电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

# 硕士学位论文

# MASTER THESIS


论文题目 SMS4 算法的白盒密码算法设计与实现

学科专业 密码学

学　　号 201321260350

作者姓名 尚　培

指导教师 李胜强　副教授

# 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：尚培　　日期：　　年　　月　　日

# 论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：尚培　　导师签名：李胜强

日期：2016 年 6 月 20 日

分类号 _____ 密级 _____

UDC [注1] _____

# 学位论文

## SMS4 算法的白盒密码算法设计与实现

（题名和副题名）


尚培

（作者姓名）

指导教师 李胜强 副教授

电子科技大学 成都

（姓名、职称、单位名称）


申请学位级别 硕士 学科专业 密码学

提交论文日期 2016.6 论文答辩日期 2016.5.26

学位授予单位和日期 电子科技大学 2016 年 6 月

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

# White-Box Cryptography Algorithm Design And Implementation Of SMS4

A Master Thesis Submitted to

University of Electronic Science and Technology of China


Major: **Cryptography**

Author: **Shang Pei**

Supervisor: **Li Shengqiang**

School: **National Key Laboratory of Science and Technology on Communications**

# 摘 要

近年来，随着密码攻击方式的发展，传统的黑盒密码攻击模型显得越来越脆弱。不同于传统的密码攻击模型，白盒攻击模型赋予了攻击者更多的能力，在白盒攻击环境下，攻击者对密码算法的运行拥有完全的控制权，并且能够观察及更改软件运行时的数据。能够抵抗白盒攻击的密码算法称为白盒密码。设计白盒密码的目的是为了在白盒攻击环境中，对密钥信息进行隐藏，有效地防止攻击者在白盒攻击环境中获得密钥信息。因此，设计安全高效的白盒密码算法具有非常重要的意义。

SMS4 算法是我国用于无线局域网产品的第一个分组密码算法，受到广泛应用。本文的研究目标是设计安全高效的白盒 SMS4 密码算法，并且开发出白盒 SMS4 算法应用软件。主要研究内容包括：

（1）对白盒密码及其设计方法和安全性评估进行了系统学习和总结；分析了肖雅莹等人提出的白盒 SMS4 算法的设计特点；分析林婷婷等对肖-白盒 SMS4 算法的攻击方法以及其能够攻击成功的原因。

（2）在肖-白盒 SMS4 算法的基础上提出两种改进算法，并对改进算法进行效率分析和安全性分析。改进算法一将肖-白盒 SMS4 算法中的四个查找表改为两个查找表，节省了查表次数，但增大了存储空间；从安全性方面来讲，改进算法一可以将攻击时间复杂度提高至少 2 倍。改进算法二简化了肖-白盒 SMS4 算法中的参数 $P_{i+j}$、$P'_{i+4}$ 和 $P''_{i+4}$，这样在计算保存参数的过程中会减小计算的复杂度，简化参数后的白盒多样性和白盒含混度仍能满足一定的安全性；另一方面增加了外部编码，完善了白盒 SMS4 算法。

（3）针对改进的白盒 SMS4 算法，完成白盒 SMS4 算法的软件实现，并在 MFC 上开发出可视化加解密应用软件；并且将白盒 SMS4 算法与黑盒 SMS4 算法进行比较，从实现效率、运行速度以及安全性等方面分别进行分析。白盒 SMS4 算法能够在白盒攻击环境中有效地保护密钥信息，拓宽了 SMS4 算法的应用领域。

**关键词：**白盒密码，白盒攻击环境，SMS4 算法

# ABSTRACT

In recent years, with the development of cipher attacks, the traditional black-box cryptographic attack model becomes more and more fragile. Unlike traditional cryptographic attack model, white-box attack model gives attackers more capacities, in White-Box Attack Context, the attacker can control cryptographic algorithms over the software and have the ability to observe and modify the running data. The cryptographic that can resist White-Box Attack Context is called White-Box Cryptography. The objective of designing White-Box Cryptography is to hide the key information, prevent attackers from obtaining the key in White-Box Attack Context effectively. Therefore, it is important to design safe and efficient White-Box Cryptographic algorithm.

SMS4 algorithm is the first block cipher algorithm for wireless LAN products, which is widely used in our country. The research goal of this paper is to design White-Box SMS4 and realize White-Box SMS4 cryptography algorithm with software platform. The main research contents include:

Firstly, this thesis describes White-Box Cryptography, and it's design methods and safety assessment, analyzes features of White-Box SMS4 implementation by Xiao Yaying, studies a attack to the scheme by Lin Tingting et al and reasons for the success.

Secondly, this thesis provides two improved algorithms of Xiao-White-Box SMS4 algorithm, and analyzes the efficiency and safety of the improved algorithm. The first improved algorithm reduces four lookup tables to two, saving the number of look-up tables, but it increases the extra space. On the other hand, in terms of safety, the attack time complexity can be increased by at least 2-fold. The second improved algorithm to Xiao-White-Box SMS4 algorithm on the one hand, it simplifies parameter $P_{i+i}$ 、 $P'_{i+4}$ and $P''_{i+4}$ , saving the computational complexity in the calculation of the parameters, and it's white-box diversity and white-box ambiguity can meet certain security; on the other hand, it increases an external encoder, making the entire algorithm integrity.

Finally, this thesis provides the implementation of White-Box SMS4 which is based on the second improved algorithm this thesis gived, and implements a visual encryption applications on MFC; this thesis also provides the compare of the implementation efficiency, speed and security between White-Box SMS4 and Black-Box SMS4 algorithm. The White-Box SMS4 algorithm can protect the key

information in White-Box Attack Context. White-box SMS4 algorithm can effectively protect the key information in White-Box Attack Context, and broaden the application field of SMS4 algorithm.

**Key words**: White-Box Cryptography, White-Box Attack Context, SMS4

# 目 录

# 图目录

# 表目录

# 第一章 绪论

## 1.1 研究背景与意义

在传统密码学中，密码算法的设计都是建立在黑盒模型的基础上。黑盒模型假设密码算法的运行环境是安全的，攻击者只能访问算法的输入和输出，不能观察代码执行及动态加密的过程，通信安全的关键是保护好密钥。随着数字化信息的广泛应用，密码软件的运行环境不再如以往一样单纯和可信。例如[1]，支持数字版权保护系统的播放器，比如一个多媒体视频播放器，由于视频的内容在被播放之前是以加密状态存在的，所以播放器中就一定要嵌入一个解密用的密钥，以便在播放时实时解密。在这种情况下，密码软件在解密的过程中就是不安全的，攻击者有可能使用静态分析工具、调试器、直接读取内存中的数据等方法和工具找出播放器中的密钥。这一幕经常出现在对电视机机顶盒的攻击案例中。除此之外，最近几年提出的旁路攻击（也称为灰盒攻击），如时间分析、功耗分析、插入错误分析等[2-5]，致使很多密码系统变得不堪一击。然而，传统的黑盒攻击模型很少考虑到这些问题。由此可见，传统的黑盒模型已经不再满足越来越高的安全要求。

2002 年，Chow 等人首先提出白盒攻击环境 [6]（White-Box Attack Context，简称 WBAC）的概念，其描述如下：

（1）在同一主机中，可以同时运行加密软件与拥有特殊权限的攻击软件，并且对密码算法的运行过程拥有完全的控制权；

（2）程序的动态执行过程可以被监视；

（3）密码算法的实现细节及执行过程是可见的，并且可被任意修改。

白盒攻击环境也可理解为白盒攻击模型，它给予了攻击者更强大的能力。与黑/灰盒攻击模型相比，白盒模型可以认为是最坏的模型，然而它也是在信息技术发展迅速的今天很适用的模型，用于分析在不可信环境下运行的密码算法。

为了保证密码算法在不可信的终端运行而不受到威胁，设计出在白盒攻击环境下能够保证安全性的密码算法将是信息安全领域面临的重大课题之一。能够抵抗白盒攻击的密码算法称为白盒密码。白盒密码的目的就是为了在白盒攻击环境中，将密钥信息隐藏在密码算法中，防止密钥被攻击者所获取。因此，研究安全高效的白盒密码算法具有非常重要的意义。

SMS4 算法是 2006 年 1 月中国国家商用密码管理办公室公布的对称密码算法，主要用于无线局域网产品中。2009 年，肖雅莹等人[7]提出 SMS4 算法的白盒设计方法，其采用的主要方式是查找表和仿射变换相结合的方式。从白盒多样性和白

盒含混度的角度来讲，肖-白盒 SMS4 的安全性可以达到一定的需要；其作者也针对现有的攻击方法，即 Billet 等人提出的 BGE 攻击方法[8]，作出了分析，说明了其白盒方案的安全性。但是，这个方案存在安全漏洞。2013 年，林婷婷等人[9]提出针对文献[7]中白盒 SMS4 算法的一种有效攻击方法。该攻击方法利用类似 Billet 等人的分析方法，并结合差分分析法、求解方程组等方法，最终能以较低的复杂度恢复出 SMS4 的轮密钥。因此，SMS4 算法的白盒设计研究还有很大潜力，其在白盒环境中的应用更是十分重要，研究安全有效的 SMS4 算法的白盒设计方法具有很大的价值。

## 1.2 白盒密码研究现状

近年来密码学领域兴起了一种新型攻击——白盒攻击，它比恶意主机攻击[10,11]和旁信道攻击[12]对传统密码有更大的威胁。可以说，白盒攻击环境是在恶意主机的基础上发展起来的。虽然白盒密码的研究还处于初级阶段，但仍然取得了一些研究成果。

2002 年，Chow 等人[6]首先提出白盒攻击环境的概念，并分别提出了 AES 算法的白盒实现[6]以及 DES 算法的白盒实现[13]，主要思想是：将已有的密码算法进行分割，将选定的密钥隐藏在算法的某个步骤中，然后对每个小部分进行可逆的置乱编码，并将每个部分的所有输入输出做成查找表形式。同时，Chow 等还给出了白盒密码设计的两个评价指标：白盒多样性和白盒含混度。

2002 年，Jacob 等人构造了一种注入错误攻击[14]实现了对 Chow 白盒 DES 的一种有效攻击。

2004 年，Billet 等人[8]提出了 BGE 攻击方法，该方法有效地攻击了 Chow 白盒 AES。

2005 年，Link 等人[15] 改进了 Chow 白盒 DES，得出一种能抵抗当时已知攻击的新的 DES 白盒实现方案。

2006 年，Bringer 等人[16]提出了利用插入扰乱项的方法来构造白盒 AES 的方案，并说明了该方案满足一定的安全性。

2007 年，Wyseur 等人[17]利用截断差分分析成功攻击了无外部编码的 Chow 白盒 DES。

2009 年，肖雅莹等人[18]对 Chow 白盒 AES 进行改进，提出一种新的白盒 AES 实现方法，该方法能够抵抗类似 BGE 的攻击。同时，肖雅莹等人[7]还提出了白盒 SMS4 算法，主要采用查找表和仿射变换相结合的方式。

同年，Wyseur 在其博士论文[19]中给出了一个白盒密码的目标描述，并对混淆

理论进行详细的描述，说明了混淆理论在白盒密码中的地位。

2010 年，Mulder 等人[20]提出了对 Bringer 白盒 AES 方案的一种攻击方法，以较低的复杂度恢复出了等价密钥。

2012 年，Mulder 等人[21]破解了 Xiao-Lai[18]的白盒 AES，采用的基本算法是线性/仿射等价算法。

2013 年，林婷婷等人[9]提出针对文献[7]中白盒 SMS4 算法的一种有效攻击方法。该攻击方法利用类似 Billet 等人的分析方法，并结合差分分析法、求解方程组等方法，最终能以较低的复杂度恢复出肖-白盒 SMS4 的轮密钥。

总的来说，白盒密码的研究还处于探索阶段 。对白盒密码的基础理论研究以及如何将白盒密码应用到实际环境中将是白盒密码未来的发展趋势。

## 1.3 研究目标及研究内容

SMS4 算法是我国用于无线局域网产品的第一个分组密码算法，受到广泛应用。本文的研究目标是：研究 SMS4 算法及肖-白盒 SMS4 算法设计的特点，提出在白盒攻击环境下适用的安全高效地白盒 SMS4 算法；并且完成白盒 SMS4 算法的软件实现。主要研究内容包括：

（1）对白盒攻击环境、白盒密码的定义、白盒密码的设计方法及安全性评估标准进行系统地学习；分析肖雅莹等人提出的 SMS4 算法的白盒设计的特点，以及林婷婷等人对此设计实施的攻击方法和能够攻击成功的原因；在此基础上提出白盒 SMS4 算法的改进算法，并对改进算法进行效率分析和安全性分析。

（2）针对改进算法，完成白盒 SMS4 算法的软件实现，并在 MFC 上开发出可视化加解密应用软件；并且将白盒 SMS4 算法与黑盒 SMS4 算法进行比较，分别从实现效率、运行速度以及安全性等方面进行分析。

## 1.4 本文结构安排

第一章 绪论，介绍本文的研究背景、白盒密码的研究现状、研究目标及研究内容；

第二章 介绍相关密码学基础知识。主要包括密码体制与常见的密码攻击方法；白盒密码的设计方法以及安全性分析方法；密码算法的安全性评估以及效率评估；

第三章 介绍肖雅莹等人提出的白盒 SMS4 算法的设计方法、效率分析以及安全性分析；

第四章 提出白盒 SMS4 算法的两个改进算法，并对改进算法进行安全性分析和计算复杂度分析；

第五章 介绍白盒 SMS4 算法的软件实现，首先简单介绍软件实现的需求分析、总体结构以及开发平台，然后具体介绍各部分的设计与实现，最后给出系统界面及一个实例，并与黑盒 SMS4 加解密过程进行对比。

第六章 对本文进行全文总结，并指出进一步研究工作。

# 第二章 基础知识

本章介绍密码学相关基本概念和白盒密码的相关基础知识。首先介绍密码体制和密码攻击方法；然后介绍白盒密码的相关基本概念、白盒密码的设计方法以及安全性评估标准；最后介绍一般密码算法安全性评估以及实现效率评估方法。

## 2.1 密码学相关基本概念

### 2.1.1 密码体制

密码学这个词来源于希腊单词kryptós和gráphein，意思分别是“隐藏”和“书写”。严格地说，它是研究如何隐秘的传递信息的学科。密码学包括两个互补的领域，密码编码学研究密码变化的客观规律，应用于保守通信的秘密；而密码分析是研究隐藏信息的方法来获取通信情报。

一个密码通信系统模型如图 2-1 所示。

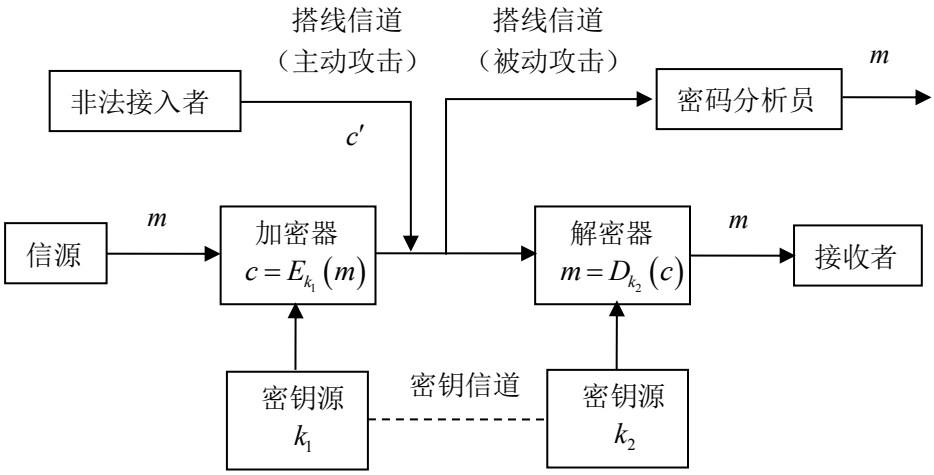


图 2-1 密码通信系统模型

密码体制可以分为两大类：对称密码体制和非对称密码体制，前者加解密使用相同的密钥，又称单钥体制；后者加密密钥和解密密钥不同，又称为双钥体制。

对称密码体制具有以下特点：优点是加解密速度快，而且算法公开，主要用于数据的加密及消息的认证；缺点是密钥的分发实现困难，密钥管理复杂，而且对称密码算法无法完成身份认证等功能，因此不适用于开放的网络环境。非对称密码体制的算法强度复杂，安全性较高，不过这也导致了其加密与解密的速度慢

于对称密码体制；另外，非对称密码体制的加密与解密能力是独立的，多个用户使用同一个用户的公钥加密消息，得到的密文只能由一个用户用其私钥解密，因此可以在公开网络中实现保密通信；同时，它的另一个特点：一个用户用其私钥加密的消息可以被多个用户使用其公钥进行解密，使得非对称密码体制可以在认证系统中对消息进行数字签名。由于对称密码与非对称密码各自的特点，在密码系统中常用的一种方法是使用非对称加密算法的公钥对对称加密算法的密钥进行加密后发送出去，接收方收到后使用非对称加密算法的私钥进行解密可得到对称加密算法的密钥，然后双方使用对称加密算法进行传输数据。

### 2.1.2 密码攻击

根据“Kerckhoffs 假设”，我们分析一个密码系统是否安全，一般是在假设攻击者知道密码系统的情况下进行的。在这个假设下，密码攻击可以分为以下五类：

**唯密文攻击**：攻击者只掌握截获的密文。

**已知明文攻击**：攻击者除了掌握密文，还知道当前密钥下的部分明/密文对。

**选择明文攻击**：攻击者能够获得当前密钥下一些特定的明文所对应的密文。

**选择密文攻击**：攻击者能够获得当前密钥下一些特定的密文所对应的明文。

**选择文本攻击**：选择明文攻击和选择密文攻击两种方式的结合。攻击者知道待破译的密文，以及知道任意选择的明文和它对应的密文；并且知道当前密钥下特定的密文和它对应的明文。

显然，以上五种攻击类型的攻击强度依次增大。

攻击密码体制的方法有两种：穷举攻击法和密码分析法。

穷举攻击法是最基本的方法，有时也是很有效的一种方法。然而穷举攻击所付的代价与密钥大小成正比，当密钥达到一定的长度时，穷举攻击将变得不切实际。因此，人们越来越重视密码分析法。目前具有代表性的密码分析方法有：线性密码分析法、差分密码分析法、相关密钥密码分析法、侧信道攻击等等。

线性密码分析法是一种已知明文攻击的分析方法。它最早是由 Matsui 在 1993 年的欧密会上针对 DES 算法提出的[22]，并在第二年的美密会上进行了改进，第一次用实验的方式给出了对 16 轮 DES 的攻击[23]。线性分析法的基本思想是:利用密码算法中的明文、密文和密钥的不平衡性，采用线性逼近来恢复某些密钥。

差分密码分析法是一种选择明文攻击的分析方法。它是密码学家 Eli Biham 和 Adi Shamir 于 1990 年提出的[24]，虽然最初也是针对 DES 算法提出的，不过它适用于几乎所有的分组密码，被公认为近年来密码分析的最大成果。差分密码分析的基本思想是:通过分析明文对的差值对密文对的差值的影响，来恢复某些密钥信

息。以差分密码分析为基础，有一些扩展密码分析法，包括截断差分密码分析、高阶差分密码分析、不可能差分密码分析和飞来器攻击等。

相关密钥密码分析法是 1992 年和 1993 年，Knudsen 和 Biham 各自独立提出的[25,26]。它不同于线性密码分析和差分密码分析，相关密钥密码分析赋予了攻击者更多的能力，即攻击者已知（或可以选择）密钥之间的关系。在攻击密码算法时，攻击者可以利用密钥扩展算法的不足，选择合适的与当前密钥相关的密钥，寻求在原有密钥和选择的相关密钥下分别对应的加密算法之间的关系，这样就有可能恢复出原密钥本身。低轮 AES-192 就是一个相关密钥攻击的实例[27]。

侧信道攻击是由 Paul Kocher 于 1996 年提出的[28]。它不同于上述三种密码分析法，侧信道攻击主要是利用密码算法运行过程中泄漏出的侧信道信息，如能量消耗、运行时间、电磁辐射信息等，通过对这些侧信道信息进行统计分析，进而恢复密钥信息。目前主要的侧信道攻击包括：能量分析、计时攻击、电磁分析、缓存攻击、故障攻击等等。

## 2.2 白盒密码

### 2.2.1 白盒密码概述

传统密码学中，密码算法的设计都是建立在黑盒模型的基础上，即假设密码算法的运行环境是安全的，攻击者只能访问算法的输入和输出，代码执行及动态加密不被观察，也无法获得密钥，保护好密钥是安全通信的关键。随着数字化信息的普遍应用，密码软件的运行环境不再单纯和可信。2002 年，Chow 等人在文献[6]中首先提出白盒攻击环境的概念。在白盒攻击环境中，假设攻击者可以访问同黑盒模型相同的资源，并对加密/解密软件完全控制，攻击者还可以观察动态密码运行过程，并且对内部的算法完全可见，可随意更改。因此, 传统的密码算法在白盒攻击环境中就显得极度脆弱，不再能够安全使用。

在白盒攻击环境的背景下，Chow 等人提出白盒密码的概念，它最初被定义为有下列目标的混乱技术：

**定义 2-1**[6]（Chow 等人）白盒密码　是一个混乱技术，它旨在以这样一种方式实现加密原语，即即使敌手可以访问密码实现的过程及其所执行的平台，也无法从中获取密码算法的密钥信息。

从上述定义描述可知，白盒密码利用混乱技术保护密码算法，虽然密码的执行过程发生了变化，但是其在功能上是等价的，唯一不同的是密钥信息不再像传统密码算法一样是可见的，而是被隐藏起来了。然而，定义 2-1 中有一个基本的问

题，也是从攻击者角度来讲应用程序中最薄弱的一环，那就是攻击者可以不直接提取密钥而是重用它，从而达到攻击的目的。这在数字版权管理中是一种常见的威胁。从某种意义上来说，白盒密码是一种隐藏密钥的技术，但是在这种攻击下，防止密钥被恢复就成了毫无意义的了。因此，对于白盒密码的目标，Wyseur 在其博士论文中给出了白盒密码的新定义：

**定义 2-2**[19]（Wyseur）白盒密码　是实现密码学功能的一种方式，这种方式可以这样描述：即使攻击者对整个算法的实现内容拥有完全的控制，也并不能使攻击者对同样密码算法进行黑盒攻击有任何的优势。

定义 2-2 相比于定义 2-1 来说所涵盖的范围更加全面，但是它也没有很明确的说明白盒密码的目标，只是扩大了原本只针对密钥保护的范围。Hohenberger 在文献[29]中从混乱的角度有类似的说法：

如果一个密码方案在黑盒模型下是安全的，那么这个经过混乱的密码算法仍然保持安全。

同时 Hohenberger 指出，并没有一个针对所有程序的通用混乱器，针对不同的密码算法需要设计不同的白盒密码方案。

目前，关于白盒密码的定义描述仍然比较模糊。在《软件加密与解密》一书中这样描述白盒密码：能用纯软件的方式把解密的密钥隐藏在软件中，同时还能正常进行解密工作？对于这个问题及相关问题的研究被统称为白盒加密。这类研究的目的是实现一种使攻击者即便在能够完全得到加/解密程序的源码并能运行它的情况下，也不能从中找出密钥的密码学原语。白盒密码的工作原理如下[1]：

（1）选择一种加密算法A，记其加密过程为 $E_k(m)=c$，解密过程为 $D_k(c)=m$，其中 $k$ 为对称加密/解密的密钥，$m$ 是明文，$c$ 是密文；

（2）选取一个密钥 $skey$；

（3）针对选定的密钥 $skey$ 和解密过程 $D$，构造一个白盒解密器 $D^{wb}(c)$，使得 $D^{wb}(c)=D_{skey}(c)$；

（4）对 $D^{wb}(c)$ 进行混淆，使 $skey$ 能安全隐藏在 $D^{wb}(c)$ 中，成为它不可分割的一部分。

目前的白盒密码技术的实例大部分都是基于已有密码算法的白盒设计，即根据现有的加密算法，将已知的密码算法通过白盒密码技术进行设计，利用算法中的某些特性来隐藏密钥，使得在白盒攻击环境下，不改变原算法的功能但能够达到白盒攻击环境下的安全，并且保持原算法的安全性不受破坏。白盒密码算法与传统密码算法有很大的区别，它其实是指一种新的密码算法，能够抵抗白盒攻击

环境下的攻击，而不是指单纯的在已有的密码算法上进行白盒设计。因此，在理解白盒密码概念的时候不可太狭隘。

### 2.2.2 白盒密码的设计方法

目前，常见的密码算法的白盒实现方式有三种：Chow 等人[6]的查找表方式、Bringer 等人[16]的插入扰乱项的方式、Biryukov 等人[30]的多变量密码的方式。

#### 2.2.2.1 查找表方式

查找表方式的主要思想是：采用混淆的思想隐藏密钥信息，使得密钥信息不会暴露在密码算法执行过程中。一种常用的混淆方法是置乱编码（可简称为编码），具体做法是：对一个已有的分组密码，选定一个密钥，然后对明文到密文的映射进行置乱编码，将置乱后的映射做成查找表的形式，这样密码算法的计算过程就转化为了查找表格的过程，整个密码算法就转化为了一系列查找表的过程。由于密钥被隐藏在查找表中，攻击者只能知道查找表的输入对应的输出，隐藏在内部的对应关系及密钥信息无法获知。

置乱编码的定义如下：

**定义 2-3**[6] 置乱编码　设 $X$ 是一个 $m$ 比特到 $n$ 比特的变换，任意选定一个 $m$ 比特到 $m$ 比特的双射 $F$ 和一个 $n$ 比特到 $n$ 比特的双射 $G$。把 $X'=G\circ X\circ F$ 看成 $X$ 的一个置乱编码形式，其中 $F$ 为输入置乱编码，$G$ 为输出置乱编码。

基于查找表方式设计白盒密码算法的实例有：Chow 等人的白盒 AES 算法[6]、Chow 等人的白盒 DES 算法[13]、Xiao-Lai 的白盒 AES 算法[18]、肖雅莹等人的白盒 SMS4 算法[7]等。Chow 等人的白盒 AES 算法（简称 Chow 白盒 AES）是采用查找表方式实现白盒密码的典型代表。Chow 白盒 AES 的设计思想是：通过将原有的 AES 算法转换成一系列的、相互独立的查找表操作，再对查找表进行外部编码和内部编码操作，编码后的查找表内容会被隐藏，从而实现对密钥信息的隐藏、混乱和扩散；同时利用两个相邻查找表的输入置乱编码与输出置乱编码在两两结合以后会相互抵消的特性，以此保证密码算法的完整性和可用性。

内部编码及外部编码的定义描述如下：

**定义 2-4** 内部编码　是指将随机置乱项结合到查找表中，在算法的每一轮内部或轮与轮之间进行的置乱编码，其目的是对密钥信息进行混淆/隐藏，防止攻击者通过查找表推出密钥信息。

**定义 2-5** 外部编码　有时攻击者的目的并不是想要得到密钥，而是为了重用

它，这时他就根本没必要把密钥提取出来，只要把白盒加密的实现代码整个嵌入到他的程序中即可。针对这种可能被攻击者利用的漏洞，Chow 等人建议使用外部编码。外部编码是在分组密码的第一轮输入之前以及最后一轮的输出之后加入的置乱编码。一般外部编码作用在整个程序的输入输出之上。

查找表方式设计白盒算法的方法将原始密码算法的计算过程转化为查找表过程以及少量的数学运算的过程，在白盒攻击环境中具有一定的安全性。然而，按照查找表方式设计的密码算法容易受到一种被称为 BGE 攻击方法的攻击，该攻击方法主要利用输入编码与输出编码的可逆关系这个弱点将相邻部分结合，进而抵消其中某些置乱混淆作用。因此，在利用查找表方式设计白盒算法时，要注意避免密码算法受到 BGE 的攻击。总体来说，查找表方式的白盒设计方式在实现效率、安全性和性能上有较好的平衡，可以在实际环境中酌情采用。

#### 2.2.2.2 插入扰乱项方式

插入扰乱项方式的主要思想是：将分组密码的每一轮运算看作是一个有限域上的多项式，使用 IP 问题（Isomorphism of Polynomials，多项式的同构）[31]，在原始的方程中增加额外的扰乱项，扰乱原方程的代数结构，从而使得针对代数结构的攻击变得困难。

基于插入扰乱项的方式设计白盒密码算法的代表算法是 Bringer 等人提出的 AES 白盒算法。Bringer 等人的白盒 AES 算法的特点是，在第一轮之前加入一些特定的项作为混乱信息，而这些项只能在最后一轮被抵消，这些额外的混乱信息会在运行过程中增加冗余，以此扰乱攻击者。最终，这些扰乱信息利用 $\tilde{0}$ 多项式[16]以及投票机制[16]来消除。

插入扰乱项方式设计白盒算法的方法是在有限域上的多项式的角度上，添加额外的混乱信息，更加提高了置乱信息的隐蔽性，在很大程度上提高了攻击难度。但是，由于在白盒实现的最后一轮，需要运行多个实例实现投票机制，因此，基于 Bringer 的插入扰乱项方式的白盒算法需要占用更多的额外空间，运行时间也会延长，其可用性及可扩展性也比较弱。

#### 2.2.2.3 多变量方式

多变量密码的方式是 Alex Biryukov 等人[30]于 2014 年提出的一种新的白盒密码设计方式。它并不是对已有的某种密码算法做白盒设计，而是一种基于 ASASA（Affine-Sbox）结构的通用白盒密码设计。

Alex 等人在文献[30]中提到的白盒设计分为两类：强白盒密码设计与弱白盒密

码设计。强白盒密码设计的主要思想是：使用有限域上的多变量多项式的方法，以$\chi$-scheme 和扩展 S 盒来分别实现 ASASA 结构中的 S 盒，这样可以得到两种不同的 ASASA 结构的白盒方案，用插入扰乱项的方式避免这两种方案受到攻击。

弱白盒密码设计也有两种方案，均采用查找表的方式实现。第一种设计方案是对单一 ASASA 结构的对称加密做白盒设计。第二种设计方案是对一个 SPN 结构的对称加密做白盒设计，这种方案的 S 层（代换层）是由多个第一种方案中构造的 ASASA 或者单一的 S 盒组成，每一个 ASASA 模块或者 S 盒都使用一个查找表来实现。

Alex 等人提出的这四种方案，目前还没有有效的攻击方法。

### 2.2.3 白盒密码安全性分析方法

目前，白盒密码还没有一个统一的理论基础，如何衡量一个白盒密码实现的安全性也没有严格的定义和统一的标准，对这方面的研究也是白盒密码理论基础的重点研究问题。目前白盒密码的安全性评价指标主要有：本地安全性、白盒多样性和白盒含混度、混乱度。然而，这几个度量单位并没有从密码学理论上给出明确的安全性证明，其是否可以准确的反映白盒实现的安全性还有待验证。因此，这些衡量方式只具有参考价值。下面具体介绍这几个衡量标准。

#### 2.2.3.1 本地安全性

Chow 等人提出的白盒密码的实现的核心就是含有混乱的查找表方式，也就是说，S 盒经过置乱编码后，攻击者将不能从中提取出任何有用的信息。这被称为本地安全性。对 S 盒进行变换编码并用查找表表示，每个独立的查找表都是本地安全的，例如一个与密钥相关的 $n$ 比特到 $m$ 比特的变换函数 $F_{n\to m}$：

$$F_{n\to m}: GF_{2^n} \to GF_{2^m} \tag{2-1}$$

用两个随机的双射矩阵 $P$ 和 $Q$ 对 $F_{n\to m}$ 进行编码，$F_{n\to m}' = P \circ F_{n\to m} \circ Q$，其中，$P$ 和 $Q$ 分别是 $n$ 比特到 $n$ 比特和 $m$ 比特到 $m$ 比特的双射。那么 $F_{n\to m}'$ 不会泄露关于 S 盒的信息。

#### 2.2.3.2 白盒多样性和白盒含混度

Chow 等人在文献[6]中提出了评价白盒 AES 算法的安全性的两个指标：白盒多样性（White-Box Diversity）和白盒含混度（White-Box Ambiguity）。这两个指标同样适用于类似的采用查找表方式设计的白盒密码算法中。

**定义 2-6** 白盒多样性　白盒多样性是指查找表所有可能的构造方法的个数，它取决于输入输出置乱编码的选择种数和置乱编码的步骤数。若查找表有 $n$ 个步骤，每个步骤有 $c_i$ 种选择，那么查找表的白盒多样性为 $\prod c_i$ 。

因为不同的输入输出置乱编码或者密钥可能生成相同的查找表，所以计算出的白盒多样性的值可能远大于查找表的实际个数。理论上来讲，白盒多样性的值越大，攻击者就越难分析出具体的输入输出置乱编码及隐藏在查找表中的密钥信息。

**定义 2-7** 白盒含混度　白盒含混度是衡量同一个白盒实现中有多少种不同的构造方法会产生相同的查找表的一个指标，它的值等于白盒多样性的值除以实际个数。

如果白盒含混度的值为 1，说明一个查找表对应唯一的具体的置乱混淆方式，则当攻击者得到查找表的时候，他们就比较容易知道该查找表所对应的输入输出置乱编码和密钥。因此，白盒含混度的值越大，一个查找表对应的输入输出编码方式越多，攻击者就越难推断出隐藏的置乱编码和密钥。

#### 2.2.3.3 混乱度

2011 年，王冰在其硕士学位论文[32]中提出一种改进的 AES 白盒实现方法，该方法是在 Chow 等人的查找表的基础上结合插入扰乱项方式，构造出的一个新的白盒 AES 算法。同时，他还将对原密码算法的编码类比为分组密码设计中的扩散原则，将加入的混乱项类比为混淆原则，并提出了衡量白盒实现的安全性的另一个指标：混乱度，其定义如下：

**定义 2-8**[32]　混乱度　在白盒环境中，白盒实现的一轮中不同的编码 $G$ 以及不同混乱的代数表示项 $\alpha$ 中的次数称为混乱度，最低的次数称为该白盒实现的混乱度。

理论上来讲，白盒实现的混乱度越高，安全性越高。同时他也给出结论，增加扩散度比增加混乱项所带来的效率要高。因此，在设计新的白盒密码以及设计新的易于白盒化的分组密码时可以将此结论作为参考。

## 2.3 密码算法评估

一般说来，评估一个密码算法的优劣可从安全性和实现效率两方面来进行。

### 2.3.1 密码算法的安全性评估

评价密码算法的安全性有以下几种方式：

**无条件安全**：原理上不可破译，无论截获多少密文，花费多少时间，都不能解密密文。

Shannon 指出要达到无条件安全，必须使密钥至少和明文一样长（即一次一密）[33]。

**计算安全**：如果被加密信息的价值远小于破译密码要付出的代价，或者破译密码的时间超过该信息的有效生命期，则它在计算上是安全的。

**可证明安全**：如果某个密码方案（或签名方案等）的安全性可以采用形式化的方法来证明，则它是可证明安全的。

**抵抗现有攻击的能力**：研究算法抵抗现有攻击的能力是目前评估分组密码安全性的主要手段。算法应能很好的抵抗现有的所有分析方法，如穷举密钥搜索攻击、差分密码分析、线性密码分析、相关密钥攻击、插值分析、代数攻击、能量攻击、故障攻击等等。

**实际安全性**：密码算法的实际安全性可细分为：理论上破译、设计主张的破译和实际应用的破译。如果存在比强力攻击更有效的攻击方法，则称为理论上被破译；如果存在比设计者主张的更有效的攻击，则称为设计主张的被破译；如果存在攻击，其复杂度比实际应用更低，则称为实际应用的可破译。

### 2.3.2 密码算法的实现效率评估

密码算法的实现效率是指密码算法在特定应用环境中，软件或者硬件设计的物理实现的效率[34]。密码算法实现的性能差异主要是不同的物理实现方式引起的。根据具体的应用环境设计的密码算法会表现出更高的算法实现效率。例如，对于 32 位平台加密算法能更好的适应 32 位处理器平台，而在 16 位或 8 位处理器平台其实现效率可能会差一些，反之亦然。

衡量密码算法实现效率的具体指标包括：密码算法在软件或者硬件平台上的运行时间以及算法程序占用的存储空间。由于不同的密码算法所包含的运算结构单元不同，因此，它在不同的平台上耗费的资源不同，消耗的运行时间也不同。

## 2.4 本章小结

本章介绍了白盒密码设计相关的密码学基本概念和白盒密码基础知识。

在白盒攻击环境中，传统的密码算法已无法保证所需的安全性。因此，需要设计能够抵抗白盒攻击的密码算法，这种密码被称为白盒密码。

白盒密码的设计方法主要有三种：查找表方式、插入扰乱项的方式以及多变量密码的方式。其中，以Chow的查找表方式最具代表性，它的主要思想是：对现有的一个分组密码，选定密钥，采用置乱编码的方式对明文到密文的映射进行混淆，把密码算法转换为查找表，最终一系列表格查找的过程就替换了原密码算法的计算过程。

白盒密码的安全性评估方法主要有：本地安全性、白盒多样性、白盒含混度以及白盒混乱度。其中，本地安全性、白盒多样性和白盒含混度对于含有混乱的查找表方式的白盒密码算法适用，白盒混乱度适用于查找表方式与插入混乱项方式相结合的白盒密码算法。

# 第三章 白盒 SMS4 密码算法

本章介绍白盒 SMS4 密码算法。首先介绍 SMS4 算法，描述 SMS4 算法的整体结构、轮函数、加/解密算法以及密钥扩展算法；然后介绍肖-白盒 SMS4 密码的设计方法，并对肖-白盒 SMS4 算法的执行效率进行了分析；同时，对肖-白盒 SMS4 算法的安全性进行分析，包括白盒多样性及白盒含混度的分析、抵抗 BGE 方法的分析以及林婷婷等人针对肖-白盒 SMS4 的一种有效攻击方法。

## 3.1 SMS4 算法

SMS4算法是我国官方公布的第一个分组密码算法，被用于无线局域网产品设备中。2013 年 6 月，SM4 算法被国家密码管理局正式确定为国家密码行业标准标准，它也就是原来的SMS4算法。

### 3.1.1 算法整体结构

SMS4算法是典型的 Feistel 结构，其明文分组长度和密钥长度均为 128 比特，加密后的密文分组也是 128 比特，加密过程为 32 轮迭代操作及最后一轮输出的反序变换，每一轮的迭代操作称为轮函数。其解密过程类似于加密过程，但是轮密钥使用了与加密轮密钥相反的顺序。轮密钥的产生类似于加解密算法，除了轮函数中循环左移操作部分不同外，其他都相同。SMS4算法的整体结构如图 3-1 所示。

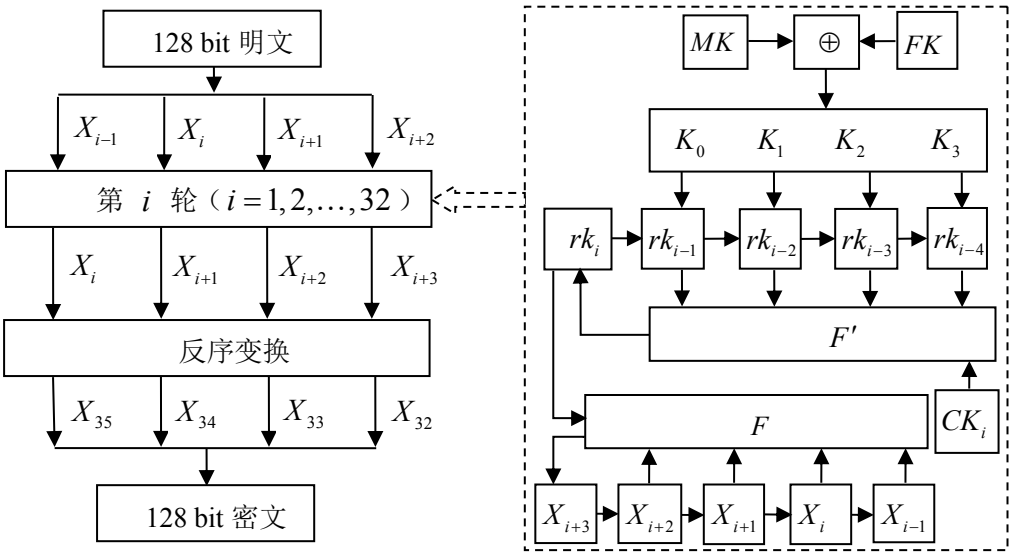


图 3-1 SMS4 算法整体结构图

算法中用到的运算及符号表示如下：

1.基本运算

（1）⊕：32 比特异或

（2）<<< $i$：32 比特循环左移 $i$ 位

2.S 盒

S 盒为固定的 8 比特输入 8 比特输出的置换，用 $Sbox(\bullet)$ 表示。S 盒中的数据都是以十六进制的形式表示，其置换规则是：以输入的前半字节为行号，后半字节为列号，行列交叉处的数据即为输出。具体变换规则如图 3-2 所示。

3.密钥及密钥参量

用 $MK=(MK_0,MK_1,MK_2,MK_3)$ 表示加密密钥，其中 $MK_i(i=0,1,2,3)$ 为字，加密密钥长度为 128 比特。

用 $(rk_0,rk_1,\ldots,rk_{31})$ 表示轮密钥，其中 $rk_i(i=0,1,\ldots,31)$ 为字，由加密密钥生成轮密钥。

$FK=(FK_0,FK_1,FK_2,FK_3)$ 为系统参数，$CK=(CK_0,CK_1,\ldots,CK_{31})$ 为固定参数，用于密钥扩展算法，其中 $FK_i(i=0,1,2,3)$ 、$CK_i(i=0,1,\ldots,31)$ 为字。

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | d6 | 90 | e9 | fe | cc | e1 | 3d | b7 | 16 | b6 | 14 | c2 | 28 | fb | 2c | 05 |
| 1 | 2b | 67 | 9a | 76 | 2a | be | 04 | c3 | aa | 44 | 13 | 26 | 49 | 86 | 06 | 99 |
| 2 | 9c | 42 | 50 | f4 | 91 | ef | 98 | 7a | 33 | 54 | 0b | 43 | ed | cf | ac | 62 |
| 3 | e4 | b3 | 1c | a9 | c9 | 08 | e8 | 95 | 80 | df | 94 | fa | 75 | 8f | 3f | a6 |
| 4 | 47 | 07 | a7 | fc | f3 | 73 | 17 | ba | 83 | 59 | 3c | 19 | e6 | 85 | 4f | a8 |
| 5 | 68 | 6b | 81 | b2 | 71 | 64 | da | 8b | f8 | eb | 0f | 4b | 70 | 56 | 9d | 35 |
| 6 | 1e | 24 | 0e | 5e | 63 | 58 | d1 | a2 | 25 | 22 | 7c | 3b | 01 | 21 | 78 | 87 |
| 7 | d4 | 00 | 46 | 57 | 9f | d3 | 27 | 52 | 4c | 36 | 02 | e7 | a0 | c4 | c8 | 9e |
| 8 | ea | bf | 8a | d2 | 40 | c7 | 38 | b5 | a3 | f7 | f2 | ce | f9 | 61 | 15 | a1 |
| 9 | e0 | ae | 5d | a4 | 9b | 34 | 1a | 55 | ad | 93 | 32 | 30 | f5 | 8c | b1 | e3 |
| a | 1d | f6 | e2 | 2e | 82 | 66 | ca | 60 | c0 | 29 | 23 | ab | 0d | 53 | 4e | 6f |
| b | d5 | db | 37 | 45 | de | fd | 8e | 2f | 03 | ff | 6a | 72 | 6d | 6c | 5b | 51 |
| c | 8d | 1b | af | 92 | bb | dd | bc | 7f | 11 | d9 | 5c | 41 | 1f | 10 | 5a | d8 |
| d | 0a | c1 | 31 | 88 | a5 | cd | 7b | bd | 2d | 74 | d0 | 12 | b8 | e5 | b4 | b0 |
| e | 89 | 69 | 97 | 4a | 0c | 96 | 77 | 7e | 65 | b9 | f1 | 09 | c5 | 6e | c6 | 84 |
| f | 18 | f0 | 7d | ec | 3a | dc | 4d | 20 | 79 | ee | 5f | 3e | d7 | cb | 39 | 48 |

例：S 盒输入为‘a7’，则变换后的值为表中第 a 行第 7 列的值，$Sbox('a7')='60'$。

图 3-2 SMS4 算法的 S 盒

### 3.1.2 轮函数

SMS4 算法采用非线性迭代结构，每一次的迭代运算称为一轮变换。轮函数算法过程如图 3-2 所示：

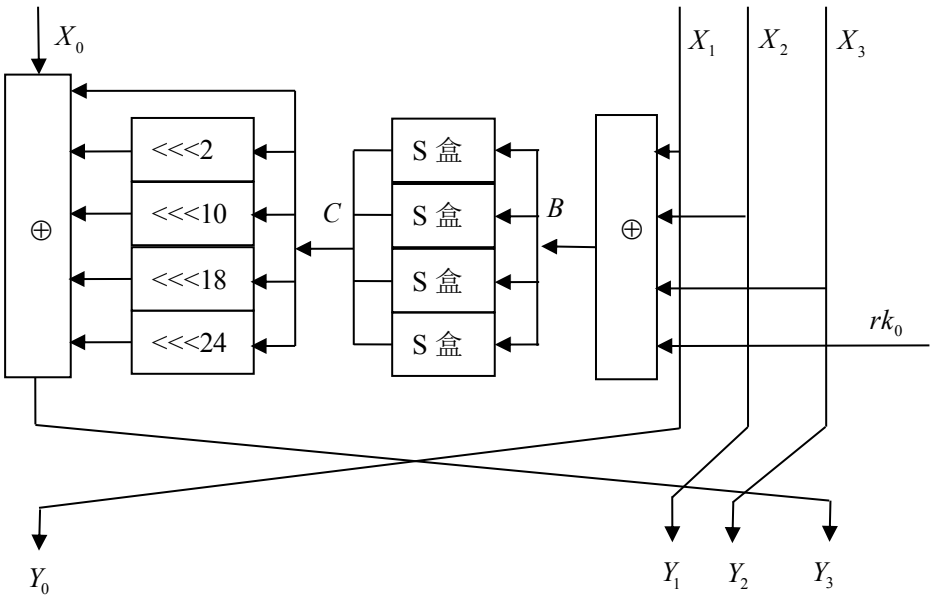


图 3-3 轮函数算法流程

设输入为$(X_0,X_1,X_2,X_3)$，轮密钥为$rk_0$，则轮函数$F$为：

$$F(X_0,X_1,X_2,X_3,rk_0)=X_0\oplus T(X_1\oplus X_2\oplus X_3\oplus rk_0) \tag{3-1}$$

$T$由非线性变换$\tau$和线性变换$L$复合而成，即

$$T(x)=L(\tau(x)) \tag{3-2}$$

非线性变换$\tau$由四个并行的 S 盒构成，若$A=(a_0,a_1,a_2,a_3)$，则

$$\tau(A)=(S_{box}(a_0),S_{box}(a_1),S_{box}(a_2),S_{box}(a_3)) \tag{3-3}$$

线性变换$L$定义为：

$$L(X)=X\oplus(X<<<2)\oplus(X<<<10)\oplus(X<<<18)\oplus(X<<<24) \tag{3-4}$$

### 3.1.3 加/解密算法

设$(X_0,X_1,X_2,X_3)$为明文输入，$(Y_0,Y_1,Y_2,Y_3)$为密文输出，$rk_i(i=0,1,\ldots,31)$为轮密钥。则 SMS4 算法的加密变换可以表示为：

$$\begin{aligned}X_{i+4}&=F(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i)\\&=X_i\oplus T(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i),i=0,1,\ldots,31\end{aligned} \tag{3-5}$$

$$\left(Y_0,Y_1,Y_2,Y_3\right)=R\left(X_{32},X_{33},X_{34},X_{35}\right)=\left(X_{35},X_{34},X_{33},X_{32}\right) \tag{3-6}$$

其中，$R$ 为反序变换：$R\left(X_0,X_1,X_2,X_3\right)=\left(X_3,X_2,X_1,X_0\right)$。

解密变换的轮密钥采用加密变换顺序相反的轮密钥，加密时轮密钥的使用顺序为：$\left(rk_0,rk_1,\ldots,rk_{31}\right)$，解密时轮密钥的使用顺序为：$\left(rk_{31},rk_{30},\ldots,rk_0\right)$。

### 3.1.4 密钥扩展算法

SMS4 算法的密钥扩展算法将加密密钥生成轮密钥。128 比特的加密密钥用 $MK=\left(MK_0,MK_1,MK_2,MK_3\right)$ 表示。轮密钥的生成方法如图 3-4 所示：

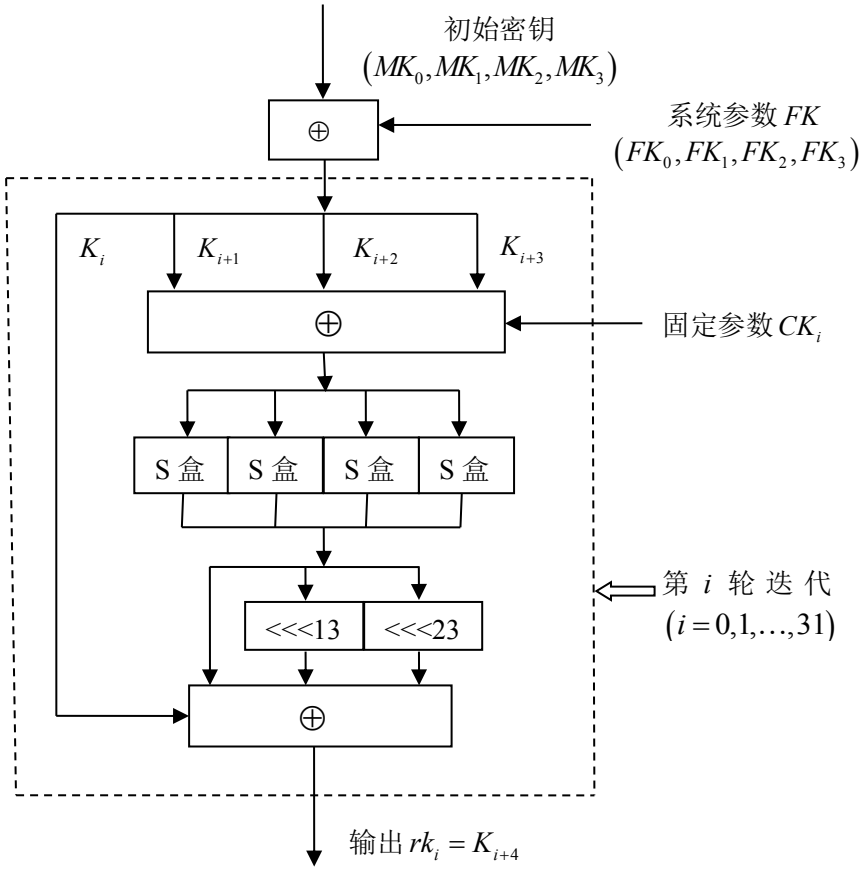


图 3-4 密钥扩展算法

$$\left(K_0,K_1,K_2,K_3\right)=\left(MK_0\oplus FK_0,MK_1\oplus FK_1,MK_2\oplus FK_2,MK_3\oplus FK_3\right) \tag{3-7}$$

$$rk_i=K_{i+4}=K_i\oplus T'\left(K_{i+1}\oplus K_{i+2}\oplus K_{i+3}\oplus CK_i\right),i=0,1,\ldots,31 \tag{3-8}$$

其中：

1. $T'$ 变换与加密算法轮函数中的 $T$ 变换仅其中线性变换 $L'$ 不同。$T'$ 变换的线性变换 $L'$ 为：

$$L'(B)=B\oplus(B<<<13)\oplus(B<<<23) \tag{3-9}$$

2.系统参数 $FK$ 取值依次为：$0xa3b1bac6$，$0x56aa3350$，$0x677d9197$，$0xb27022dc$。

3.32 个固定参数 $CK_i$ 的取值依次为：$0x00070e15$，$0x1c232a31$，$0x383f464d$，$0x545b6269$，$0x70777e85$，$0x8c939aa1$，$0xa8afb6bd$，$0xc4cbd2d9$，$0xe0e7eef5$，$0xfc030a11$，$0x181f262d$，$0x343b4249$，$0x50575e65$，$0x6c737a81$，$0x888f969d$，$0xa4abb2b9$，$0xc0c7ced5$，$0xdce3eaf1$，$0xf8ff060d$，$0x141b2229$，$0x30373e45$，$0x4c535a61$，$0x686f767d$，$0x848b9299$，$0xa0a7aeb5$，$0xbcc3cad1$，$0xd8dfe6ed$，$0xf4fb0209$，$0x10171e25$，$0x2c333a41$，$0x484f565d$，$0x646b7279$。

## 3.2 肖-白盒 SMS4 算法

2009 年，肖雅莹和来学嘉在密码学年会上提出了 SMS4 算法的白盒实现，本文将其简称为“肖-白盒 SMS4 算法”。它的基本思想是：把每一轮 SMS4 算法拆分成三小块，对每个小块进行置乱编码，并将每一轮中的某些步骤合成一个查找表，再利用可逆仿射变换作为输入置乱编码和输出置乱编码对其混淆，将密钥信息隐藏在查找表中防止攻击者获得密钥信息。整个算法的过程转化为计算仿射变换和查找表的过程。

### 3.2.1 肖-白盒 SMS4 算法描述

肖-白盒 SMS4 算法是在 SMS4 算法基础上的白盒设计，其明文分组长度和密钥长度均为 128 比特，加密后的密文分组也是 128 比特，加密过程为 32 轮迭代过程及最后一轮输出的反序变换组成，每一轮操作又分为三个部分 Part 1、Part 2 和 Part 3。整体的肖-白盒 SMS4 算法结构图如图 3-5 所示。

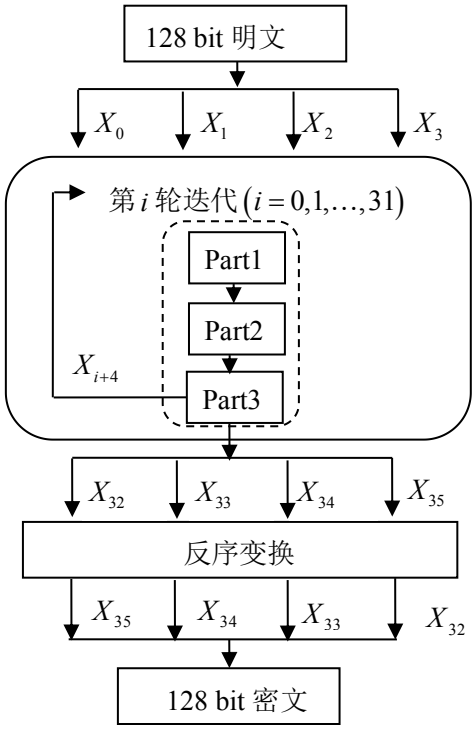


图 3-5 肖-白盒 SMS4 算法整体结构图

每一轮白盒 SMS4 的实现步骤如下：

步骤一：计算 $X = X_{i+1} \oplus X_{i+2} \oplus X_{i+3}$

因为采用网络化编码方式，即上一变换的输出编码与下一变换的输入编码相抵消，因此每个变换的前后都需要进行输入置乱编码与输出置乱编码，每一轮的输出是经过输出置乱编码的，所以对每一轮的输入要进行置乱处理，消去上一个变换的输出置乱编码，再进行计算。

因此，先要对第 $i$ 轮的输入 $X_{i+1}$、$X_{i+2}$、$X_{i+3}$ 进行输入置乱处理，消去上一轮变换中的输出置乱编码，然后进行输出置乱编码。整个步骤一的过程（Part 1）如图 3-6 所示。

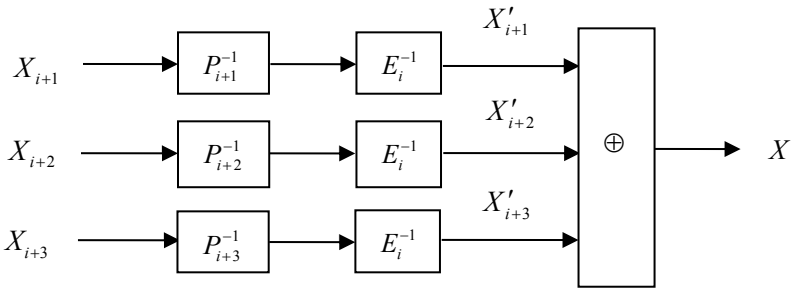


图 3-6 肖-白盒 SMS4 算法的 Part 1 部分

在肖的白盒SMS4算法中，置乱编码均为可逆仿射变换的形式，其数学表达式可描述为：$P(x)=l[P](x)\oplus c[P]$。其中$P$代表一个仿射变换，$l[P]$是可逆矩阵，是$P$的线性部分，$c[P]$是列向量形式，是$P$的常数项。

这一步骤的计算过程可以表示成如下形式：

$$\begin{aligned} X &= X'_{i+1}\oplus X'_{i+2}\oplus X'_{i+3} \\ &= E_i^{-1}\circ P_{i+1}^{-1}(X_{i+1})\oplus E_i^{-1}\circ P_{i+2}^{-1}(X_{i+2})\oplus E_i^{-1}\circ P_{i+3}^{-1}(X_{i+3}) \end{aligned} \tag{3-10}$$

其中，$P_{i+j}{}^{-1}$与$E_i^{-1}$都是可逆仿射变换，$P_{i+j}(x)=A_{i+j}(x)\oplus a_{i+j}$，$A_{i+j}$为$GF(2)$上的$32\times 32$可逆矩阵，$a_{i+j}$是32 比特的列向量；$E_i=diag(E_{i0},E_{i1},E_{i2},E_{i3})$，每个$E_{ij}$均为$GF(2)$上 8 比特到 8 比特的可逆仿射变换。$P_{i+j}$与$E_i$都是随机选择并且保密的，只需保存$M_{i+j}{}^{i}=E_i^{-1}\circ P_{i+j}{}^{-1}$，这样就转化为了只有一个 32 比特到 32 比特的仿射变换了。

整个步骤一由三个仿射变换和两个异或计算构成。

步骤二：对$T$变换做输入输出置乱编码，形成查找表，将密钥隐藏在查找表中；

在每一轮SMS4的加密算法中，

$$\begin{aligned} &T(X\oplus rk_i) \\ &= L(\tau(X\oplus rk_i)) \\ &= L(S_{box}(x_{i0}\oplus rk_{i0}),S_{box}(x_{i1}\oplus rk_{i1}),S_{box}(x_{i2}\oplus rk_{i2}),S_{box}(x_{i3}\oplus rk_{i3})) \end{aligned} \tag{3-11}$$

其中，$X=X_{i+1}\oplus X_{i+2}\oplus X_{i+3}=(x_{i0},x_{i1},x_{i2},x_{i3})$，$rk_i=(rk_{i0},rk_{i1},rk_{i2},rk_{i3})$。

由于 S 盒是公开的，如果已知$S_{box}(x\oplus k_{ij})$和$x_{ij}$，则不难得出轮密钥$rk_{ij}, j=0,1,2,3$。因此对$T$变换分别做输入和输出置乱编码（Part 2），如图 3-7 所示。图中，$E_{ij}$是 8 比特到 8 比特的可逆仿射变换，$Q_i$是 32 比特到 32 比特的可逆仿射变换。

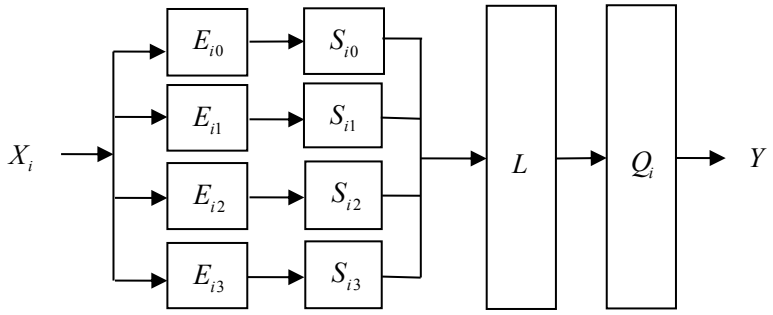


图 3-7 肖-白盒 SMS4 算法的 Part 2 部分

由于这一部分是输入 32 比特输出 32 比特的变换，如果做成一张查找表的话，其所占空间为 $2^{32}\times 32=16\,\text{GB}$，这在实际中并不适用。因此，将整个变换拆分成四个小的查找表。

将 $X=(x_{i0},x_{i1},x_{i2},x_{i3})$ 经过 $E_{ij}$ 和 $S_{ij}$ 变换后的值记为 $(z_{i0},z_{i1},z_{i2},z_{i3})$，$x_{ij}$ 与 $z_{ij}$（$j=0,1,2,3$）一一对应。

$$
\begin{aligned}
Y &= Q_i \circ L\cdot\begin{bmatrix} z_{i0}\\ z_{i1}\\ z_{i2}\\ z_{i3}\end{bmatrix} = l[Q_i]\cdot\left(L\cdot\begin{bmatrix} z_{i0}\\ z_{i1}\\ z_{i2}\\ z_{i3}\end{bmatrix}\right)\oplus c[Q_i] = (R_{i0},R_{i1},R_{i2},R_{i3})\cdot\begin{bmatrix} z_{i0}\\ z_{i1}\\ z_{i2}\\ z_{i3}\end{bmatrix}\oplus c[Q_i]\\
&= (R_{i0}\cdot z_{i0})\oplus(R_{i1}\cdot z_{i1})\oplus(R_{i2}\cdot z_{i2})\oplus(R_{i3}\cdot z_{i3}\oplus c[Q_i])\\
&= v_{i0}\oplus v_{i1}\oplus v_{i2}\oplus v_{i3}
\end{aligned}
\tag{3-12}
$$

其中，$R_{ij}, j=0,1,2,3$ 是 $32\times 8$ 的矩阵，$l[Q_i]$ 是 $Q_i$ 的线性部分，$c[Q_i]$ 是 $Q_i$ 的常数项。由上式可以看出，$z_{ij}$ 与 $v_{ij}$（$j=0,1,2,3$）也是一一对应，所以 $x_{ij}$ 与 $v_{ij}$（$j=0,1,2,3$）也是一一对应的，这样整个变换可以划分成 4 个 8 bit 到 32 bit 的查找表。计算 Part2 时，先进行查找表操作，共四次，然后对这 4 个结果做异或操作，就能得到本次变换的输出 $Y$。

步骤三：计算 $X_{i+4}$。

同理，这一部分的计算也要进行输入输出置乱编码，$X_{i+4}$ 的计算过程（Part 3）如图 3-8 所示。

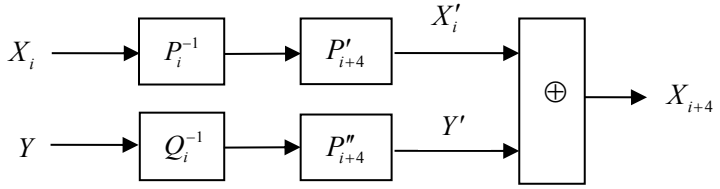


图 3-8 肖-白盒 SMS4 算法的 Part 3 部分

图 3-8 中，$P'_{i+4}(x)=P_{i+4}(x)\oplus a'_{i+4}$，$P''_{i+4}(x)=P_{i+4}(x)\oplus a''_{i+4}$，均为 $GF(2)$ 上 32 bit 到 32 bit 的可逆仿射变换，他们将与下一轮中对 $X_{i+4}$ 的置乱编码抵消一部分；$Q_i^{-1}$ 与步骤二中的 $Q_i$ 相抵消。

将 $P'_{i+4}\circ P_i^{-1}$ 的作用结果记为 $D_i$，将 $P''_{i+4}\circ Q_i^{-1}$ 的作用结果记为 $C_i$，则 $D_i$ 和 $C_i$ 均为 32 bit 到 32 bit 的仿射变换。整个步骤三的计算过程包括两个仿射变换和一个异或。

总的来说，肖-白盒SMS4算法的每一轮分为三个部分：第一部分使用 3 个仿射变换 $M_{i+j}^{\ i}$ 和两个异或计算，第二部分查找 4 个 8 比特到 32 比特的查找表和 3 个异或计算，第三部分使用两个仿射变换 $D_i$ 与 $C_i$ 和一个异或计算。每个部分都采用仿射变换进行输入输出编码，密钥隐藏在查找表中，即使敌手可以读取内存，也无法获得密钥信息。

### 3.2.2 肖-白盒 SMS4 的实现复杂度分析

肖-白盒 SMS4 算法是基于查找表方式的，算法执行过程可以通过仿射变换和查找表来完成，其算法复杂度可以用额外占用的空间以及查表、计算异或、仿射变换的次数来衡量。肖-白盒 SMS4 算法每一轮过程划分为三个部分，整个白盒 SMS4 算法的执行需要 128 次表格查找以及 160 次 32 比特到 32 比特的仿射变换。

肖-白盒 SMS4 的每一轮所占用的空间为：

Part 1 包括 3 个 32 比特到 32 比特的仿射变换：$3\times(32\times32+32)$=3168（比特）；

Part 2 包括 4 个 8 比特到 32 比特的查找表：$4\times(2^8\times32)$=32768（比特）；

Part 3 包括 2 个 32 比特到 32 比特的仿射变换：$2\times(32\times32+32)=2112$（比特）；

因此，整个肖-白盒 SMS4 共占用的空间为：

$$(3168+32768+2112)\times32=1217536\text{ 比特}=152192\text{B}=148.625\text{KB}$$

表 3-1 是几种密码算法执行效率的比较。

表 3-1 几种密码算法执行效率的比较（一）

| 加密算法 | 占用空间 | 查表或异或计算次数 | 仿射变换次数 |
|---|---|---|---|
| AES-128 | 4352 B | 300 | 0 |
| Chow 白盒 AES | 752 KB | 3104 | 0 |
| SMS4 | 544 B | 256 | 64 |
| 肖-白盒 SMS4 | 148.625KB | 128 | 160 |

### 3.2.3 肖-白盒 SMS4 的安全性分析

#### 3.2.3.1 白盒多样性及白盒含混度

白盒密码的主要目的是防止密码算法中的密钥在算法执行过程中被攻击者抽取。由于白盒SMS4算法的密钥信息隐藏在查找表中，并进行了输入输出置乱编码，

因此，从查找表中恢复出密钥信息或者输入输出置乱编码的难度决定了白盒SMS4算法的安全程度。

白盒多样性指的是查找表所有可能的构造方法的种数，它与输入输出置乱编码有多少种选择以及置乱编码的步骤数有关。

$Z_n$ 上 $m$ 阶可逆矩阵的个数 $N_m(n)$ 可用下述定理计算：

**定理 3-1**[35]：设 $n \geq 2$ 为整数， $n = p_1^{r_1} p_2^{r_2} \cdots p_s^{r_s}$ 为 $n$ 的既约因子分解， $r_i \geq 1$，$p_1, p_2, \cdots, p_s$ 是互异素数，则 $N_m(n) = \prod_{i=1}^{s} \prod_{j=0}^{m-1} \left(p_i^m - p_i^j\right) p_i^{(r_i-1)m^2}$ 。

因此， $GF(2)$ 上 $m$ 阶可逆矩阵的个数为：

$$N_m(2) = \prod_{j=0}^{m-1} \left(2^m - 2^j\right)$$

$GF(2)$ 上 8 阶可逆矩阵的个数约为 $2^{62}$，32 阶可逆矩阵的个数约为 $2^{922}$ 。

在肖-白盒 SMS4 算法中，每一轮的白盒多样性的值如下：

Part 1： $\left(2^{922} \times 2^{32}\right)^3 \times \left(2^{62} \times 2^8\right)^4 = 2^{3142}$

Part 2： $\left(2^{62} \times 2^8\right)^4 \times 2^{32} \times \left(2^{922} \times 2^{32}\right) = 2^{1266}$

Part 3： $\left(2^{922} \times 2^{32}\right)^3 \times 2^{32} = 2^{2894}$

白盒含混度用来衡量有多少种不同的构造方法会产生相同的查找表，它等于白盒多样性的值除以实际的查找表的个数。在肖-白盒 SMS4 算法中，每一轮的白盒含混度的估计值如下：

Part 1： $\left(2^{62} \times 2^8\right)^4 = 2^{280}$

Part 2： $\left(2^{62} \times 2^8\right)^4 \times 2^{32} = 2^{312}$

Part 3： $2^{922} \times 2^{32} \times 2^{32} = 2^{986}$

从白盒多样性和白盒含混度来看，肖-白盒SMS4是安全的，攻击者很难从仿射变换或者查找表中推断出输入输出置乱编码及密钥信息。

#### 3.2.3.2 BGE 攻击

BGE攻击是由Billet 、Gilbert 和Ech-Chatbi提出的一种攻击方法，最初是针对Chow 白盒 AES 算法，此方法可以在时间复杂度为 $o\left(2^{30}\right)$ 以内获得其密钥。由于白盒设计本地安全性和多样性的特性，攻击单独的查找表很难获得密钥信息。不

过将多个查找表组合起来，就是完整的 AES 密码算法，最终可以得到每一轮操作的 4 个输入块对应 4 个输出块的查找表。由于白盒 AES 中的输出置乱编码与输入置乱编码存在互逆的关系，即若用 $r$ 表示白盒 AES 第 $r$ 轮的变换，$In^r$ 表示输入置乱编码，$Out^r$ 表示输出置乱编码，则 $Out^r=\left(In^{r+1}\right)^{-1}$，这样将白盒 AES 一轮中查找表组合在一起时内部编码将会抵消，只剩下外部编码的作用，再通过分析组成这个查找表的代数结构，就能获得密钥信息。这就是 BGE 攻击的主要思想。

虽然 BGE 攻击方法是针对 Chow 等设计的白盒 AES 提出的，但是基于查找表方式构造分组密码的白盒算法的本质与构造方法都类似，所以基于查找表方式设计的白盒 SMS4 算法在进行安全性评估时也要考虑其是否能够抵抗 BGE 攻击。

BGE 攻击方法能够攻破 Chow 的白盒 AES 的一个最重要的原因是它的每一轮的输出置乱编码和下一轮的输入置乱编码是互逆的，在相邻查找表进行组合时能够相互抵消。肖-白盒 SMS4 算法的查找表结构与 Chow 的白盒 AES 中的查找表结构类似，但是有一点不同：在肖-白盒 SMS4 算法中，$Out^r\neq\left(In^{r+1}\right)^{-1}$，它们之间相差一个常数，而攻击者却无法知道它，也无法从查找表或者仿射变换中得到。

如果将 Part 2 中的查找表和 Part 3 中的一部分仿射变换结合起来考虑，我们可以得到如图 3-9 所示的变换。

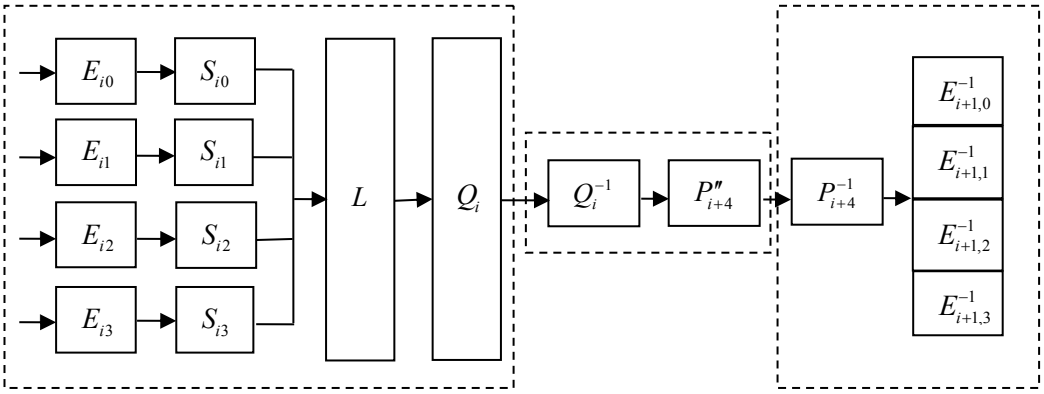


图 3-9 查找表和仿射变换结合的结构图

在图 3-9 中，$Q_i$ 和它的逆完全抵消，$P_{i+4}^{-1}$ 与 $P''_{i+4}$ 抵消后还剩下一个常数 $A_{i+4}^{-1}\cdot a''_{i+4}$。而这个常数是设计者在选择置乱参数 $P''_{i+4}$ 时确定的，攻击者无法得知，进而无法计算出隐藏在查找表中的密钥信息。因此，肖-白盒 SMS4 算法能够抵抗 BGE 攻击。

#### 3.2.3.3 对肖-白盒 SMS4 的一种有效攻击方法

虽然从白盒多样性和白盒含混度的角度，以及分析 BGE 攻击的角度，肖-白盒

SMS4 算法能达到所需的安全性，然而林婷婷等人却构造了一种有效攻击方法，能以低于 $o(2^{47})$ 的时间复杂度找出 SMS4 算法的轮密钥。该攻击方法利用 BGE 攻击方法与差分密码分析法，以及求解方程组等相结合的方法，其主要思想是：针对白盒 SMS4 的某一轮，首先将白盒SMS4算法中的Part 2和Part 3以及下一轮的Part 1组合起来，消去中间的部分输入输出置乱编码，得到关于某个仿射变换的一系列代数关系式，通过求解这些关系式，获得白盒 SMS4 方案中的某些仿射变换变量，进而得到轮密钥。

如图 3-10 所示，在白盒攻击环境下，肖-白盒 SMS4 算法对于攻击者可见的只有 $M_{i+j}^{i}$、$D_i$、$C_i$ 以及查找表，其他参数均不可见。密钥隐藏在查找表中，无法直接获得。分析 Part 2 部分（见图 3-7）的查找表，查找表的对应关系 $X \to Y$ 我们是知道的，S 盒与线性变换 $L$ 我们也是已知的，因此要获得密钥，我们需要求出变量 $E_i$ 和 $Q_i$，由于它们均是仿射变换，所以要分别求出其线性部分与常数项。

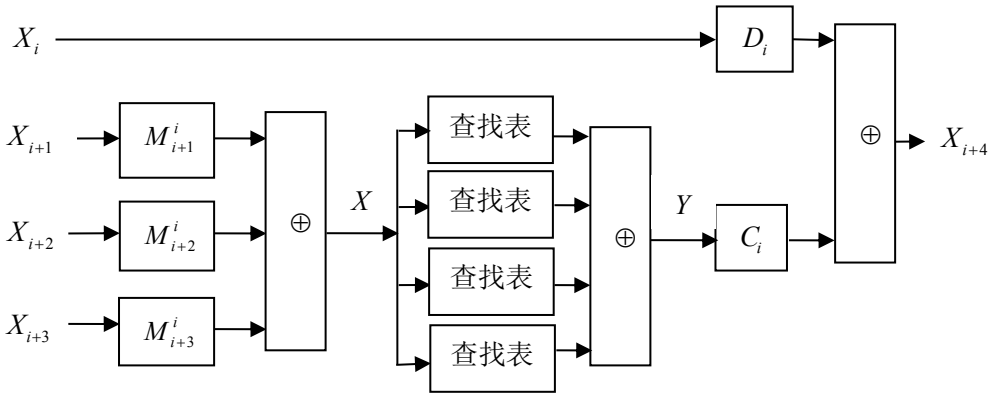


图 3-10 一轮肖-白盒 SMS4 算法

在介绍具体攻击步骤之前，这里先介绍一下仿射变换的一些性质。

**性质** 1:若可逆仿射变换的形式为：$P(x)=l[P](x)\oplus c[P]$，则此仿射变换的逆为：$P^{-1}(x)=(l[P])^{-1}(x)\oplus(l[P])^{-1}\cdot c[P]$。

**证明**：令仿射变换 $B$ 为：$B(x)=(l[P])^{-1}(x)\oplus(l[P])^{-1}\cdot c[P]$，则

$$\begin{aligned} B\circ P(x) &= B\left(l[P](x)\oplus c[P]\right) \\ &= (l[P])^{-1}\left(l[P](x)\oplus c[P]\right)\oplus(l[P])^{-1}\cdot c[P] \\ &= (l[P])^{-1}\cdot l[P](x)\oplus(l[P])^{-1}\cdot c[P]\oplus(l[P])^{-1}\cdot c[P] \\ &= E(x) \end{aligned}$$

这里，$E$ 表示单位阵。

由逆仿射变换的唯一性可知，仿射变换 $P$ 的逆变换即为 $B$，也即

$$P^{-1}(x)=\left(l[P]\right)^{-1}(x)\oplus\left(l[P]\right)^{-1}\cdot c[P]$$ ■

**性质 2**：仿射变换不满足分配律。确切的说是，偶数项的仿射变换不满足分配律。

**证明**：假设仿射变换 $P$ 为 $P(x)=l[P](x)\oplus c[P]$，则

$$P(x\oplus y)=l[P](x\oplus y)\oplus c[P]$$

$$\begin{aligned}P(x)\oplus P(y)&=\left(l[P](x)\oplus c[P]\right)\oplus\left(\left[l[P](y)\oplus c[P]\right]\right)\\&=l[P](x)\oplus l[P](y)\\&=l[P](x\oplus y)\end{aligned}$$

可知，偶数项的仿射变换不满足分配律。显然，奇数项的仿射变换满足分配律。 ■

下面介绍攻击的具体步骤。

**第一步：恢复 $E_{i+1}^{-1}$ 的线性部分**

首先利用类似 BGE 攻击方法对肖-白盒 SMS4 进行分析，合并 Part 2，Part 3 的一部分以及下一轮的 Part 1 的一部分，如图 3-11 所示，由于 $X_i'$ 是由仿射变换 $D_i$ 独立计算的，因此将 Part 3 中的一部分异或去掉是合理的。

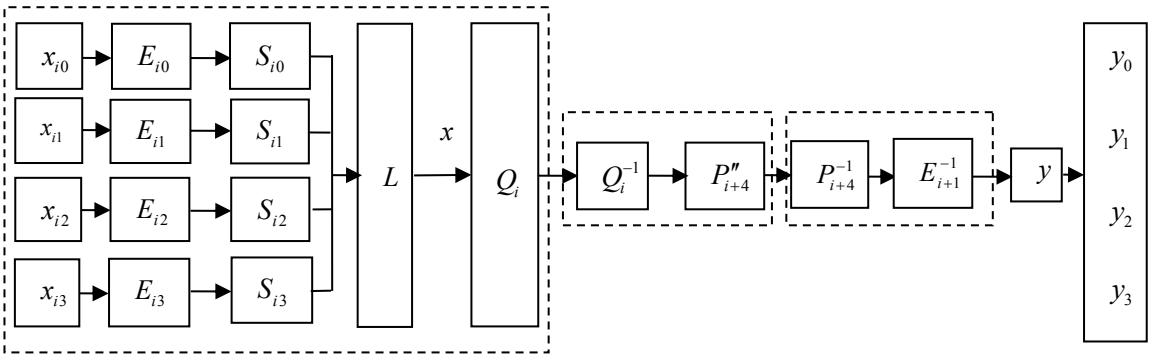


图 3-11 图解合并部分 Part 2，Part 3 下一轮的 Part 1

由图 3-11 可知，$Q_i$ 与 $Q_i^{-1}$ 完全抵消，而 $P''_{i+4}$ 与 $P_{i+4}^{-1}$ 相抵消剩下常数 $A_{i+4}^{-1}\cdot a''_{i+4}$。因此，经 $L$ 输出的数据 $x$，事实上只需与常数 $A_{i+4}^{-1}\cdot a''_{i+4}$ 异或，再经过 $E_{i+1}^{-1}$ 变换即可得到 $y$，整个过程可以表示为：$y=\tilde{E}_{i+1}(x)=E_{i+1}^{-1}\left(x\oplus A_{i+4}^{-1}\cdot a''_{i+4}\right)$。又因为 $E_{i+1}^{-1}=diag\left(E_{i+1,0}^{-1},E_{i+1,1}^{-1},E_{i+1,2}^{-1},E_{i+1,3}^{-1}\right)$，$E_{i+1,j}^{-1}$ 是 8 比特到 8 比特的仿射变换，可表示为

$E_{i+1,j}^{-1}(x)=l\left[E_{i+1,j}^{-1}\right](x)\oplus c\left[E_{i+1,j}^{-1}\right]$；假设仿射变换 $\tilde{E}_{i+1}$ 的常数项是 $g_{i+1}$，$x$ 划分为 4 部分：$x=(x_0,x_1,x_2,x_3)$，则

$$\begin{aligned}y=\tilde{E}_{i+1}(x)&=E_{i+1}^{-1}\left(x\oplus A_{i+4}^{-1}\cdot d_{i+4}''\right)=l[E_{i+1}^{-1}](x)\oplus g_{i+1}\\&=diag\left(l\left[E_{i+1,0}^{-1}\right](x_0)\oplus g_{i+1,0},l\left[E_{i+1,1}^{-1}\right](x_1)\oplus g_{i+1,1},l\left[E_{i+1,2}^{-1}\right](x_2)\oplus g_{i+1,2},l\left[E_{i+1,3}^{-1}\right](x_3)\oplus g_{i+1,3}\right)\end{aligned}\tag{3-13}$$

线性变换 $L$ 可以表示成 32×32 的矩阵，可将其分成 16 个 8×8 的子块，

$$L=\begin{bmatrix}L_{00}&L_{01}&L_{02}&L_{03}\\L_{10}&L_{11}&L_{12}&L_{13}\\L_{20}&L_{21}&L_{22}&L_{23}\\L_{30}&L_{31}&L_{32}&L_{33}\end{bmatrix}$$

将图 3-11 的合成变换的输出记为 $y=(y_0,y_1,y_2,y_3)$，则每个 $y_j$ 均可以看成是输入 $(x_{i0},x_{i1},x_{i2},x_{i3})$ 的函数，

$$y_j(x_{i0},x_{i1},x_{i2},x_{i3})=l\left[E_{i+1,j}^{-1}\right]\left(\left\{\bigoplus_{t=0}^{3}L_{jt}\cdot S_{it}\cdot E_{it}(x_{it})\right\}\right)\oplus g_{i+1,j}\tag{3-14}$$

有了上述代数关系式，要恢复出 $E_{i+1}^{-1}$ 的线性部分还需要一个重要的命题来说明任意的两个 $(y_j,y_r)$ 之间存在仿射关系，该命题如下：

**命题** 1. 对于上述任意的 $(y_j,y_r)\left(j,r\in\{0,1,2,3\}\right)$，存在唯一的线性映射 $A_{jr}$ 和常数 $con_{jr}$，使得：$\forall x_{i0}\in GF(2^8)$，有

$$y_j(x_{i0},0,0,0)=A_{jr}\left(y_r(x_{i0},0,0,0)\right)\oplus con_{jr}\tag{3-15}$$

该命题的证明参见文献[15]。唯一的 $A_{jr}$ 和 $con_{jr}$ 的表达式分别为：

$$A_{jr}=l\left[E_{i+1,j}^{-1}\right]\cdot L_{j0}\cdot(L_{r0})^{-1}\cdot\left(l\left[E_{i+1,r}^{-1}\right]\right)^{-1}\tag{3-16}$$

$$con_{jr}=g_{i+1,j}\oplus l\left[E_{i+1,j}^{-1}\right](\beta_j)\oplus l\left[E_{i+1,j}^{-1}\right]\cdot L_{j0}\cdot L_{r0}^{-1}(\beta_r)\oplus l\left[E_{i+1,j}^{-1}\right]\cdot L_{j0}\cdot L_{r0}^{-1}\left(l\left[E_{i+1,\mathrm{r}}^{-1}\right]\right)^{-1}(g_{i+1,r})\tag{3-17}$$

其中，$\beta_j=\bigoplus_{t=1}^{3}L_{jt}\cdot S_{it}\cdot E_{it}(0)$，$\beta_r=\bigoplus_{t=1}^{3}L_{rt}\cdot S_{it}\cdot E_{it}(0)$。

线性映射 $A_{jr}$ 可以看成是含有 64 个未知项的 8×8 的矩阵，常数 $con_{jr}$ 可以看成是含 8 个未知项的列向量；对于一对 $(y_j,y_r)$，将等式 $y_j(x_{i0},0,0,0)=A_{jr}\left(y_r(x_{i0},0,0,0)\right)\oplus con_{jr}$ 展开可获得 8 个线性方程，当 $x_{i0}$ 取遍所有可能的值时，共有 $2^8$ 对 $(y_j,y_r)$，可获得 $2^8\times 8$ 个方程。所以，$A_{jr}$ 和 $con_{jr}$ 中共 72 个未知数，可以从中选择 8 个含 9 个未知数的方程组，每个方程组有 9 个方程，解方程组即可确定该唯一的 $A_{jr}$ 和 $con_{jr}$ 的值。

再次应用命题 1，可得

$$y'_r\left(0,x_{i1},0,0\right)=A'_{rj}\left(y'_j\left(0,x_{i1},0,0\right)\right)\oplus con'_{rj} \tag{3-18}$$

其中，$A'_{rj}=l\left[E_{i+1,r}^{-1}\right]\cdot L_{r1}\cdot\left(L_{j1}\right)^{-1}\cdot\left(l\left[E_{i+1,j}^{-1}\right]\right)^{-1}$，那么

$$\begin{aligned}A_{jr}\cdot A'_{rj}&=l\left[E_{i+1,j}^{-1}\right]\cdot L_{j0}\cdot\left(L_{r0}\right)^{-1}\cdot\left(l\left[E_{i+1,r}^{-1}\right]\right)^{-1}\cdot l\left[E_{i+1,r}^{-1}\right]\cdot L_{r1}\cdot\left(L_{j1}\right)^{-1}\cdot\left(l\left[E_{i+1,\mathrm{j}}^{-1}\right]\right)^{-1}\\&=l\left[E_{i+1,j}^{-1}\right]\cdot L_{j0}\cdot\left(L_{r0}\right)^{-1}\cdot L_{r1}\cdot\left(L_{j1}\right)^{-1}\cdot\left(l\left[E_{i+1,\mathrm{j}}^{-1}\right]\right)^{-1}\end{aligned} \tag{3-19}$$

$A_{jr}$ 和 $A'_{rj}$ 可以求出，各分块矩阵 $L_{ij}$ 是已知的，因此上式只有 $l\left[E_{i+1,j}^{-1}\right]$ 及其逆是未知的。$l\left[E_{i+1,j}^{-1}\right]$ 为 $8\times8$ 的矩阵，含有 64 个未知数，上式展开可以获得 64 个含有 64 个未知数的方程，通过解方程组可以求出 $l\left[E_{i+1,j}^{-1}\right]$。

经过四次相同的计算 $(j=0,1,2,3)$ 即可以求出 $E_{i+1}^{-1}$ 的线性部分。

**第二步：获取 $Q_i$ 的常数项 $r_i$**

按照上一步骤所述的方法，我们可以分别求出 $l\left[E_i^{-1}\right]$ 和 $l\left[E_{i+1}^{-1}\right]$。由于 $l[E_i]=\left(l\left[E_i^{-1}\right]\right)^{-1}$，因此 $l[E_{i0}]$，$l[E_{i1}]$，$l[E_{i2}]$ 和 $l[E_{i3}]$ 均可求出。又仿射变换 $M_{i+4}^{i+1}$ 和 $C_i$ 已知，因为 $M_{i+4}^{i+1}=E_{i+1}^{-1}\circ P_{i+1}^{-1}$，因此 $l\left[P_{i+4}^{-1}\right]=\left(l\left[E_{i+4}^{-1}\right]\right)^{-1}\cdot l\left[M_{i+4}^{i+1}\right]$ 可以求出，$l\left[P_{i+4}\right]=\left(l\left[P_{i+4}^{-1}\right]\right)^{-1}$ 也可以求出来，由于 $P''_{i+4}(x)=P_{i+4}(x)\oplus a''_i$，$l\left[P''_{i+4}\right]=l\left[P_{i+4}\right]$ 也可以求出；又因为 $P''_{i+4}\circ Q_i^{-1}=C_i$，从而 $l\left[Q_i^{-1}\right]=\left(l\left[P''_{i+4}\right]\right)^{-1}\cdot l[C_i]$，所以 $l[Q_i]=\left(l\left[Q_i^{-1}\right]\right)^{-1}$ 也可以求出。

将肖-白盒 SMS4 的 Part 2 部分变换为如下图所示。

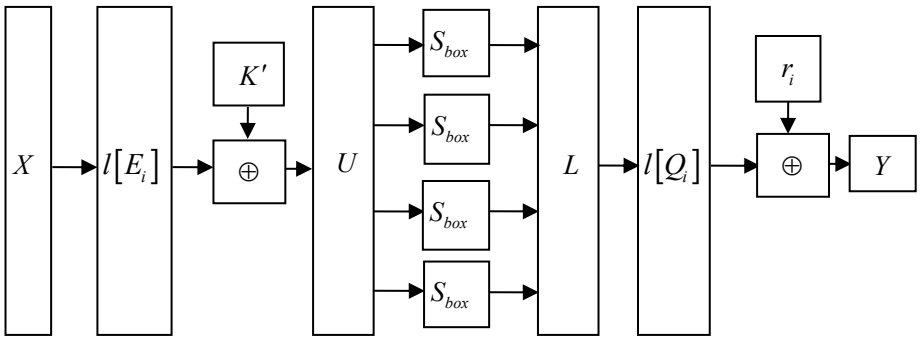


图 3-12 肖-白盒 SMS4 Part 2 的变换形式

图 3-12 中，$K'=rk_i\oplus c\left[E_i\right]$，$U$ 为 S 盒的输入。

给定 $X_1$ 和 $X_2$，则经过查找查找表及异或操作可得 $Y_1$ 和 $Y_2$。由于 $l\left[E_i\right]$ 已知，则可以计算出 S 盒的输入差分 $\Delta U=U_1\oplus U_2=l\left[E_i\right](X_1)\oplus l\left[E_i\right](X_2)$，整个变换的输

出差分为 $\Delta Y = Y_1 \oplus Y_2$。

利用差分分析法，可以获得 $Q_i$ 的常数项 $r_i$，步骤如下：

第 1 步：在查找表中随机的选出两组数 $(v_{i0}, v_{i1}, v_{i2}, v_{i3})$ 和 $(v_{i0}^*, v_{i1}^*, v_{i2}^*, v_{i3}^*)$，则 $Y = v_{i0} \oplus v_{i1} \oplus v_{i2} \oplus v_{i3}$，$Y^* = v_{i0}^* \oplus v_{i1}^* \oplus v_{i2}^* \oplus v_{i3}^*$。对 $(v_{i0}, v_{i1}, v_{i2}, v_{i3})$ 和 $(v_{i0}^*, v_{i1}^*, v_{i2}^*, v_{i3}^*)$ 反向查找查找表可以得到对应的明文 $(x_{i0}, x_{i1}, x_{i2}, x_{i3})$ 和 $(x_{i0}^*, x_{i1}^*, x_{i2}^*, x_{i3}^*)$，则

$$\begin{aligned}&\Delta U = U \oplus U^* \\ &= l[E_i](X) \oplus l[E_i](X^*) \\ &= \left(l[E_{i0}](x_{i0}) \| l[E_{i1}](x_{i1}) \| l[E_{i2}](x_{i2}) \| l[E_{i3}](x_{i3})\right) \oplus \left(l[E_{i0}](x_{i0}^*) \| l[E_{i1}](x_{i1}^*) \| l[E_{i2}](x_{i2}^*) \| l[E_{i3}](x_{i3}^*)\right)\end{aligned} \tag{3-20}$$

其中，符号“$\|$”表示联接。

第 2 步：对 $2^{32}$ 个可能的值 $r_i$ 设置 $2^{32}$ 个计数器 $\Lambda_j$（$1 \le j \le 2^{32}$），用每个 $r_i$ 解密 $Y$ 和 $Y^*$，并计算：

$$\begin{aligned}U' = &S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_0 \| S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_1 \\ &\| S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_2 \| S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_3\end{aligned} \tag{3-21}$$

$$\begin{aligned}U'^* = &S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_0 \| S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_1 \\ &\| S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_2 \| S_{box}^{-1}\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_3\end{aligned} \tag{3-22}$$

其中，$\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_0$ 代表 $\left\{L^{-1} \cdot \left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}$ 的第 1 个字节，依次类推；$S_{box}^{-1}$ 代表 S 盒的逆向查找。

验证 $\Delta U \overset{?}{=} U' \oplus U'^*$，如果相等，则对应的计算器 $\Lambda_j$ 加 1。

第 3 步：重复执行前两个步骤，直到某个计数器的值明显高于其他计数器的值，则这个计算器所对应的 $r_i$ 就是所求。

**第三步：恢复 $\tilde{E}_{i+1}$ 的常数项 $g_{i+1}$**

在第一步中，我们通过合并 Part 2，Part 3 的一部分以及下一轮的 Part 1 的一部分，得到了一个新的仿射变换 $\tilde{E}_{i+1}$，并记它的常数项为 $g_{i+1}$。

由于 $S_{ij}(*) = S_{box}(* \oplus rk_{ij})$，所以构造映射 $x \to S_{box}^{-1} \circ S_{ij} \circ E_{ij}(x) = E_{ij}(x) \oplus rk_{ij}$，其中 $rk_{ij}$ 是第 $i$ 轮密钥的第 $j$ 个字节。可知它也是一个仿射变换。由此我们可以得出下面这个命题：

**命题 2.** 存在唯一的 $GF(2^8)$ 中的元素对 $(\delta_i, \gamma_i)$（$i = 0,1,2,3$，$\delta_i$ 不为 0）（$\Lambda_\delta$ 代表乘以 $\delta$），使得

$$\tilde{P}_0 : x \to \left(S^{-1} \circ \Lambda_{\delta_0} \circ l\left[E_{i+1,0}\right]\right)\left(y_0\left(x,'00','00','00'\right) \oplus \gamma_0\right)$$
$$\tilde{P}_1 : x \to \left(S^{-1} \circ \Lambda_{\delta_1} \circ l\left[E_{i+1,0}\right]\right)\left(y_0\left('00',x,'00','00'\right) \oplus \gamma_1\right)$$
$$\tilde{P}_2 : x \to \left(S^{-1} \circ \Lambda_{\delta_2} \circ l\left[E_{i+1,0}\right]\right)\left(y_0\left('00','00',x,'00'\right) \oplus \gamma_2\right)$$
$$\tilde{P}_3 : x \to \left(S^{-1} \circ \Lambda_{\delta_3} \circ l\left[E_{i+1,0}\right]\right)\left(y_0\left('00','00','00',x\right) \oplus \gamma_3\right)$$

均为仿射变换。更进一步，这些仿射变换实际上就是 $\tilde{P}_j\left(x\right) = E_{ij}\left(x\right) \oplus rk_{ij}$ 。

命题 2 的证明参见文献[9]，且命题 2 中唯一的 $\left(\delta,\gamma\right) = \left('01','00'\right)$，其中，$\delta = \delta_0 \cdot L_{00}$，$\gamma = \delta_0 \cdot \left\{\beta_0 \oplus l\left[E_{i+1,0}\right]\left(g_{i+1,0}\right) \oplus l\left[E_{i+1,0}\right]\left(\gamma_0\right)\right\}$，$\beta_0 = L_{01} \cdot S_{i1} \cdot E_{i1}\left('00'\right) \oplus L_{02} \cdot S_{i2} \cdot E_{i2}\left('00'\right) \oplus L_{03} \cdot S_{i3} \cdot E_{i3}\left('00'\right)$。

由于 $\delta = \delta_0 \cdot L_{00}$ = ‘01’，而 $L_{00}$ 已知，所以可以求出 $\delta_0$。同理可求出 $\delta_1$， $\delta_2$， $\delta_3$。

对于所有可能的 $2^8$ 个 $\gamma_0$，对所有可能的 $x$ 依次查找查找表，在取定一个 $\gamma_0$ 的情况下验证 $\left(x,\tilde{P}_0\left(x\right)\right)$ 是否为仿射变换，即可确定该唯一的 $\gamma_0$ 的值。同理可求出 $\gamma_1$，$\gamma_2$， $\gamma_3$。

由于 $\gamma$ = ‘00’， $\gamma = \delta_0 \cdot \left\{\beta_0 \oplus l\left[E_{i+1,0}\right]\left(g_{i+1,0}\right) \oplus l\left[E_{i+1,0}\right]\left(\gamma_0\right)\right\}$，所以

$$\beta_0 \oplus l\left[E_{i+1,0}\right]\left(g_{i+1,0}\right) \oplus l\left[E_{i+1,0}\right]\left(\gamma_0\right) = \text{‘00’} \tag{3-23}$$

可以推出：

$$\begin{aligned}\gamma_0 &= l\left[E_{i+1,0}^{-1}\right]\left\{L_{00} \cdot S_{i0} \cdot E_{i0}\left(x\right) \oplus \beta_0\right\} \oplus l\left[E_{i+1,0}^{-1}\right]\left(L_{00} \cdot S_{i0} \cdot E_{i0}\left(x\right)\right) \oplus g_{i+1,0} \\ &= \left(l\left[E_{i+1,0}^{-1}\right]\left\{L_{00} \cdot S_{i0} \cdot E_{i0}\left(x\right) \oplus \beta_0\right\} \oplus g_{i+1,0}\right) \oplus l\left[E_{i+1,0}^{-1}\right]\left(L_{00} \cdot S_{i0} \cdot E_{i0}\left(x\right)\right) \\ &= y_0\left(x,'00','00','00'\right) \oplus l\left[E_{i+1,0}^{-1}\right]\left(L_{00} \cdot S_{i0} \cdot E_{i0}\left(x\right)\right)\end{aligned} \tag{3-24}$$

同理可得：

$$\gamma_1 = y_0\left('00',x,'00','00'\right) \oplus l\left[E_{i+1,0}^{-1}\right]\left(L_{01} \cdot S_{i1} \cdot E_{i1}\left(x\right)\right) \tag{3-25}$$

$$\gamma_2 = y_0\left('00','00',x,'00'\right) \oplus l\left[E_{i+1,0}^{-1}\right]\left(L_{02} \cdot S_{i2} \cdot E_{i2}\left(x\right)\right) \tag{3-26}$$

$$\gamma_3 = y_0\left('00','00','00',x\right) \oplus l\left[E_{i+1,0}^{-1}\right]\left(L_{03} \cdot S_{i3} \cdot E_{i3}\left(x\right)\right) \tag{3-27}$$

令 $\gamma_4 = y_0\left('00','00','00','00'\right)$，它也可表示为：

$$\gamma_4 = l\left[E_{i+1,0}^{-1}\right]\left(\bigoplus_{t=0}^{3} L_{0t} \cdot S_{it} \cdot E_{it}\left('00'\right)\right) \oplus g_{i+1,0} \tag{3-28}$$

则 $g_{i+1,0} = \gamma_0 \oplus \gamma_1 \oplus \gamma_2 \oplus \gamma_3 \oplus \gamma_4$，从而可以恢复出 $g_{i+1,0}$。同理可得 $g_{i+1,1}$，$g_{i+1,2}$，$g_{i+1,3}$，从而全部恢复出 $\tilde{E}_{i+1}$ 的常数项 $g_{i+1}$。

**第四步：确定子密钥 $rk_i$**

前面我们已经求出了 $E_i$ 的线性部分 $l[E_{i0}],l[E_{i1}],l[E_{i2}],l[E_{i3}]$，$Q_i$ 的线性部分 $l[Q_i]$ 以及 $Q_i$ 的常数项 $r_i$，现在只需求出 $E_i$ 的常数项，即可找到该轮所对应的轮密钥 $rk_i$。

首先，由于 $P_{i+4}(x)=A_{i+4}(x)\oplus a_{i+4}$，所以 $P_{i+4}^{-1}(x)=A_{i+4}^{-1}(x)\oplus A_{i+4}^{-1}(x)\cdot a_{i+4}$。在上面的小节中 $l\left[P_{i+4}^{-1}\right]$ 已求出，即 $A_{i+4}^{-1}$ 已知，所以 $A_{i+4}$ 也知。由于 $Q_i(x)=l[Q_i](x)\oplus r_i$，所以 $Q_i^{-1}(x)=l\left[Q_i^{-1}\right](x)\oplus l\left[Q_i^{-1}\right]\cdot r_i$。假设 $E_{i+1}^{-1}(x)=l\left[E_{i+1}^{-1}\right](x)\oplus e_{i+1}$：

（1）由于仿射变换 $M_{i+4}^{i+1}=E_{i+1}^{-1}\circ P_{i+4}^{-1}$，$M_{i+3}^{i+1}=E_{i+1}^{-1}\circ P_{i+3}^{-1}$ 和 $M_{i+3}^{i}=E_{i}^{-1}\circ P_{i+3}^{-1}$ 已知，因此 $M_{i+4}^{i+1}$、$M_{i+3}^{i+1}$ 和 $M_{i+3}^{i}$ 的常数项已知，故有：

$$c\left[M_{i+4}^{i+1}\right]=l\left[E_{i+1}^{-1}\right]\cdot A_{i+4}^{-1}\cdot a_{i+4}\oplus e_{i+1} \tag{3-29}$$

$$c\left[M_{i+3}^{i+1}\right]=l\left[E_{i+1}^{-1}\right]\cdot A_{i+3}^{-1}\cdot a_{i+3}\oplus e_{i+1} \tag{3-30}$$

$$c\left[M_{i+3}^{i}\right]=l\left[E_{i}^{-1}\right]\cdot A_{i+3}^{-1}\cdot a_{i+3}\oplus e_{i} \tag{3-31}$$

（2）同时，$\tilde{E}_{i+1}(x)=E_{i+1}^{-1}\left(x\oplus a''_{i+4}\right)$ 的常数项 $g_{i+1}$ 已知，因此有

$$g_{i+1}=l\left[E_{i+1}^{-1}\right]\cdot A_{i+4}^{-1}\cdot a''_{i+4}\oplus e_{i+1} \tag{3-32}$$

（3）此外，在白盒 SMS4 的 Part 2 部分，我们选定一个 32 比特的 $X_0$，查找查找表得到对应的 $Y_0$，然后通过正向计算与反向查找 S 盒，可以得到下面的等式：

$$\tau^{-1}\left(L^{-1}\left(Q_i^{-1}\right)(Y_0)\right)=l[E_i](X_0)\oplus e_i\oplus rk_i \tag{3-33}$$

其中，$\tau^{-1}$ 表示反向查找 SMS4 的 S 盒。

（4）最后，取 Part 1 中的一个分支与 Part 2 合并，如图 3-13 所示，

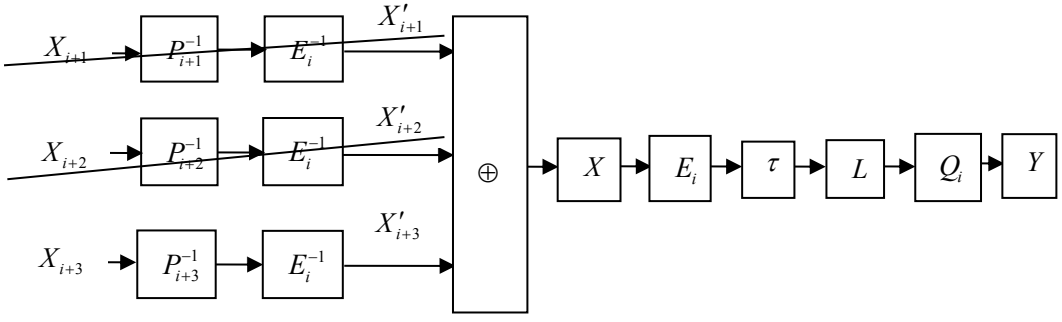


图 3-13 肖-白盒 SMS4 算法的 Part 1 与 Part 2 合并图

选定一个 32 比特的 $\tilde{X}_{i+3}$，通过上图的合并计算可得一个 32 比特的 $\tilde{Y}$，因此可得等

式：

$$A_{i+3}^{-1}\left(\tilde{X}_{i+3}\right)\oplus A_{i+3}^{-1}\cdot a_{i+3}\oplus rk_i=\tau^{-1}\left(L^{-1}\left(Q_i^{-1}\left(\tilde{Y}\right)\right)\right) \tag{3-34}$$

将上述的 6 个式子联立，有 6 个未知数 $a_{i+4}$， $a''_{i+4}$， $e_{i+1}$， $a_{i+3}$， $e_i$ 和 $rk_i$，其系数矩阵是非奇异的，因此可以通过求解 32 阶的矩阵方程组求出轮密钥 $rk_i$。

## 3.3 本章小结

本章介绍了 SMS4 算法，肖-白盒 SMS4 算法的设计方法以及其执行效率和安全性分析。

肖-白盒 SMS4 算法采用查找表与仿射变换相结合的方式，整个算法共用 148.625KB 的空间，共需执行 128 次表格查找和 160 次仿射变换。白盒 SMS4 算法是在原本 SMS4 算法的基础上做的白盒设计，需要占用额外的空间，且执行效率比原始 SMS4 算法的慢，因此，在实际应用时需要对系统资源及运行环境等因素做出综合分析。

从白盒多样性和白盒含混度的角度来看，肖-白盒 SMS4 算法能够抵抗穷举攻击；同时，肖-白盒 SMS4 算法也能有效的抵抗 BGE 攻击。然而，林婷婷等人利用将 BGE 攻击、差分分析法及求解方程组等方法相结合的方法，成功的恢复出了肖-白盒 SMS4 算法的轮密钥。该攻击方法能够有效的攻击成功的原因主要有两个，首先，它的每一轮的输出置乱编码和下一轮的输入置乱编码是互逆的，在合并相邻 Part 的时候会消去部分仿射变换对原 SMS4 算法的混淆作用；其次，由于仿射变换可以用矩阵和向量表示，这为建立输入输出线性方程组提供了条件。尽管肖-白盒 SMS4 算法被破译了，但是它仍具有比原 SMS4 算法更高的安全性，其破译方法也为我们研究具有更高安全性的白盒 SMS4 算法提供了思路。

# 第四章 白盒 SMS4 算法设计

本章提出对肖-白盒 SMS4 算法的两种改进算法。详细介绍这两种改进算法的算法描述、实现复杂度分析以及安全性分析，其中实现复杂度分析主要指算法占用的额外空间以及查表和仿射变换的次数，安全性分析包括白盒多样性及白盒含混度的分析以及抵抗现有攻击的能力。

## 4.1 白盒 SMS4 改进算法一

### 4.1.1 改进算法一算法描述

肖-白盒 SMS4 算法采用查找表与仿射变换相结合的方式，密钥信息隐藏在查找表中，能够抵抗穷举攻击和 BGE 攻击。然而在文献[9]中，林婷婷等人利用将 BGE 攻击、差分分析法及求解方程组等方法相结合的方法，成功的恢复出了肖-白盒 SMS4 算法的轮密钥，所用总时间复杂度为 $<2^{47}$，其中在获取 $Q_i$ 的常数项 $r_i$ 时所用时间复杂度最多，为 $<2^{46}$，其他攻击过程所用的时间复杂度均为 $<2^{25}$。改进算法一即从提高攻击时间复杂度的角度进行改进。

肖-白盒 SMS4 算法的每一轮的第一部分使用 3 个仿射变换 $M_{i+j}^{\ i}$ 和两个异或计算，而 $M_{i+j}^{\ i}$ 是由两个可逆仿射变换 $P_{i+j}^{-1}$（$j=1,2,3$）和 $E_i^{-1}$ 结合而成，其中 $P_{i+j}^{-1}$ 是 $GF(2)$ 上 32 比特到 32 比特的可逆仿射变换，$E_i^{-1}=diag\left(E_{i0}^{-1},E_{i1}^{-1},E_{i2}^{-1},E_{i3}^{-1}\right)$，$E_{ij}^{-1}$ 是 $GF(2)$ 上 8 比特到 8 比特的可逆仿射变换。$E_i^{-1}$ 之所以这样设定，是为了在 Part 2 中（参见图 3-7）可以将一个 32 比特到 32 比特的查找表分割为 4 个 8 比特到 32 比特的查找表，节省存储空间。

将肖-白盒 SMS4 中的参数 $E_i=diag\left(E_{i0},E_{i1},E_{i2},E_{i3}\right)$ 改为 $E_i=diag\left(E'_{i0},E'_{i1}\right)$，$E'_{i0}$、$E'_{i1}$ 均为 $GF(2)$ 上的 16 比特到 16 比特的仿射变换，Part 1 与 Part 3 不变。改进算法 Part 2 部分如图 4-1 所示。

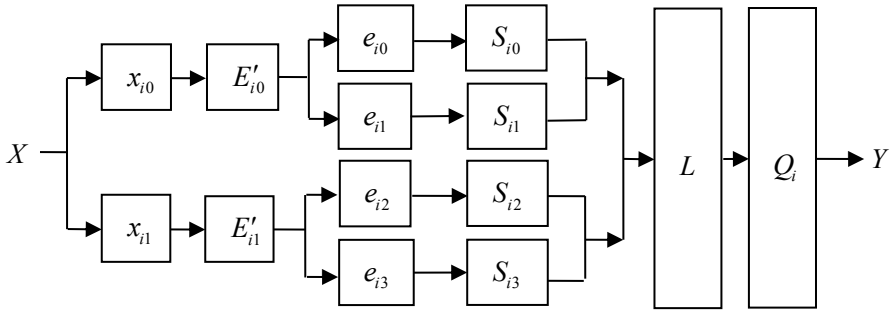


图 4-1 改进算法一的 Part 2 部分

将 $X=(x_{i0},x_{i1})$ 经 $E_{ij}$ 和 $S_{ij}$ 变换后的值记为 $(z_{i0},z_{i1})$，将仿射变换 $Q_i\cdot L$ 拆分如下：

$$(Q_i\cdot L)\begin{bmatrix}z_{i0}\\z_{i1}\end{bmatrix}=(R_{i0},R_{i1})\cdot\begin{bmatrix}z_{i0}\\z_{i1}\end{bmatrix}\oplus r_i=(R_{i0}\cdot z_{i0})\oplus(R_{i1}\cdot z_{i1}\oplus r_i)=v_{i0}\oplus v_{i1} \quad (4\text{-}1)$$

其中，$R_i$ 为 $32\times16$ 的矩阵，$r_i$ 为 32 比特的常数列向量。这样，$X\to Y$ 的过程就可以转化为两个 16 比特到 32 比特的查找表：$x_{ij}(\to z_{ij})\to v_{ij}$，$j=0,1$。Part 2 的过程即为查找两个查找表，再计算这两个查找表结果的异或过程。

改进算法一每一轮分为三个部分：第一部分使用 3 个仿射变换 $M_{i+j}{}^{i}$ 和 2 个异或计算；第二部分查找 2 个 16 比特到 32 比特的查找表和 1 个异或计算；第三部分使用两个仿射变换 $D_i$ 与 $C_i$ 和一个异或计算。该算法的设计思路与肖-白盒 SMS4 算法相同，不同的是每一轮 $E_i$ 的取值以及查找表的大小与个数。

### 4.1.2 改进算法一实现复杂度分析

改进算法一的实现是基于查找表方式的，算法执行过程主要通过仿射变换和查找表来完成。算法共 32 轮迭代过程，每一轮划分为三个部分，整个改进算法一的执行过程共需要 64 次表格查找以及 160 次 32 比特到 32 比特的仿射变换。

每一轮所占用的空间为：

Part 1 包括 3 个 32 比特到 32 比特的仿射变换：$3\times(32\times32+32)$=3168（比特）；

Part 2 包括 2 个 16 比特到 32 比特的查找表：$2\times(2^{16}\times32)$=4194304（比特）；

Part 3 包括 2 个 32 比特到 32比特的仿射变换：$2\times(32\times32+32)$=2112（比特）；

所以，整个白盒SMS4所占用的空间为：

$(3168+4194304+2112)\times32$ =134386688 比特=16798336B=16.012014MB。

### 4.1.3 改进算法一安全性分析

#### 4.1.3.1 白盒多样性及白盒含混度

由本文第三章的定理 3-1 的计算方式可知，$GF(2)$ 上 16 阶可逆矩阵的个数约为 $2^{254}$。

改进算法一中，每一轮的白盒多样性的值如下：

Part 1：$\left(2^{922}\times2^{32}\right)^3\times\left(2^{254}\times2^{16}\right)^2=2^{3402}$

Part 2：$\left(2^{254}\times2^{16}\right)^2\times2^{32}\times\left(2^{922}\times2^{32}\right)=2^{1526}$

Part 3：$\left(2^{922}\times2^{32}\right)^3\times2^{32}=2^{2894}$

每一轮的白盒含混度的值如下（大致的估计）：

Part 1：$\left(2^{254}\times 2^{16}\right)^2 = 2^{540}$

Part 2：$\left(2^{254}\times 2^{16}\right)^2 \times 2^{32} = 2^{572}$

Part 3：$2^{922}\times 2^{32}\times 2^{32} = 2^{986}$

从白盒多样性和白盒含混度来看，白盒 SMS4 改进算法一是安全的，攻击者很难从仿射变换或者查找表中推断出输入输出置乱编码及密钥信息。

#### 4.1.3.2 抵抗现有攻击的能力

对于基于 Chow 等人的查找表方式设计的白盒算法而言，常用的攻击方式是 BGE 攻击。肖-白盒 SMS4 算法是能够抵抗 BGE 攻击的，而本文的改进算法一是在肖-白盒 SMS4 的基础上作出的改进，其采用的主要方法也是仿射变换与查找表相结合的方式，因此，改进算法一能够抵抗 BGE 攻击。然而，肖-白盒 SMS4 算法不能抵抗林婷婷等人的一种攻击方法，该方法可概括为 BGE 攻击方法与差分密码分析法，以及求解方程组等相结合的方法。针对改进算法一，我们分析其能否抵抗这种攻击。

改进算法一的设计方法与肖-白盒 SMS4 类似，故可按照类似林等人的攻击方法进行分析。合并 Part 2，Part3 的一部分以及下一轮的 Part 1 的一部分，如图 4-2 所示。

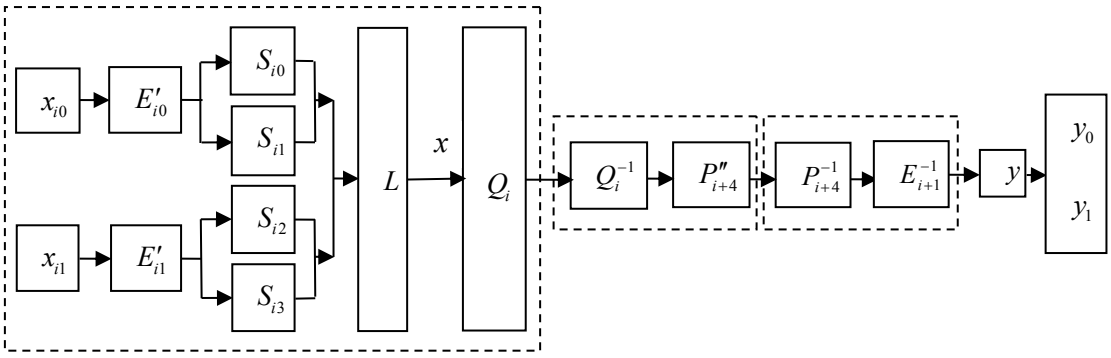


图 4-2 合并变换

由图 4-2 可知，$Q_i$ 与 $Q_i^{-1}$ 完全抵消，而 $P''_{i+4}$ 与 $P_{i+4}^{-1}$ 相抵消剩下常数 $A_{i+4}^{-1}\cdot a''_{i+4}$。因此，经 $L$ 输出的数据 $x$ 只需要与常数 $A_{i+4}^{-1}\cdot a''_{i+4}$ 异或，再经过 $E_{i+1}^{-1}$ 变换即可得到 $y$，整个过程可以表示为：$y=\tilde{E}_{i+1}(x)=E_{i+1}^{-1}\left(x\oplus A_{i+4}^{-1}\cdot a''_{i+4}\right)$。又 $E_{i+1}^{-1}=diag\left(E_{i+1,0}^{-1},E_{i+1,1}^{-1}\right)$，$E_{i+1,j}^{-1}$ 是 16 比特到 16 比特的仿射变换，可表示为 $E_{i+1,j}^{-1}(x)=l\left[E_{i+1,j}^{-1}\right](x)\oplus c\left[E_{i+1,j}^{-1}\right]$；假设仿射变换 $\tilde{E}_{i+1}$ 的常数项是 $g_{i+1}$，$x=(x_0,x_1)$，则

$$\begin{aligned} y = \tilde{E}_{i+1}(x) &= E_{i+1}^{-1}\left(x \oplus A_{i+4}^{-1} \cdot a''_{i+4}\right) = l[E_{i+1}^{-1}](x) \oplus g_{i+1} \\ &= diag\left(l\left[E_{i+1,0}^{-1}\right](x_0) \oplus g_{i+1,0}, l\left[E_{i+1,1}^{-1}\right](x_1) \oplus g_{i+1,1}\right) \end{aligned} \tag{4-2}$$

线性变换 $L$ 可以表示成32×32的矩阵，可将其分成 4 个16×16的子块，

$$L = \begin{bmatrix} L_{00} & L_{01} \\ L_{10} & L_{11} \end{bmatrix}$$

将图 4-2 的合成变换的输出记为 $y=(y_0,y_1)$，则每个 $y_j$ 均可以看成是输入 $(x_{i0}x_{i1})$的函数，

$$y_j(x_{i0},x_{i1}) = l\left[E_{i+1,j}^{-1}\right]\left(\begin{Bmatrix} L_{j0}\cdot\left(S_{i0}\left(h\left(E_{i0}(x_{i0})\right)\right) \| S_{i1}\left(l\left(E_{i0}(x_{i0})\right)\right)\right) \\ \oplus L_{j1}\cdot\left(S_{i2}\left(h\left(E_{i1}(x_{i1})\right)\right) \| S_{i3}\left(l\left(E_{i1}(x_{i1})\right)\right)\right) \end{Bmatrix}\right) \oplus g_{i+1,j} \tag{4-3}$$

其中，$h(x)$表示 $x$ 的前 8 比特，$l(x)$表示 $x$ 的后 8 比特，“$\|$”表示联接。

由于在林等人的攻击方法中，总时间复杂度为$<2^{47}$，其中在获取 $Q_i$ 的常数项 $r_i$ 时所用时间复杂度最多，为$<2^{46}$，因此，我们按照类似的方法分析获取 $Q_i$ 的常数项 $r_i$的过程。如图 3-12 所示的 Part 2 的变形，按照差分分析法，可以得到 S 盒的输入差分 $\Delta U = U_1 \oplus U_2 = l[E_i](X_1) \oplus l[E_i](X_2)$，整个变换的输出差分 $\Delta Y = Y_1 \oplus Y_2$。计算常数项 $r_i$的步骤如下：

（1）在查找表中随机的选出两组数 $(v_{i0},v_{i1})$ 和 $(v_{i0}^*,v_{i1}^*)$，则 $Y = v_{i0} \oplus v_{i1}$，$Y^* = v_{i0}^* \oplus v_{i1}^*$。对 $(v_{i0},v_{i1})$ 和 $(v_{i0}^*,v_{i1}^*)$ 反向查找查找表可以得到对应的明文 $(x_{i0},x_{i1})$ 和 $(x_{i0}^*,x_{i1}^*)$，则

$$\begin{aligned} \Delta U &= U \oplus U^* \\ &= l[E_i](X) \oplus l[E_i](X^*) \\ &= \left(l[E_{i0}](x_{i0}) \| l[E_{i1}](x_{i1})\right) \oplus \left(l[E_{i0}](x_{i0}^*) \| l[E_{i1}](x_{i1}^*)\right) \end{aligned} \tag{4-4}$$

其中，符号“$\|$”表示联接。

（2）对 $2^{32}$ 个可能的值 $r_i$ 设置 $2^{32}$ 个计数器 $\Lambda_j$（$1 \le j \le 2^{32}$），用每个 $r_i$ 解密 $Y$ 和 $Y^*$，并计算：

$$\begin{aligned} U' = S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_0 \| S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_1 \\ \| S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_2 \| S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y \oplus r_i)\right\}_3 \end{aligned} \tag{4-5}$$

$$\begin{aligned} U'^* = S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_0 \| S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_1 \\ \| S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_2 \| S_{box}^{-1}\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}(Y^* \oplus r_i)\right\}_3 \end{aligned} \tag{4-6}$$

其中，$\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}\left(Y\oplus r_i\right)\right\}_0$ 代表 $\left\{L^{-1}\cdot\left(l[Q_i]\right)^{-1}\left(Y\oplus r_i\right)\right\}$ 的第 1 个字节，依次类推；$S_{box}^{-1}$ 代表 S 盒的逆向查找。

验证 $\Delta U\overset{?}{=}U'\oplus U'^{*}$，如果相等，则对应的计算器 $\Lambda_j$ 加 1。

（3）重复执行前两个步骤，直到某个计数器的值明显高于其他计数器的值，那么这个计算器所对应的 $r_i$ 即为所求。

我们记 $n$ 阶矩阵相乘的时间复杂度为 $n^{\omega}$，一般情况下 $\omega = 3$，有更好的算法可以使 $\omega = 2.376$，这里我们取 $\omega = 2.4$；$n$ 阶矩阵求逆的时间复杂度为 $n^3$。

因此，步骤（1）包含 4 个 $16\times16$ 矩阵与 $16\times1$ 矩阵的相乘，时间复杂度为 $4\times16^2=2^{10}$；步骤（2）忽略 S 盒查表与异或计算，计算矩阵乘法的复杂度为 $32^{\omega}+2\times32^2$；计算 $r_i$ 的总的时间复杂度为 $2^{32}\left(2^{10}+32^{\omega}+2\times32^2\right)+o\left(l[E_{ij}]\right)<2^{47}$，其中 $o\left(l[E_{ij}]\right)$ 为计算 $l[E_{ij}]$ 的时间复杂度。

由于在林等人的攻击方法中，在获取 $Q_i$ 的常数项 $r_i$ 时所用时间复杂度最多，且比较式（3-14）与式（4-3），可以发现式（3-14）较工整一些，考虑一种理想的情况，即按照类似林等人的攻击方法可以对改进算法一攻击成功，那么其攻击的时间复杂度将为 $<2^{48}$。可知，改进算法一可以将攻击时间复杂度至少提高 2 倍。

## 4.2 白盒 SMS4 改进算法二

### 4.2.1 改进算法二算法描述

肖-白盒 SMS4 算法的设计方法本质是 Chow 等人的查找表法，查找表方法的主要思想是采用混淆的方式，对现有的一个分组密码，选定一个密钥，然后对明文到密文的映射进行置乱编码，将密码算法转换为查找表，然后利用外部编码和内部编码对查找表进行编码，编码后的查找表内容会被隐藏，从而实现对密钥信息的隐藏、混乱和扩散，同时利用两个相邻查找表的输入置乱编码与输出置乱编码在两两级联以后会相互抵消的特性，来保证密码算法的完整性和可用性。

根据查找表法设计白盒密码的思想，我们分析肖-白盒 SMS4 中参数 $P'_{i+4}$、$P''_{i+4}$ 与 $P_{i+4}^{-1}$ 的关系。将白盒 SMS4 设计中的 Part 3 和接下来的关于输出 $X_{i+4}$ 的输入置乱部分结合起来，如图 4-3 所示。

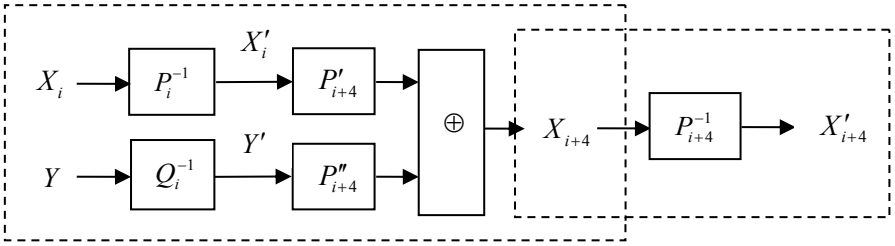


图 4-3 Part 3 与相关输入置乱相结合

由于每一轮的输入置乱是对上一轮的输出置乱的抵消，因此 $X'_{i+4}=X'_i\oplus Y'$。又

$$\begin{aligned}X'_{i+4}&=P_{i+4}^{-1}\left(X_{i+4}\right)=P_{i+4}^{-1}\left(P'_{i+4}\left(X'_i\right)\oplus P''_{i+4}\left(Y'\right)\right)\\&=P_{i+4}^{-1}\left(P_{i+4}\left(X'_i\right)\oplus a'_{i+4}\oplus P_{i+4}\left(Y'\right)\oplus a''_{i+4}\right)\\&=P_{i+4}^{-1}\left(l\left[P_{i+4}\right]\left(X'_i\right)\oplus c\left[P_{i+4}\right]\oplus a'_{i+4}\oplus l\left[P_{i+4}\right]\left(Y'\right)\oplus c\left[P_{i+4}\right]\oplus a''_{i+4}\right)\\&=P_{i+4}^{-1}\left(l\left[P_{i+4}\right]\left(X'_i\oplus Y'\right)\oplus a'_{i+4}\oplus a''_{i+4}\right)\\&=\left(l\left[P_{i+4}\right]\right)^{-1}\left(l\left[P_{i+4}\right]\left(X'_i\oplus Y'\right)\oplus a'_{i+4}\oplus a''_{i+4}\right)\oplus\left(l\left[P_{i+4}\right]\right)^{-1}\cdot c\left[P_{i+4}\right]\\&=X'_i\oplus Y'\oplus\left(l\left[P_{i+4}\right]\right)^{-1}\cdot\left(a'_{i+4}\oplus a''_{i+4}\oplus c\left[P_{i+4}\right]\right)\end{aligned}$$

因此，$\left(l\left[P_{i+4}\right]\right)^{-1}\cdot\left(a'_{i+4}\oplus a''_{i+4}\oplus c\left[P_{i+4}\right]\right)=0$。因为$\left(l\left[P_{i+4}\right]\right)^{-1}$可逆，所以可知

$$a'_{i+4}\oplus a''_{i+4}\oplus c\left[P_{i+4}\right]=0 \qquad (4\text{-}7)$$

因此，我们可以令$\begin{cases}c\left[P_{i+4}\right]=0\\a'_{i+4}=a''_{i+4}\end{cases}$，即 $P_{i+j}$ 是 $32\times32$ 的可逆矩阵，$P'_{i+4}=P''_{i+4}$ 均为 32 比特到 32 比特的可逆仿射变换，且 $P'_{i+4}\left(x\right)=P''_{i+4}\left(x\right)=P_{i+4}\cdot x\oplus a_{i+4}$。这样做的目的是减小一定的额外空间。

肖-白盒 SMS4 算法实质是在原黑盒 SMS4 算法上的白盒设计，为了保证白盒算法的完整性和可用性，要在进行内部编码之后进行外部编码。SMS4 算法是 32 轮的非线性迭代分组算法，因此，我们除了在做白盒设计时，不仅要对每一轮进行白盒化，而且在第一组输入之前和最后一组输出之后进行置乱编码，以保证算法的完整性。

整体的白盒 SMS4 算法流程如图 4-4 所示。

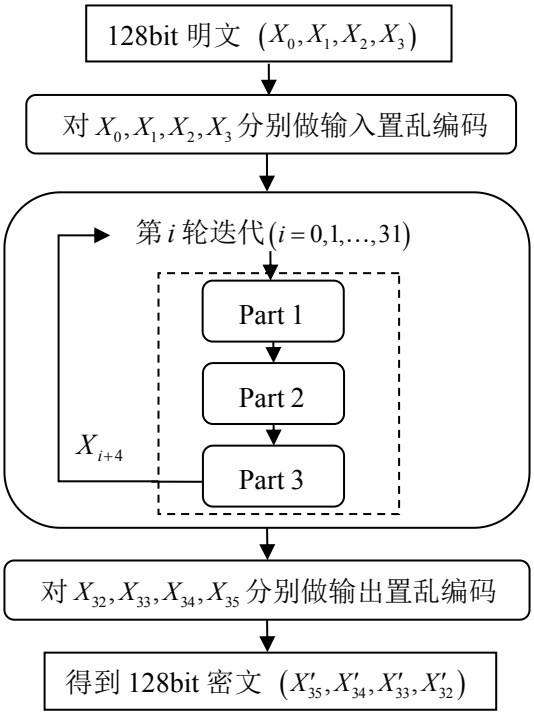


图 4-4 改进算法二整体算法流程

改进的白盒 SMS4 算法在第一轮之前和最后一轮之后分别做了输入、输出置乱编码。对明文分组 $(X_0, X_1, X_2, X_3)$ 分别用 $32\times32$ 的可逆矩阵 $P_i$ 进行混淆，得到的 $X'_i = P_i \cdot X_i \ (i = 0,1,2,3)$ 作为轮函数的输入，每一轮函数分割为 Part 1、Part 2 和 Part 3 三个部分，经过 32 轮变换，得到输出 $X_{32}, X_{33}, X_{34}, X_{35}$，对它们分别用 $32\times32$ 的可逆矩阵 $P_i$ 进行置乱编码，得到 $X'_i = P_i^{-1} \cdot X_i \ (i = 32,33,34,35)$，然后进行一次 $R$ 变换，得到的 $(X'_{35}, X'_{34}, X'_{33}, X'_{32})$ 即为密文分组。其中，$P_i \ (i = 0,1,2,3)$ 与前四轮中的 Part 1 部分对 $X_i \ (i = 0,1,2,3)$ 的输入置乱相抵消，$P_i^{-1} \ (i = 32,33,34,35)$ 与最后四轮中的 Part 3 部分对 $X_i \ (i = 32,33,34,35)$ 的输出置乱相抵消，以此确保算法的完整性。

### 4.2.2 改进算法二的实现复杂度分析

改进算法二的实现也是基于查找表方式的，算法执行过程可以通过仿射变换和查找表来完成。每一轮过程划分为三个部分，在第一轮之前和最后一轮之后执行外部编码，整个白盒 SMS4 改进算法二的执行需要 128 次表格查找、160 次 32 比特到 32 比特的仿射变换以及 8 次 32 比特到 32 比特的双射变换。

改进算法二的每一轮所占用的空间与肖-白盒 SMS4 算法的每一轮所占用的空间完全一样。

第一轮之前与最后一轮之后所占用空间均包括 4 个 32 比特到 32 比特的可逆双射，共占用空间为：$2\times4\times(32\times32)$=8192（比特）。

所以，整个白盒 SMS4 所占用的空间为：

1217536+8192=1225728 比特=153216B=149.625KB

下表是几种密码算法的执行效率比较。

表 4-1 几种密码算法执行效率的比较（二）

| 加密算法 | 占用空间 | 查表次数 | 仿射变换次数 |
|---|---|---|---|
| 肖-白盒 SMS4 | 148.625KB | 128 | 160 |
| 本文改进算法一 | 16.012MB | 64 | 160 |
| 本文改进算法二 | 149.625KB | 128 | 160 |

根据表 4-1 的数据可知，改进算法一、改进算法二与肖-白盒 SMS4 计算仿射变换的次数相同；改进算法一占用较多的额外空间，但是它节约了一半的查表次数；改进算法二的查表次数与肖-白盒 SMS4 相同，总体占用空间比肖-白盒 SMS4 高 1KB。

### 4.2.3 改进算法二的安全性分析

#### 4.2.3.1 白盒多样性及白盒含混度

在改进算法二中，每一轮的白盒多样性的值如下：

Part 1：$\left(2^{922}\times2^{32}\right)^{3}\times\left(2^{62}\times2^{8}\right)^{4}=2^{3142}$

Part 2：$\left(2^{62}\times2^{8}\right)^{4}\times2^{32}\times\left(2^{922}\times2^{32}\right)=2^{1266}$

Part 3：$2^{922}\times\left(2^{922}\times2^{32}\right)^{2}\times2^{32}=2^{2862}$

每一轮的白盒含混度的估计值如下：

Part 1：$\left(2^{62}\times2^{8}\right)^{4}=2^{280}$

Part 2：$\left(2^{62}\times2^{8}\right)^{4}\times2^{32}=2^{312}$

Part 3：$2^{922}\times2^{32}\times2^{32}=2^{986}$

从白盒多样性和白盒含混度来看，白盒 SMS4 改进算法二是安全的，攻击者很难从仿射变换或者查找表中推断出输入输出置乱编码及密钥信息。

表 4-2 是几种密码算法的白盒多样性及白盒含混度比较。

表 4-2 几种密码算法的每一轮各部分白盒多样性及白盒含混度比较

| 加密算法 | | 白盒多样性 | 白盒含混度 |
|---|---|---|---|
| Part 1 | 肖-白盒 SMS4 | $2^{3142}$ | $2^{280}$ |
| | 改进算法一 | $2^{3402}$ | $2^{540}$ |
| | 改进算法二 | $2^{3142}$ | $2^{280}$ |
| Part 2 | 肖-白盒 SMS4 | $2^{1266}$ | $2^{312}$ |
| | 改进算法一 | $2^{1256}$ | $2^{572}$ |
| | 改进算法二 | $2^{1266}$ | $2^{312}$ |
| Part 3 | 肖-白盒 SMS4 | $2^{2894}$ | $2^{986}$ |
| | 改进算法一 | $2^{2894}$ | $2^{986}$ |
| | 改进算法二 | $2^{2862}$ | $2^{986}$ |

由表 4-2 中的数据可知，改进算法一总体的白盒多样性比肖-白盒 SMS4 大约高 $2^{250}$，白盒含混度比肖-白盒 SMS4 大约高 $2^{520}$；改进算法二总体的白盒多样性比肖-白盒 SMS4 大约低 $2^{32}$，白盒含混度与肖-白盒 SMS4 相同。不过，从白盒多样性及白盒含混度的角度来讲，两种改进算法都均有效地抵抗穷举攻击。

#### 4.2.3.1 抵抗现有攻击的能力

对于基于 Chow 等人的查找表方式设计的白盒算法而言，常用的攻击方式是 BGE 攻击。肖-白盒 SMS4 算法是能够抵抗 BGE 攻击的，而本文的改进算法二是在肖-白盒 SMS4 的基础上作出的改进，其采用的主要方法也是仿射变换与查找表相结合的方式，因此，改进算法二能够抵抗 BGE 攻击。然而，肖-白盒 SMS4 算法不能抵抗林婷婷等人的一种攻击方法，该方法可概括为 BGE 攻击方法与差分密码分析法，以及求解方程组等相结合的方法。

改进算法二与肖-SMS4 白盒设计的区别是：简化了肖-白盒 SMS4 算法中的参数 $P_{i+j}$、$P'_{i+4}$ 和 $P''_{i+4}$， 并且增加了外部编码，使算法具有完整性。但是其主要设计思想与肖-SMS4 白盒算法相同，故可采用林等人的攻击方法对其实现有效攻击，恢复出轮密钥。

## 4.3 本章小结

本章提出了对肖-白盒 SMS4 设计的两种改进算法，并分别对其执行效率、安全性进行了分析。

改进算法一将肖-白盒 SMS4 算法中的四个查找表改为两个查找表，节省了查

表次数，但是它增大了额外的空间，不过，从安全性方面来讲，改进算法一可以将攻击时间复杂度至少提高 2 倍。

改进算法二一方面简化了肖-白盒 SMS4 算法中的参数 $P_{i+j}$ 、 $P'_{i+4}$ 和 $P''_{i+4}$ ，其好处是在计算保存参数的过程中会减小计算的复杂度，而简化参数后算法的白盒多样性和白盒含混度仍能满足一定的安全性，且能抵抗 BGE 攻击；另一方面增加了外部编码，使得整个算法具有完整性。不过，改进算法二不能抵抗林等人的攻击方法，可以采用其方法在 $<2^{47}$ 的时间复杂度恢复出轮密钥。

# 第五章 白盒 SMS4 算法的软件实现

本章在改进算法二的基础上，对一些参数进行简化，完成白盒 SMS4 算法的软件实现，并在 MFC 上开发出可视化加解密应用软件。这一章首先简单介绍软件实现的需求分析、整体结构以及开发平台，然后具体介绍各部分的具体实现，最后作出系统界面功能演示及说明。

## 5.1 需求分析与整体结构

### 5.1.1 需求分析

根据白盒 SMS4 算法的特点，结合白盒 SMS4 算法与原 SMS4 算法的加解密及轮过程的输出做对比验证的需求，本文开发的白盒 SMS4 加/解密应用软件包括的功能如下：

（1）实现参数的动态配置；

（2）由初始密钥和明文分组经白盒 SMS4 加密过程加密得到密文分组；

（3）白盒 SMS4 加密过程的轮输出；

（4）由初始密钥和密文分组经白盒 SMS4 解密过程加密得到明文分组；

（5）白盒 SMS4 解密过程的轮输出；

（6）由初始密钥和明文分组经 SMS4 加密过程加密得到密文分组；

（7）SMS4 加密过程的轮输出；

（8）由初始密钥和密文分组经 SMS4 解密过程加密得到明文分组。

（9）SMS4 加密过程的轮输出；

本文白盒 SMS4 算法的软件实现是在 Microsoft Visual Studio（简称 VS）中建立 MFC（Microsoft Foundation Classes，微软基础类库）项目完成的，采用的版本为 Visual Studio 2013 版本。

### 5.1.2 整体结构

白盒 SMS4 算法的软件实现包括以下几个部分：参数配置，初始密钥经过密钥扩展函数得轮密钥，白盒 SMS4 算法加密及轮函数输出过程，白盒 SMS4 算法解密及轮函数输出过程，SMS4 算法加密及轮函数输出过程，以及 SMS4 算法解密及轮函数输出过程。其总体结构如图 5-1 所示。

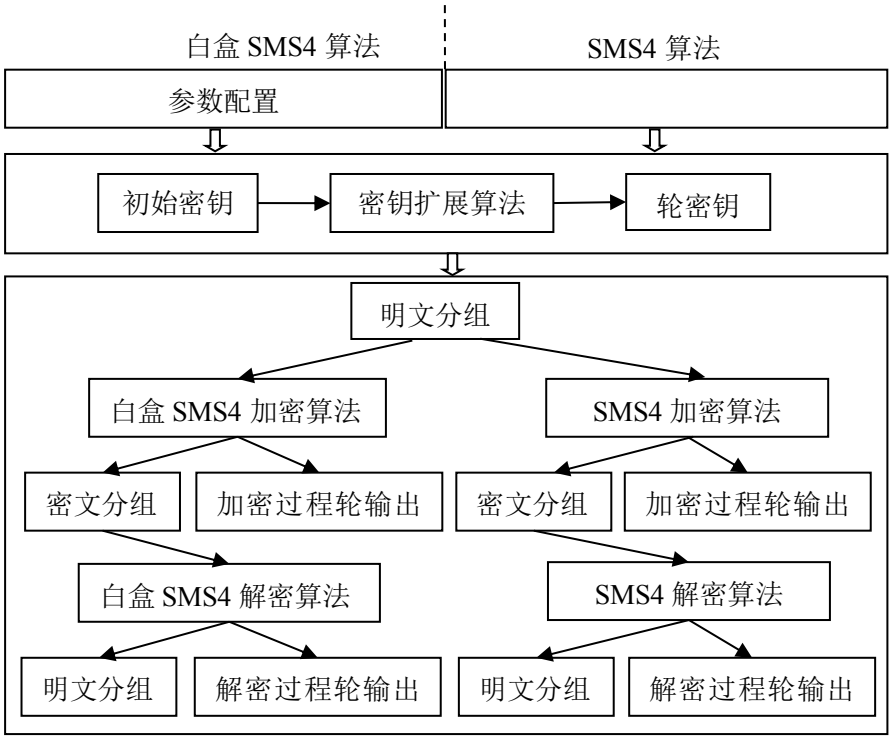


图 5-1 白盒 SMS4 算法的软件实现的整体结构图

## 5.2 算法模块设计与实现

白盒 SMS4 算法的软件实现过程包括以下几个部分：参数配置，查找表实现、加/解密。下面具体介绍这几部分的实现过程。

### 5.2.1 参数配置

白盒 SMS4 算法中，参数主要有 $M_{i+j}=E_i^{-1}\circ P_{i+j}^{-1}$，$D_i=P'_{i+4}\circ P_i^{-1}$，$C_i=P''_{i+4}\circ Q_i^{-1}$，其中，$E_i=diag\left(E_{i0},E_{i1},E_{i2},E_{i3}\right)$，每个 $E_{ij}$ 均为 $GF\left(2\right)$ 上 8 比特到 8 比特的可逆仿射变换，$Q_i$ 是 32 比特到 32 比特的可逆仿射变换；根据本文改进算法二，$P_{i+j}$ 是 $32\times32$ 的可逆矩阵，$P'_{i+4}=P''_{i+4}$ 均为 32 比特到 32 比特的可逆仿射变换，且 $P'_{i+4}\left(x\right)=P''_{i+4}\left(x\right)=P_{i+4}\cdot x\oplus a_{i+4}$。

由于参数 $E_i$、$Q_i$、$P_{i+j}$ 和 $P'_{i+4}$ 都是随机选择的 $GF\left(2\right)$ 上的 $32\times32$ 的可逆矩阵或仿射变换。因此，要随机产生有限域 $GF\left(2\right)$ 上的 $32\times32$ 的可逆矩阵以及 32 比特的列向量，并保存参数 $M_{i+j}$、$D_i$ 和 $C_i$，其计算方法分别如下：

$$\begin{aligned} M_{i+j} &= E_i^{-1} \circ P_{i+j}^{-1}\left(X_{i+j}\right) \\ &= E_i^{-1}\left(P_{i+j}^{-1} \cdot X_{i+j}\right) \\ &= \left(l[E_i^{-1}] \cdot P_{i+j}^{-1}\right) \cdot \left(X_{i+j}\right) \oplus l[E_i^{-1}] \cdot c[\mathrm{E}_i] \end{aligned}$$

即 $M_{i+j}$ 的线性部分为 $l[E_i^{-1}] \cdot P_{i+j}^{-1}$，常数项为 $l[E_i^{-1}] \cdot c[E_i]$；

$$\begin{aligned} D_i &= P'_{i+4} \circ P_i^{-1}\left(X_i\right) \\ &= l[P'_{i+4}] \cdot P_i^{-1}\left(X_i\right) \oplus a_{i+4} \\ &= \left(P_{i+4} \cdot P_i^{-1}\right)\left(X_i\right) \oplus a_{i+4} \end{aligned}$$

即 $D_i$ 的线性部分为 $P_{i+4} \cdot P_i^{-1}$，常数项为 $a_{i+4}$；

$$\begin{aligned} C_i &= P'_{i+4} \circ Q_i^{-1}\left(Y\right) \\ &= P'_{i+4}\left(l[Q_i^{-1}]\left(Y\right) \oplus l[Q_i^{-1}] \cdot c[Q_i]\right) \\ &= l[P'_{i+4}] \cdot \left(l[Q_i^{-1}]\left(Y\right) \oplus l[Q_i^{-1}] \cdot c[Q_i]\right) \oplus a_{i+4} \\ &= \left(P_{i+4} \cdot l[Q_i^{-1}]\right)\left(Y\right) \oplus P_{i+4} \cdot l[Q_i^{-1}] \cdot c[Q_i] \oplus a_{i+4} \end{aligned}$$

即 $C_i$ 的线性部分为 $P_{i+4} \cdot l[Q_i^{-1}]$，常数项为 $P_{i+4} \cdot l[Q_i^{-1}] \cdot c[Q_i] \oplus a_{i+4}$。

系统随机产生一个 $GF(2)$ 上的 $32 \times 32$ 的可逆矩阵及求逆的过程如下：

（1）随机产生范围在[0，255]的 128 个整数，将每个整数转化为 8 位的二进制数，从上到下从左到右排列，组成 32 阶矩阵 $P$ 的 1024 个元素；

（2）利用初等变换对矩阵 $P$ 进行变换，并判断矩阵 $P$ 是否可逆，若 $P$ 可逆，则保存矩阵 $P$ 的逆矩阵 $P^{-1}$；

（3）若 $P$ 不可逆，再次执行（1）、（2），直到产生可逆的矩阵 $P$，并计算出逆矩阵 $P^{-1}$。

判断 $GF(2)$ 上 $n$ 阶矩阵 $P$ 是否可逆并且计算其逆矩阵的算法如下：

（1）构造增广矩阵 $B = [P \mid E]$，这里 $E$ 为单位阵；

（2）从 $B$ 的 $i = 0$ 行开始，依次判断 $B[i][i]$（$0 \le i \le n-1$）是否为 1，若不为 1，则从第 $j = i+1$ 行查询，若 $B[j][i] = 1$，则交换第 $i$ 行与第 $j$ 行；若从第 $j = i+1$ 行到第 $j = n-1$ 行所有的 $B[j][i]$ 都不为 1，则从 $k = 0$ 行开始再次查询，若同时满足 $B[k][k] = 1$、$B[i][k] = 1$ 以及 $B[k][i] = 1$，则交换第 $i$ 行与第 $k$ 行；

此步骤的目的是尽量将矩阵 $B$ 的所有主对角线元素变换为 1，变换后的矩阵记为 $B_1$；

（3）从矩阵 $B_1$ 的 $i = 1$ 行开始，依次判断 $B_1[i][j]$（$0 \le j < i$）是否为 0，若不

为 0，则 $B_1[i][k] \leftarrow B_1[i][k] \oplus B_1[j][k]$（$0 \le k \le 2 \cdot n-1$）；对变换后的第 $i$ 行，按照步骤（2）的方式从第 $i+1$ 行开始，对每行进行调整，使得主对角线元素变换为 1；

此步骤的目的是尽量将矩阵 $B_1$ 的所有下三角元素变换为 0，变换后的矩阵记为 $B_2$；

（4）从矩阵 $B_2$ 的 $i=0$ 行开始，依次判断 $B_2[i][j]$（$i < j \le n-1$）是否为 0，若不为 0，则 $B_2[i][k] \leftarrow B_2[i][k] \oplus B_2[j][k]$（$0 \le k \le 2 \cdot n-1$）；

此步骤的目的是尽量将矩阵 $B_2$ 的所有上三角元素变换为 0，变换后的矩阵记为 $B_3$；

（5）判断矩阵 $B_3$ 的所有主对角线元素是否为 1，若有 $B_3[i][i]=0$（$0 \le i \le n-1$），则说明矩阵 $P$ 为非满秩矩阵，$P$ 不可逆；反之，则矩阵 $B_3$ 必为 $[E \mid P']$ 的形式，而矩阵 $P$ 的逆即为 $P^{-1}=P'$。

系统随机产生一个 $GF(2)$ 上的 32 比特的常向量的过程是：随机产生范围在[0，255]的 4 个整数，将每个整数转化为 8 位的二进制数，从左到右排列，组成常向量的 32 个元素。

由于参数 $E_i = diag(E_{i0}, E_{i1}, E_{i2}, E_{i3})$，每个 $E_{ij}$ 均为 $GF(2)$ 上 8 比特到 8 比特的可逆仿射变换，因此可以按照类似上述方法先产生 4 个 8×8 的可逆矩阵，并计算它们的逆，然后将这 4 个可逆矩阵组成一个 32×32 的块对角矩阵，最后再随机产生一个 $GF(2)$ 上的 32 比特的常向量。

这样就完成了白盒 SMS4 算法的参数动态配置以及参数保存的过程。

### 5.2.2 查找表实现

白盒 SMS4 算法的密钥是隐藏在查找表中的，也就是说算法中的密钥是事先选择的固定的，因此，在制作查找表之前要先由选定的初始密钥根据 SMS4 算法的密钥扩展函数计算出轮密钥。根据白盒 SMS4 算法中的 Part 2 部分做查找表，每一轮有 4 个查找表，共 128 个查找表。

Part 2 部分中，查找表的对应关系是 $x_{ij} \rightarrow v_{ij}\ (j=0,1,2,3)$，每个 $x_{ij}$ 是 8 比特向量，每个 $v_{ij}$ 是 32 比特向量。将 $X=(x_{i0}, x_{i1}, x_{i2}, x_{i3})$ 经过 $E_{ij}$ 和 $S_{ij}$ 变换后的值记为 $(z_{i0}, z_{i1}, z_{i2}, z_{i3})$，则

$$Z_{ij} = S_{ij}\left(E_{ij}\left(x_{ij}\right)\right) = S_{box}\left(E_{ij}\left(x_{ij}\right) \oplus rk_{ij}\right), j=0,1,2,3 \qquad (5\text{-}1)$$

而又有

$$Y = Q_i \circ L \cdot \begin{bmatrix} z_{i0} \\ z_{i1} \\ z_{i2} \\ z_{i3} \end{bmatrix} = l[Q_i] \cdot \left( L \cdot \begin{bmatrix} z_{i0} \\ z_{i1} \\ z_{i2} \\ z_{i3} \end{bmatrix} \right) \oplus c[Q_i] = \left(R_{i0}, R_{i1}, R_{i2}, R_{i3}\right) \cdot \begin{bmatrix} z_{i0} \\ z_{i1} \\ z_{i2} \\ z_{i3} \end{bmatrix} \oplus c[Q_i]$$

$$= \left(R_{i0} \cdot z_{i0}\right) \oplus \left(R_{i1} \cdot z_{i1}\right) \oplus \left(R_{i2} \cdot z_{i2}\right) \oplus \left(R_{i3} \cdot z_{i3} \oplus c[Q_i]\right)$$

$$= v_{i0} \oplus v_{i1} \oplus v_{i2} \oplus v_{i3}$$

因此，

$$\begin{aligned} v_{ij} &= R_{ij} \cdot z_{ij}, j = 0,1,2 \\ v_{i3} &= R_{i3} \cdot z_{i3} \oplus c\left[Q_i\right] \end{aligned} \tag{5-2}$$

其中，$R_{ij} = \left[l[Q_i] \cdot L\right]_j$ 是 $32 \times 8$ 的矩阵，表示 $l[Q_i] \cdot L$ 的第 $j\left(j = 0,1,2,3\right)$ 个分块。

综合上面的几个式子，可以得到

$$\begin{aligned} v_{ij} &= \left[l[Q_i] \cdot L\right]_j \cdot S_{box}\left(E_{ij}\left(x_{ij}\right) \oplus rk_{ij}\right), j = 0,1,2 \\ v_{i3} &= \left[l[Q_i] \cdot L\right]_3 \cdot S_{box}\left(E_{i3}\left(x_{i3}\right) \oplus rk_{i3}\right) \oplus c\left[Q_i\right] \end{aligned} \tag{5-3}$$

由于式中的 $L$ 为 SMS4 算法中线性变换 $L$ 的矩阵表示，

$$L\left(X\right) = X \oplus \left(X <<< 2\right) \oplus \left(X <<< 10\right) \oplus \left(X <<< 18\right) \oplus \left(X <<< 24\right)$$

设 $E$ 为单位阵，$X <<< 2$ 对应的线性变换矩阵为 $A_1$，$X <<< 10$ 对应的线性变换矩阵为 $A_2$，$X <<< 18$ 对应的线性变换矩阵为 $A_3$，$X <<< 24$ 对应的线性变换矩阵为 $A_4$，则

$$\begin{aligned} L\left(X\right) &= X \oplus \left(X <<< 2\right) \oplus \left(X <<< 10\right) \oplus \left(X <<< 18\right) \oplus \left(X <<< 24\right) \\ &= E \cdot X \oplus A_1 \cdot X \oplus A_2 \cdot X \oplus A_3 \cdot X \oplus A_4 \cdot X \\ &= \left(E \oplus A_1 \oplus A_2 \oplus A_3 \oplus A_4\right) \cdot X \\ &= L \cdot X \end{aligned}$$

最终得到的 $L$ 矩阵为

$$L = \begin{bmatrix}
1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1\\
1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1\\
1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&1\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&1
\end{bmatrix}$$

因此，将$l[Q_i]$、$L$、$E_{ij}$、$rk_{ij}$和$c[Q_i]$的值代入式（5-3），针对$x_{ij}$所有的 128 个取值（$x_{ij}=\{(00000000),(00000001),\cdots,(11111111)\}$），可计算出对应的 128 个$v_{ij}$的值（32 比特的向量表示）。按照这种方式，可做成 128 个查找表。整个 Part 2 部分就转化为查找表格及异或的操作。

### 5.2.3 加/解密

前面两个小节是对白盒 SMS4 算法的加/解密过程的前提工作。为了简化实现过程，这里我们假设每一轮的参数$E_i$、$Q_i$、$P_{i+j}$和$P'_{i+4}$都相同，且每个$P_{i+j}(j=0,1,2,3)$也都相同，我们简化符号表示为：参数$E$、$Q$、$P'$、$M$、$D$和$C$为 32 比特到 32 比特的可逆仿射变换，参数$P$为 32×32 的可逆矩阵，且$P'(x)=P\cdot x\oplus c[P]$，$c[P]$为 32 比特的列向量。

加密过程描述如下：

（1）将 128 bit 的明文分组分成四部分，$X=(X_0,X_1,X_2,X_3)$，用$P$分别对$X_0,X_1,X_2,X_3$做输入置乱编码，得$X'_i=P\cdot X_i\ (i=0,1,2,3)$；

（2）进入轮迭代过程，令$l=0$；

（3）计算

$$\begin{aligned} X_l &= M \circ X'_{l+1} \oplus M \circ X'_{l+2} \oplus M \circ X'_{l+3} \\ &= l[M] \cdot X'_{l+1} \oplus l[M] \cdot X'_{l+2} \oplus l[M] \cdot X'_{l+3} \oplus c[M]; \\ &= l[M] \cdot (X'_{l+1} \oplus X'_{l+2} \oplus X'_{l+3}) \oplus c[M] \end{aligned}$$

（4）将 $X_l$ 分成四部分，$X_l = (x_{l0}, x_{l1}, x_{l2}, x_{l3})$，分别查找第 $l$ 轮四个查找表，得到对应值 $v_{l0}, v_{l1}, v_{l2}, v_{l3}$，计算 $Y_l = v_{l0} \oplus v_{l1} \oplus v_{l2} \oplus v_{l3}$；

（5）分别计算 $X''_l = D(X'_l) = l[D] \cdot X'_l \oplus c[D]$，$Y'_l = C(Y_l) = l[C] \cdot Y_l \oplus c[C]$，计算 $X'_{l+4} = X''_l \oplus Y'_l$；

（6）$l \leftarrow l+1$，判断 $l$ 是否等于 32，若 $l \neq 32$，返回步骤（3），反之，执行（7）；

（7）用 $P^{-1}$ 分别对 $X'_{32}, X'_{33}, X'_{34}, X'_{35}$ 做输出置乱编码，得 $X_i = P^{-1} \cdot X'_i \ (i = 32, 33, 34, 35)$；

（8）对 $X'_{32}, X'_{33}, X'_{34}, X'_{35}$ 做 $R$ 变换得到的 $(X'_{35}, X'_{34}, X'_{33}, X'_{32})$ 即为密文分组。

白盒 SMS4 算法的解密算法与加密算法几乎相同，不同的是解密算法是 128 比特的密文分组解密成 128 比特的明文分组，解密过程的查找表做法是将式（5-3）中的轮密钥 $rk_{ij}$ 换成逆序的。

## 5.3 白盒 SMS4 算法应用软件

### 5.3.1 应用软件功能

白盒 SMS4 算法的软件界面主要包括以下功能：

（1）参数的动态配置，包括系统随机产生和文件选择两种方式，以及参数查看功能；

（2）初始密钥的输入以及轮密钥查看功能；

（3）明文分组的输入、白盒 SMS4 加密以及加密过程轮函数输出查看功能；

（4）密文分组显示、白盒 SMS4 解密以及解密过程轮函数输出查看功能；

（5）SMS4 算法加密以及加密过程轮函数输出查看功能；

（6）SMS4 算法解密以及解密过程轮函数输出查看功能。

白盒 SMS4 算法软件实现的界面如图 5-2 所示。


图 5-2 白盒 SMS4 算法软件实现的界面

### 5.3.2 应用软件实例

下面给出一个白盒 SMS4 算法应用软件的实例。该实例描述了此应用软件的操作过程、选择的参数以及算法执行的结果。

首先进行参数的配置，软件提供两种方式：系统生成和文件选择。选择任意一种方式完成参数的配置，可以点击“查看”按钮查看参数。本例中参数配置如图 5-3，5-4，5-5 所示。

参数 $P$ 是可逆矩阵形式，该实例中取值为图 5-3 中的 $32\times32$ 的可逆矩阵；参数 $P'$ 是可逆仿射变换的形式，该实例中的取值由图 5-3 中 $32\times32$ 的可逆矩阵以及 $32\times1$ 的列向量组成。

参数 $Q$ 是可逆仿射变换形式，该实例中的取值由图 5-4 中 $32\times32$ 的可逆矩阵以及 $32\times1$ 的列向量组成。

参数 $E$ 是可逆仿射变换的形式，且其线性部分为可逆对角矩阵形式，参数 $E$ 在该实例中的取值由上图中 $32\times32$ 的可逆矩阵以及 $32\times1$ 的列向量组成。

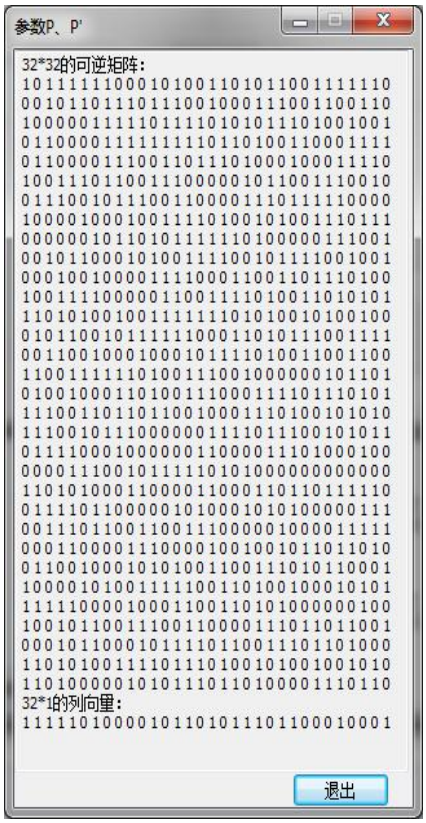


图 5-3 参数 $P$ 、 $P'$ 取值

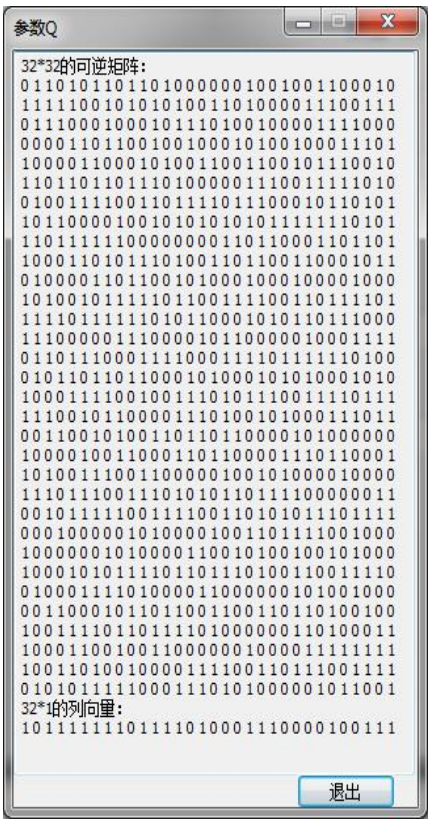


图 5-4 参数 $Q$ 取值

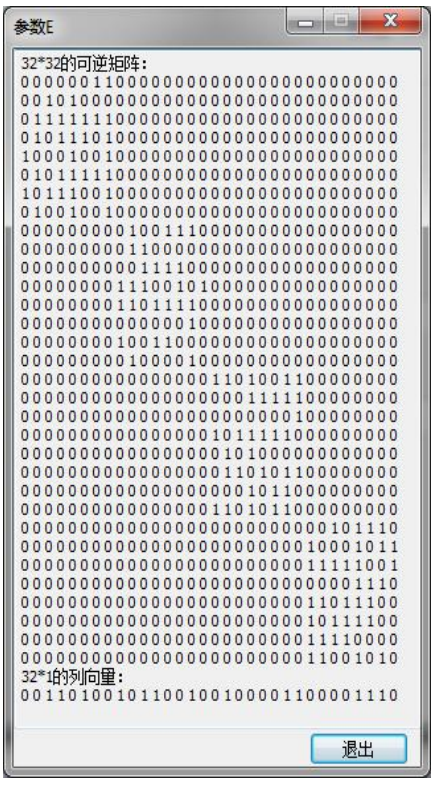


图 5-5 参数 $E$ 取值

其次，输入 128 比特的初始密钥，本例中初始密钥设定为：0123456789abcdef fedcba9876543210（十六进制表示）；查看轮密钥如图 5-6 所示，轮密钥显示顺序为按行依次显示。

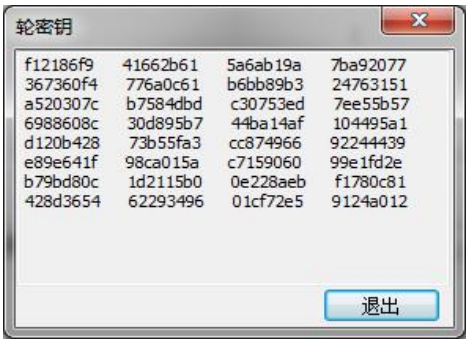


图 5-6 轮密钥

然后，输入 128 比特的明文分组，本例中明文分组设定为：0123456789abcdef fedcba9876543210（十六进制表示）；加密并查看轮函数输出结果如图 5-7 所示，轮函数输出显示顺序为按行依次显示。


图 5-7 白盒 SMS4 算法加密图

最后，对加密后的密文分组解密并查看解密过程轮函数输出结果如图 5-8 所示，轮函数输出显示顺序为按行依次显示。


图 5-8 白盒 SMS4 算法解密图

### 5.3.3 白盒 SMS4 与黑盒 SMS4 算法比较

通常，我们认为密码系统中的发送端是安全可信的，只是客户端运行环境不一定安全可信，又因为白盒算法一般比黑盒算法复杂，执行效率低，因此在使用时一般是在发送端使用黑盒算法进行加密，而在客户端使用白盒算法进行解密。这就要求相同明文及相同密钥的情况下，白盒密码与黑盒密码加密得到的密文及解密得到的明文是一样的。因此，本文开发的白盒 SMS4 算法软件提供了如下功能：1. 改进算法的正确性检验测试；2. 改进算法与 SMS4 算法的执行效率对比。

首先，我们取与白盒 SMS4 算法相同的明文分组和初始密钥，即明文分组设为：0123456789abcdeffedcba9876543210，初始密钥设为：0123456789abcdeffedcba9876543210（均为十六进制表示）。按照 SMS4 加密算法进行加密操作，得到的密文分组及加密过程轮函数输出如图 5-9 所示。


图 5-9 黑盒 SMS4 算法加密图

可知，最终得到的密文为：681edf34d206965e86b3e94f536e4246。


图 5-10 黑盒 SMS4 算法解密图

同样，对得到的密文分组按照 SMS4 解密算法进行解密操作，得到解密出的明文分组以及解密过程轮函数输出如图 5-10 所示。

由此可知，白盒 SMS4 与黑盒 SMS4 算法在明文分组与初始密钥都相同的情况下，得到的密文分组是一样的，不同的是加密过程的轮函数输出以及解密过程中的轮函数输出。这就验证了本文改进算法的正确性。

另外，我们对密码算法的实现效率做了评估，分别测试了黑盒 SMS4 算法与白盒 SMS4 算法的加密过程平均每次加密所需要的时间。

测试环境如下：

处理器：型号 Intel 酷睿 i3 3240，频率 3.4GHz

内存：4G

主板芯片：Intel H61

操作系统：Windows 7

测试过程如下：分别测试了黑盒 SMS4 算法与白盒 SMS4 算法的加密过程，针对一组明文，设置加密次数，求其平均值，得到的结果即为平均一次加密过程所需的时间。表 5-1 为测试结果，表中为 1000000 次加密过程的结果，时间为平均一次加密过程所需的时间，单位为毫秒。

表 5-1 黑盒 SMS4 与白盒 SMS4 平均一次加密过程运行时间对比

| 黑盒 SMS4 加密过程 | 白盒 SMS4 加密过程 |
|---|---|
| 0.00040 | 0.03683 |

测试结果表明，白盒 SMS4 算法比黑盒 SMS4 算法的实现效率要低，主要原因在于查表以及仿射变换。白盒 SMS4 算法一次加密过程需要 128 次查找表、160 次 32 比特到 32 比特的仿射变换以及 8 次 32 比特到 32 比特的双射变换。因此，其运行时间要比黑盒 SMS4 算法的运行时间要多。但是这不能抹杀掉它在白盒攻击环境下所具有的安全性，而从上表的测试结果可知，其所需要的运行时间也是在可以接受的范围之内的。

## 5.4 本章小结

本章介绍了白盒 SMS4 算法的软件实现，首先简单介绍了软件实现的需求分析、总体结构以及开发平台，然后具体介绍各部分的实际与实现，最后给出系统界面及一个实例，并与黑盒 SMS4 加密过程进行对比。

本章的白盒 SMS4 算法的软件实现是在本文第四章白盒 SMS4 改进算法二的

基础上，对一些参数进行简化，并在 MFC 上作出系统界面完成的。该软件实现过程包括以下几个功能：参数动态配置，初始密钥的输入以及轮密钥查看功能，明文分组的输入、加密以及加密过程轮函数输出查看功能，密文分组显示、解密以及解密过程轮函数输出查看功能。

本章还给出了一个白盒 SMS4 算法的一个实例，并将它与黑盒 SMS4 算法的加密过程进行对比。结果表示，白盒 SMS4 与黑盒 SMS4 算法在明文分组与初始密钥都相同的情况下，得到的密文分组是一样的。验证了算法的正确性。

# 第六章 总结与展望

## 6.1 本文总结

随着密码攻击方式的发展，传统的黑盒密码攻击模型显得越来越脆弱。与传统的密码攻击模型不同，白盒攻击模型赋予了攻击者更多的能力，在此模型下，攻击者对密码算法的运行拥有完全的控制权，并且能够观察及更改软件运行时的数据。能够抵抗白盒攻击的密码算法称为白盒密码。本文对白盒密码以及白盒SMS4 算法的设计与实现进行了系统的研究，主要的研究和结论包括：

首先，对肖-白盒 SMS4 算法进行系统学习和分析。肖-白盒 SMS4 算法采用查找表与仿射变换相结合的方式，整个算法共需执行 128 次表格查找和 160 次仿射变换。白盒 SMS4 算法是在原本 SMS4 算法的基础上做的白盒设计，需要占用额外的空间，且执行效率比原始 SMS4 算法的慢，因此，在实际应用时需要对系统资源及运行环境等因素做出综合分析。从白盒多样性和白盒含混度的角度来看，肖-白盒 SMS4 算法能够抵抗穷举攻击；同时，肖-白盒 SMS4 算法也能有效的抵抗 BGE 攻击。然而，林婷婷等人利用将 BGE 攻击、差分分析法及求解方程组等方法相结合的方法，成功的恢复出了肖-白盒 SMS4 算法的轮密钥。尽管肖-白盒 SMS4 算法被破译了，但是它仍具有比原 SMS4 算法更高的安全性，其破译方法也为我们研究具有更高安全性的白盒 SMS4 算法提供了思路。

其次，提出肖-SMS4 白盒设计的两种改进算法，并分别对其实现复杂度、安全性等进行了分析。改进算法一将肖-白盒 SMS4 算法中的四个查找表改为两个查找表，节省了查表次数，但是它增大了额外的空间，不过，从安全性方面来讲，改进算法一可以将攻击时间复杂度至少提高 2 倍。改进算法二一方面简化了肖-白盒 SMS4 算法中的某些参数，其好处是在计算保存参数的过程中会减小计算的复杂度，而简化参数后的白盒多样性和白盒含混度仍能满足一定的安全性；另一方面增加了外部编码，使得整个算法具有完整性。

最后，针对改进的白盒 SMS4 算法，完成白盒 SMS4 算法的软件实现，并在 MFC 上开发出可视化加解密应用软件。本文还给出了白盒 SMS4 应用软件的一个实例，结果表明，白盒 SMS4 与黑盒 SMS4 算法在明文分组与初始密钥都相同的情况下，得到的密文分组是一样的。并且，本文给出的白盒 SMS4 与黑盒 SMS4 算法的实现效率测试数据显示，本文中的白盒 SMS4 加密过程的速度大约是黑盒 SMS4 加密过程的 100 倍。虽然这个结果在需要高安全性的情况下可以被接受，但是白盒 SMS4 算法在设计上还有很大的提升空间。

## 6.2 进一步研究工作

白盒密码是密码学理论研究的一个新方向。白盒攻击颠覆了传统密码学对攻击者能力的很多限制，更加符合实际应用环境中的安全威胁。目前，白盒密码的研究仍处于初级阶段。

一方面，白盒密码的理论研究尚不成熟。首先，白盒密码没有一个统一的理论基础，现有的研究方向都是个人对于白盒密码的理解和摸索；其次，没有一个统一的评价标准来衡量白盒密码的安全性，现有的几个评价白盒密码安全性的指标都只适用于基于特定技术下的白盒密码，而这些衡量标准在密码学理论上也没有明确的安全性证明；再次，目前的白盒密码都是基于现有的分组密码，对其进行白盒设计实现的，而能否设计出一种适用于白盒攻击环境下的新的白盒算法也未可知。

另一方面，针对白盒密码在实际应用中的研究也不太理想。在实际应用中，安全性与运行效率都很重要，很多情况下需要在二者之间权衡。白盒 SMS4 算法虽然比传统的 SMS4 算法安全性高，但是它需要占用额外的空间，运行效率也比较低，因此，在保证安全性的情况下研究更有效的白盒密码算法或者更高效的白盒设计方法以便于更好的在实际中应用是今后白盒密码的一个研究方向。另外，目前的白盒密码算法中的密钥都是固定的，若长时间使用该密码算法其安全性无法保障，因此能否在白盒密码中采用密钥更新机制，利用密钥的更新提高其安全寿命，这也将成为未来白盒密码研究的一个研究方向。

# 致 谢

两个月后，我将告别我的母校，我的导师，踏向新的旅途。多情自古伤离别，就让我勾勒出最后的一笔，我要致谢曾经给予我帮助与关怀的人。

黑发积霜织日月，粉笔无言写春秋。伴随着成都无声的细雨，我的耳畔似乎又传来李胜强老师的谆谆教导。初见李老师，亲切，和蔼，严肃中又不失幽默，总是在关键的时刻让我避开捷径，在我一筹莫展时，挥洒画龙点睛之笔。三年磨一剑，时光褪去我的娇弱与稚嫩，我也将昂首阔步，越走越远。桃花潭水深千尺，不及导师赠我情。

在漫漫的项目之路，团队周亮老师渊博的知识和严谨的态度，让我受益终生。在他的身上，我更加理解了什么是博览群书，思维缜密。我很感激，也很庆幸能遇到如此良师。

当然少不了和我一个教研室的徐丹、刘庭廷、周林、韩承昊等同学以及同门师兄陈嘉鑫在我遇到挫折时伸出的援助之手。还有我可爱的小伙伴吴玉香、李文、田佳佳等，感谢陪我度过快乐的研究生时光。

感谢在我背后默默支持我的家人，他们无微不至的关怀着我，是我生命中坚实的后盾，让我可以在这喧嚣浮躁的尘世中，宁静致远。

# 参考文献


[1] Christian Collberg,Jasvir Nagra 著.崔孝晨 译.软件加密与解密[M].人民邮电出版社, 2012.5.

[2] Anderson R,Kuhn M. Low cost attacks on tamper-resistant devices. In: Proc. of the 5th Int'l Workshop on Security Protocols. LNCS 1361,Springer-Verlag, 1997.125-136.

[3] Biham E,Shamir A. Differential fault analysis of secret key cryptosystems. In: Proc. of the 17th Annual Int'l Cryptology Conf. on Advances in Cryptology. New York,1997.513-525.

[4] Biham E,Shamir A. Power analysis of the key scheduling of the AES candidates. In: Proc. of 2nd AES Candidate Conf. Rome, 1999.22-23.

[5] Boneh D,DeMillo RA,Lipton RJ. On the importance of eliminating errors in cryptographic computations. Journal of Cryptology, 2001,14(2):101-119.

[6] Chow. S. and Eisen. P. and Johnson. H. et al.. White-Box Cryptography and an AES Implementation, Proceedings of the Ninth Workshop on Selected Areas in Cryptography.2002.

[7] 肖雅莹, 来学嘉. 白盒密码及 SMS4 算法的白盒实现[中国密码学会 2009 年会论文集]. 2009.

[8] Billet O,Gilbert H,Ech-Chatbi C. Cryptanalysis of a white box AES implementation[C]. In: Selected Areas in Cryptography. Springer Berlin Heidelberg,2005: 227–240

[9] 林婷婷, 来学嘉.对白盒 SMS4 实现的一种有效攻击[J].软件学报, 2013(9).

[10] Hohl F. Time limited black box security: protecting mobile agents from malicious hosts[C]. In: Mobile Agents and Security. Springer Berlin Heidelberg, 1998: 92–113.

[11] Sander T,Tschudin C F. Protecting mobile agents against malicious hosts[C]. In: Mobile Agents and Security. Springer Berlin Heidelberg, 1998: 44–60.

[12] Kocher P C. BTiming attacks on implementations of Diffie-Hellman[C]. In: CRYPTO '96.Springer Berlin Heidelberg，1996: 104–113.

[13] Chow S,Eisen P,Johnson H.et al.. A white-box DES implementation for DRM applications[C]. In: Digital Rights Management. Springer Berlin Heidelberg, 2003: 1–15.

[14] Jacob M,Boneh D,Felten E. Attacking an obfuscated cipher by injecting faults[C]. In: Digital Rights Management. Springer Berlin Heidelberg, 2003: 16–31.

[15] Link H E,Neumann W D. Clarifying obfuscation: improving the security of white-box DES[C]. International Conference on Information Technology: Coding and Computing—ITCC 2005. IEEE, 2005: 679–684.

[16] Bringer J,Chabanne H,Dottax E. White box cryptography: another attempt[J] . IACR Cryptology ePrint Archive,2006: 468.

[17] Wyseur B,Michiels W,Gorissen P.et al.. Cryptanalysis of white-box DES implementations with arbitrary external encodings[C]. In: Selected Areas in Cryptography. Springer Berlin Heidelberg, 2007: 264–277.

[18] 肖雅莹. 白盒密码及 AES 与 SMS4 算法的实现[M].上海交通大学. 2010.

[19] Wyseur B. White-Box Cryptography[D]. Doctoral thesis, Katholieke Universiteit Leuven, 2009.

[20] De Mulder Y,Wyseur B,Preneel B. Cryptanalysis of a perturbated white-box AES implementation[C]. In: Progress in Cryptology—INDOCRYPT 2010. Springer Berlin Heidelberg, 2010: 292–310.

[21] De Mulder Y,Roelse P,Preneel B. Cryptanalysis of the Xiao–Lai white-box AES implementation[C]. In: Selected Areas in Cryptography. Springer Berlin Heidelberg, 2013: 34–49.

[22] Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. Advances in Cryptology -EUROCRYPT'93,LNCS 765,pp:386-397，Springer-Verlag, 1993.

[23] Mitsuru Matsui. The First Experimental Cryptanalysis of the Data Encryption Standard. Advances in Cryptology-EUROCRYPT'94,LNCS 839，pp:1-11,Springer-Verlag,1994.

[24] Biham E, Shamir A. Differential Cryptanalysis of DES-like Crypt0systems. Advances in Cryptology-EUROCRYPT'90,LNCS 537，pp:2-21,Springer-Verlag,1990.

[25] Knudsen L.R.. Cryptanalysis of LOKI91,Advances in Cryptology-Auscrypt'92,LNCS 718，pp:196-208,Springer-Verlag,1993.

[26] Eli Biham. New Types of Cryptanalytic Attacks Using Related Keys, Journal of Cryptology, Vol. 7,N0.4,pp:28-40. Springer-Verlag,1994.

[27] Jakimoski G, Desmedt Y, Related-Key Differential Cryptanalysis of 192-bit Key AES Variants. Selected Areas in Cryptology-SAC 2003,LNCS 3006,PP:208-221,Springer-Verlag,2004.

[28] Paul C. Kocher, Timing Attacks on Implementations of Diffie-Hellman,RSA,DSS,and Other Systerm. Advances in Cryptology-CRYPTO'96,LNCS 1109,pp:104-113,Springer-Verlag,1996.

[29] Susan Hohenberger,Guy Rothblum,Abhi Shelat et al.. Securely Obfuscating Re-Encryption. In proceedings of 4th Theory of Cryptography Conference(TCC2007),volume 4392 of Lecture Notes in Computer Science,pages 233-252. Springer-Verlag,2007.

[30] Biryukov A, Bouillaguet C,Khovratovich D. Cryptographic schemes based on the ASASA structure: black-box,white-box,and public-key[C]. In: Advances in Cryptology-ASIACRYPT 2014. Springer Berlin Heidelberg,2014: 63–84.

[31] Courtois N,Klimov A,Patarin J,et al.. Efficient algorithms for solving over defined systems of multivariate polynomial equations[C]. In: Advances in Cryptology—EUROCRYPT 2000. Springer Berlin Heidelberg,2000: 392–407.

[32] 王冰. 白盒密码的设计方法和安全性分析[M].上海交通大学.2011.

[33] C.E.Shannon. Communication Theory of Secrecy Systems, Bell System Technology Journal, Vol.28,pp:656-715,1949.

[34] 黄橙. 分组密码算法实现效率研究[M].四川大学.2005.

[35] 张胜元. $Z_n$ 上 m 阶可逆矩阵的计数[J].福建师范大学学报(自然科学版).15（1）:13-15,1999.

# 硕士研究生期间的研究成果

## 参与的科研项目：

[1] 可配置参数信道编译码技术，中国电子科技集团公司第十研究所合作项目，主研。

## 申请的专利：

[1] 李胜强，尚培. 一种 RS 码频域快速译码方法.申请号：201510590917.0


# 硕士学位论文

MASTER THESIS