP 通常依赖片外 Flash 启动系统，其主要工作原理是：DSP 系统在上电复位时序结束后，DMA 控制器从片外 Flash 的起始地址空间拷贝一定数目的代码到片内 RAM，并从 RAM 空间零地址处开始存放。当数据块传输完成后，CPU 脱离“阻塞”状态，开始从片内 RAM 的零地址处执行相应的指令[6]。通过上述一次硬件代码加载(一级 BootLoader)往往难以将用户应用搬迁到系统 RAM，需要再编写二次拷贝代码，进而形成二次启动加载程序(二级 BootLoader)。DMA 控制器先将二次启动程序拷贝到片内 RAM，执行二次启动加载程序，再将 FLASH 中的程序完全拷贝到高速的片内 RAM 中，再跳转到程序入口 _c_int00 地址，完成启动过程。

由于密文程序文件需要解密为 DSP 可识别的明文二进制码，才能够正确运行，因此解密过程一般设计在二级 BootLoader 程序中，具体过程如图 9 所示：

1) DSP 系统上电；

2) 系统启动，自动运行一级 BootLoader；

3) 将二级 BootLoader 搬移到 RAM 中运行；

4) 在二级 BootLoader 中将 Flash 中的密文程序数据搬移到 RAM 中；

5) 使用解密密钥、系统参数、固定参数进行 SM4 密钥扩展，完成密钥初始化；

6) 根据 SM4 分组要求，对 RAM 中的密文数据进行分组；

7) 将分组后数据输入到 SM4 解密函数，对密文进行解密；

8) 对解密完成后的明文数据采用加密选用的摘要算法生成摘要；

9) 与加密时生成的摘要数据比较，进行完整型检测[7]。摘要一致则证明完整型验证通过，跳转到 _c_int00 中执行；否则证明程序被篡改或者解密错误，将直接退出。

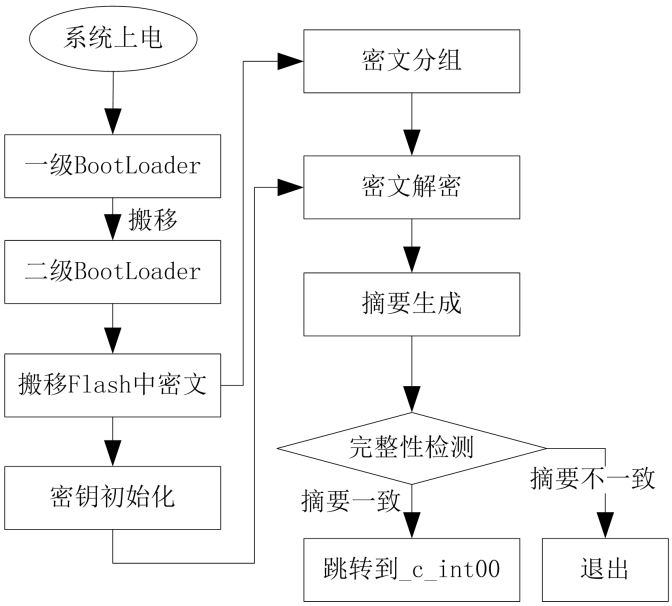

Figure 9. Decryption process
图 9. 解密流程

5. 总结

本文在分析研究 DSP 程序文件的结构和 SM4 密码算法的基础上，提出了一种基于 SM4 密码算法的 DSP 程序文件加密保护方法。该方法利用国密 SM4 对称加密算法，以软件的方式实现了 DSP 程序文件的加密保护，能够在不改变 DSP 硬件系统架构的情况下对程序文件进行加密处理和完整性验证，增加了非法反编译破解和篡改的难度，降低了系统的硬件成本，保障了 DSP 关键程序文件的安全性，具有广阔的应用前景。

参考文献

[1] TMS320x2833x Technical Reference Manual. Texas Instruments Corporation, 2006.

[2] 刘长军, 林嘉宇. 为 DSP 程序构造的加密体制[J]. 单片机与嵌入式系统应用, 2002(4): 5-7.

[3] 冯燕强, 等. SM4 算法原理及实现[J]. 有线电视技术, 2019(6): 94-96.
[4] 卞建秀, 等. 基于 SM4 和 ECC 的混合加密算法研究[J]. 计算机应用与软件, 2016, 33(10): 303-306.
[5] 王小云, 于红波. SM3 密码杂凑算法[J]. 信息安全研究, 2016, 2(11): 983-994.
[6] 孙东亚. TMS320C6000 系列 DSP 多应用的启动方法[J]. 信息通信, 2017(3): 51-52.
[7] 余国义, 等. 一种适用于 DSP 的安全模块的设计[J]. 微电子学与计算机, 2009, 26(7): 1-4.