华南理工大学

South China University of Technology

博士学位论文

几种多变量公钥密码方案的改进方案及其安全性研究

作者姓名	彭峙酿
学科专业	计算机科学与技术
指导教师	唐韶华 教授
所在学院	计算机科学与工程学院
论文提交日期	2017 年 10 月

Improvements and Security Analysis for Several Multivariate Cryptography Schemes

A Dissertation Submitted for the Degree of Doctor of Philosophy

Candidate：Peng Zhiniang

Supervisor：Prof. Tang Shaohua

South China University of Technology

Guangzhou, China

分类号： TP391 **学校代号：10561**

学　号： 201310102112

华南理工大学博士学位论文

几种多变量公钥密码方案的改进方案及其安全性研究

作者姓名：彭峙酿　　指导教师姓名、职称： 唐韶华　教授

申请学位级别：工学博士　　学科专业名称：计算机科学与技术

研究方向：信息安全

论文提交日期： 2017 年 10 月 16 日　论文答辩日期：　　年　月　日

学位授予单位：华南理工大学　　学位授予日期：　　年　月　日

答辩委员会成员：

主席：_____

委员：_____

华南理工大学
学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：　　日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属华南理工大学。学校有权保存并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文被查阅（除在保密期内的保密论文外）；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。本人电子文档的内容和纸质论文的内容相一致。

本学位论文属于：

□保密（校保密委员会审定为涉密学位论文时间：___年__月__日），于___年__月__日解密后适用本授权书。

□不保密,同意在校园网上发布，供校内师生和与学校有共享协议的单位浏览；同意将本人学位论文提交中国学术期刊(光盘版)电子杂志社全文出版和编入CNKI《中国知识资源总库》，传播学位论文的全部或部分内容。

（请在以上相应方框内打“√”）

作者签名：　　日期：

指导教师签名：　　日期

作者联系电话：　　电子邮箱：

摘　要

目前主流的公钥密码学方案安全性基本上都基于大数分解问题、离散对数问题和椭圆曲线上的离散对数问题。1993 年，Shor 提出了在量子计算机上解决大整数分解和离散对数问题的多项式时间算法，这对目前流行的公钥密码方案构成的严重的安全威胁。能够抵抗量子计算机攻击的后量子密码成为了一个重要的密码学研究方向。多变量公钥密码（简称 MPKC）是目前最受关注的后量子密码之一。

过去三十年中，多变量公钥密码得到了快速发展，许多加密和签名方案被提出。多变量公钥密码的公钥一般由有限域上非线性多变量多项式组构成。其加密和签名验证过程只需要进行简单的有限域上多项式带入求值计算，速度非常快，十分适用于在资源有限环境中使用。目前并没有有效的量子算法能够对多变量公钥密码构造威胁，所以多变量公钥密码是一个具有极高研究价值的密码学研究方向。

然而，多变量公钥密码的发展并不是一帆风顺的。目前还存在一些问题待解决。多变量公钥密码密钥过长影响其实际使用，解密和签名生成过程相对于加密和签名验证过程速度过慢，安全性缺乏足够的分析，安全的多变量加密方案和适用于传感器网络的多变量公钥密码方案十分缺乏。本文首先关注目前多变量密码密钥过长和解密、签名速度相对较慢的问题。提出了循环 UOV、循环 Rainbow 两个签名方案和循环 SRP 加密方案。本文对这些方案进行了详细的安全性分析和性能对比。最后，本文还提出了一种适用于无线传感网络的循环 UOV 在线离线签名方案。

在第三章中，本文提出了一种新型的 UOV 变种签名方案，称之为循环 UOV。本文在 UOV 中心映射中插入了部分旋转关系。这些旋转关系在 UOV 签名生成过程中能够生成循环矩阵从而提高 UOV 签名速度同时降低 UOV 私钥大小。在安全性上，本文分析了旋转关系的引入给 UOV 中心映射带来的影响。并分析了各种针对 UOV 的攻击方法对其参数选择的影响。在性能上，本文分析了有限域上随机循环矩阵可逆概率并给出了相应公式，并将循环 UOV 与普通 UOV 签名方案进行了实验对比。从理论和实验两方面，证明了方案的高效性。

在第四章中，本文首先分析了当前几种稀疏密钥的 Rainbow 变种方案的安全性，并修正其安全性参数。然后本文提出了一种新型的 Rainbow 变种签名方案，称之为循环 Rainbow。该方案可以看成是循环 UOV 的扩展方案。本文在 Rainbow 中心映射的每层中都插入了部分旋转关系。这些旋转关系在 Rainbow 签名生成过程中，能够在每一

层求逆过程中产生一个循环矩阵，从而提高 Rainbow 签名速度同时降低 Rainbow 私钥大小。在安全性上，本文分析了多层循环关系的引入给 Rainbow 中心映射带来的影响。并着重分析了低秩攻击和高秩攻击对新方案的影响。在性能上，本文结合理论分析和实验对比，证实了方案的高效性。

在第五章中，本文提出了一种新型的 SRP 变种加密方案。本文首先分析了 SRP 加密方案中存在冗余油醋多项式的原因，然后提出新的方法除去这些冗余油醋多项式。这能极大降低 SRP 加密方案的公私钥大小，同时提高加解密速度。同时本文在 SRP 中心映射中插入了部分旋转关系。这些旋转关系能够在 SRP 解密过程中生成循环矩阵，进一步提高 SRP 的解密速度并降低 SRP 的私钥大小。在安全性上，本文分析了秩攻击、差分攻击、线性攻击对 SRP 方案参数选择的影响。在性能上，本文将循环 SRP 方案与多种公钥加密方案进行实验对比，论证了方案的高效性。

在第六章中，本文提出了一种适用于无线传感网络的循环 UOV 在线离线签名方案。该方案将目前无线传感网络中的能量收集技术与循环 UOV 的签名过程相结合，能够有效的降低无线传感节点签名的实时能耗与延迟。本文使用了仿真和实验技术，证实了该方案的可行性。

关键词： 多变量公钥密码；UOV；Rainbow；SRP；无线传感网络

Abstract

Nowadays, most of the public key cryptography schemes are based on the integer factor problem, discrete logarithm problem or discrete logarithm on elliptic curve. In 1993, Shor proposed some polynomial time algorithms to solve integer factor problem and discrete logarithm on quantum computers, which poses a serious threat to the popular public key cryptography. After that, Post-Quantum-Cryptography, which is secure against attacks by quantum computer, became a very important research area. Multivariate Public Key Cryptography (MPKC) is one of the most promising candidates in Post-Quantum-Cryptography.

Over the past three decades, MPKC has undergone a rapid development, and many encryption and signature schemes have been proposed. The public key of the multivariable public key cryptography generally consists of some nonlinear multivariable polynomials over finite field. The encryption and verification process only requires exultations of those polynomials, which is extremely fast. Therefore, MPKC is very suitable for resource-limited environment. At present, there is no effective quantum algorithm can harm the security of multivariable public key cryptography, so the multivariable public key cryptography is a research direction with high value.

However, MPKC currently has some problems need to be solved. The key size of MPKC is always too large for many applications, decryption and signing process of MPKC are much slower compared with encryption and verification process of MPKC, the security of MPKC is not well understood, and there is no practical MPKC scheme suitable for wireless sensor networks. In this paper, I first focus on the first two problem. I propose Circulant UOV and Circulant Rainbow with faster signing process and smaller private key size, and then I propose Circulant SRP with faster decryption process and smaller private key size. Then I carefully analyze the security of those schemes and give overall comparisons with their corresponding schemes. At last, I propose an online/offline UOV signature scheme, which is suitable for wireless sensors network.

In chapter 3, I propose a new UOV variant with faster signing process and smaller private key size. I introduce some rotating relations into small parts of UOV central map. Those rotating relations will result in a circulant matrix during the signing process. This will improve the

signature generation speed while reducing the private key size. In terms of security, I analyze the impact of rotating relations in UOV central map and analyze its security against various attacks on UOV. In terms of performance, I give the formula of full rank probability of a random circulant matrix over finite field, and compare the performance of our new UOV variant with regular UOV. I prove the efficiency claims from theoretical and experimental aspects.

In chapter 4, I first analyze the security of a few Rainbow variant with sparse private key and revise their security parameters. Then I propose a new Rainbow variant with faster signing process and smaller private key size. It can be viewed as an extension of our new UOV variant. I introduce rotating relations, which will result in a circulant matrix during the signing process, into small parts of Rainbow central map. In terms of security, I analyze the impact of rotating relations in Rainbow central map and analyze its security against various attacks on Rainbow. In terms of performance, I combine the theoretical analysis and experimental comparison to demonstrate the efficiency of our new Rainbow variant.

In chapter 5, I propose a new SPR variant with faster decryption algorithm and smaller private key size. At first, I analyze the reason for the existence of redundant Oil-Vinegar polynomials in SRP, then I propose a new way to reduce them. This can help us to improve the speed of encryption and decryption as well as reduce the key size of SRP. After that, I insert some rotating relations in small parts of SRP central map. These rotating relations will help us to get a circulant matrix rather than normal matrix during the decryption process, thus to further improve decryption speed while reducing the private key size. I analyze the security of our new SRP variant against rank attack, differential attack and linear attack. In order to demonstrate the efficiency claims, I compare our new SRP variant with other encryption schemes through experiments.

In chapter 6, I propose an online/offline UOV signature for wireless sensor network. It combines energy harvesting technology with precomputation technique to reduce run-time latency and energy consumption of wireless sensor node. The security of this scheme is equivalent to Circulant UOV and the performance is confirmed by both simulation and practical experiments.

Keywords: MPKC; UOV; Rainbow; SRP; Wireless Sensor Networks

目　录

表格目录

插图目录

第一章　绪论

1.1　研究背景

在传统对称密码算法中，通讯双方需要共享一个密钥。用户 A 想与用户 B 建立可信信道进行通讯必须事先共享一个密钥。这样的可信信道建立的成本非常高。公钥密码学的出现，彻底改变了这一尴尬的状况。1976 年，Bailey Whitfield Diffie 和 Martin Edward Hellman 提出了 Diffie-Hellman 密钥交换协议 [1]，它是第一个实用的在非保护信道中创建共享密钥方法。随后 RSA 公钥加密方案被 Ron Rivest、Adi Shamir 和 Leonard Adleman 一起提出 [2]。Diffie-Hellman 密钥交换协议和 RSA 算法的出现，揭开了公钥密码发展的篇章。时至今日，互联网高度发达，公钥密码学已经渗入到人们生活的方方面面。

在公钥密码方案中，密钥分为公钥部分和私钥部分。私钥只有合法用户自己才拥有，它能够被用来解密密文或对消息进行签名。公钥则是对所有用户都公开，它可以被用来加密消息或者验证签名的正确性。这样的非对称构造允许人们在公开的信道中保证消息的正确性和隐私性。

公钥密码方案的安全性通常依赖于一些数学上的困难问题。目前主流的公钥密码方案包括基于大整数分解问题的方案（如 RSA）、基于离散对数问题的方案（Diffie-Hellman 密钥交换协议）、基于椭圆曲线上离散对数问题的方案（ECDSA）[3]。Shor 提出了量计算机上在多项式时间内解决大整数分解问题、离散对数问题、椭圆曲线上离散对数问题的算法 [4][5]。这些量子算法的出现，严重威胁了目前主流公钥密码算法的安全基础。

一旦量子计算机出现，目前大多数公钥密码方案都不再安全。这种不安全不仅威胁到未来的网络安全，它也会威胁到当前网络中通讯数据的安全性。对于公钥加密方案，如果攻击者在当前时间存储一些公钥加密方案通讯的数据，在将来量子计算机出现的时候，攻击者就能解开当前存储的数据。对于公钥签名方案，攻击者在量子计算机出现后攻破了某个签名算法后，攻击者就可以伪造某个过去时间段的数字签名。由此可见，解决量子计算机对当前公钥密码方案的威胁是当前十分紧迫和重要的研究方向。

能够抵抗量子计算机攻击的公钥密码方案被称为后量子密码方案 [6][7]。目前后量子密码方案主要包括基于格的公钥密码方案 [8]、基于 hash 的公钥密码方案 [9]、基于

编码的公钥密码方案 [10] 和多变量公钥密码方案 [11]。本文主要关注多变量公钥密码方案。

1.2 多变量公钥密码发展情况

1988 年 Tsutomu Matsumoto 和 Hideki Imai 提出了 C*[12] 加密方案。该方案在特征为 2 阶为 q 的基域 K 的一个 n 次扩域 E 上构造一个指数映射 $X \rightarrow X^{q^{\theta}+1}$，然后在其输入和输入两端作用上两个可逆仿射构成公钥系统（一个多变量二次方程组）。任何加密者将某个明文带入该方程组计算求值，便可完成加密运算。拥有私钥的合法用户能够对两个仿射和中心映射进行求逆，进而求解密文。非法用户由于不知道两个可逆仿射的构造，要求逆明文相当于要求解二次多变量方程组，十分困难。C* 方案具有非常快速的加解密速度，备受关注。但 Patarin 等人在 1995 年发现了 C* 方案中的弱点 [13]，提出了针对 C* 方案的线性方程攻击。

为了修复 C* 方案，Patarrin 等人提出了 C*-方案 [14]。通过将减方法运动到 C* 方案中以防止攻击者提取出线性方程。减方法的核心思想是从公钥多项式中删除 r 条式子，从而破坏明密之间的双线性关系，这能保证 C*-方案不再受到线性方程攻击。但减方法的使用，使 C* 方案的公钥系统从单射系统变成多到一映射。这使 C*-方案无法成为一个加密方案，C*-方案是一个数字签名方案。2003 年 C*-方案入选欧洲 NESSIE 密码算法项目 [15]，并被认为是当时低功耗 IC 芯片上的最佳签名算法解决方案。C*- 方案也被称为 SFLASH 方案，先后有 SFLASH^{v1}、SFLASH^{v2} 和 SFLASH^{v3} 等各个版本。不幸的是，Dubois 等人在文章 [16] 发现了映射 $X \rightarrow X^{q^{\theta}+1}$ 存在斜对称关系，使用差分攻击的方法能够恢复出被减方法删除掉的方程，然后攻击者就可以使用线性方程攻击的方法对消息进行签名。这也导致 SFLASH 签名算法被认为不安全，无法满足 NESSIE 的安全要求。

2004 年，Ding 等人提出内部扰动的方法来增强 C* 方案的安全性 [17]。该方案被称为 PMI 加密方案。PMI 方案通过加入内部扰动来扰乱 C* 方案明密文之间的双线性关系，使线性方程攻击无法使用。不幸的是 PMI 方案在 2005 年被 Fouque 等人通过差分攻击攻破 [18]。Fouque 等人通过对其公钥系统进行差分分析，能够还原出其内部扰动的一个核空间，然后利用该空间抵消内部扰动后再对 PMI 方案进行线性方程攻击。为了抵抗差分攻击，Ding 等人又提出了 PMI+ 方案 [19]。其核心思想是在 PMI 中心映射中引入加方法，即增加一些随机的的多变量方程。这样就打破 PMI 公钥系统的差分性

质，保证其安全性。PMI+ 到目前仍然被认为是安全的多变量加密方案。

对 C* 方案的另一个扩展是 HFE (Hidden Field Equation) 加密方案。Patarrin 在 [20] 中提出了 HFE 加密方案。相比较于 C* 的中心映射 $X \rightarrow X^{q^{\theta}+1}$，HFE 的中心映射为 $X \rightarrow \sum a_i X^{q^{s_i}+q^{t_i}}$。不同于 C* 中心映射的求逆，HFE 中心映射求逆时我们需要计算有限域上高次单变量多项式的根。不幸的是，HFE 方案也被攻破了。目前已知有四种针对 HFE 的攻击方法：Shamir-Kipnis 攻击 [21]、Shamir-Kipnis-Courtois[22] 攻击、Courtois[22] 攻击和基于 Grobner 基的直接攻击 [23]。为了提高 HFE 中心映射的安全性，一些基于 HFE 的变种方案被提出。HFE-签名方案方案、HFEv-签名方案 [24] 和 ZHFE 加密方案出 [25][26] 是它们中的代表。ZHFE 在 2017 年被 Cabarcas 等人攻破 [27]，HFE-方案在文章 [28] 被完全攻破。而 HFEv-提出十多年来仍被认为是安全的多变量签名方案。

除了由 C* 方案演化出来的方案之外，多变量公钥密码中相当一部分方案是油醋签名方案演化而言的。油醋签名方案最初由 Patarrin 等人提出 [29]，其核心思路是通过油醋多项式来构造可“求逆”的中心映射。平衡油醋签名方案在 1998 年被 Shamir 等人使用 UOV 攻击攻破 [20]。随后 Patarrin 等人提出了非平衡油醋签名方案 (UOV，Unbalanced Oil and Vinegar) 以抵抗 UOV 攻击。UOV 签名方案到目前为止二十年的时间仍被认为是安全的多变量签名方案。但 UOV 在效率方案离实用还有一定距离，故 UOV 提出之后有许多对其进行的改进方案。Rainbow 签名方案是由 Ding 等人在 2005 年提出的一个多层 UOV 签名方案 [30]，它相比于 UOV 有更快的签名速度、更短的公私钥大小和更短的签名长度。随后从 2010 年开始，人们开始关注于进一步提高 Rainbow 和 UOV 的性能。Cyclic Rainbow，Cyclic UOV 等方案被提出。其核心思想是在 UOV 和 Rainbow 公钥中引入一些循环序列以降低 UOV 和 Rainbow 的公钥大小及签名验证时间。受 Cyclic Rainbow 的启发，MB Rainbow (Matrix based Rainbow)、NT Rainbow 等方案被提出来降低 Rainbow 的私钥大小及签名生成时间。

除了由 C* 和 UOV 派生出来的多变量公钥方案之外，目前还有 ABC 加密方案 [31]、SRP 加密方案 [32]、基于 IP 问题的密钥交换方案等方案备受关注。以上介绍的多变量公钥密码方案的安全性大多数依赖于 MQ 问题和 IP 问题，但并没有可证明的安全性规约。2011 年，Sakumoto 等人提出了可证明安全的多变量身份认证方案 [33]，其安全性可以规约到 MQ 问题（NP 困难）。随后文章 [34] 将该方案扩展为数字签名方案，文章 [35] 将其从基于二次多变量多项式扩展到高次多变量多项式中。但总的来说，目

前具有可证明安全的多变量公钥密码方案的性能都很低。提出安全又实用的多变量公钥密码方案是后量子公钥密码当前的一大挑战 [36]。

1.3 本文的贡献及章节安排

本文主要关注于提高现有多变量公钥密码方案的性能，并分析相应变种方案的安全性。

在第二章中，我们介绍了多变量公钥密码的一些基础知识和基础方案。以便于读者更容易理解全文。

在第三章中，我们提出了一种新型的 UOV 变种方案，我们称之为循环 UOV。我们在 UOV 中心映射油醋多项式的油醋项系数和油变量线性系数中插入一些旋转关系。这些旋转关系的引入能够帮助我们在签名过程中生成更容易求解的线性系统。在安全性上，我们将目前所有针对 UOV 的攻击方法都运用上循环 UOV 上以分析其安全性。同时我们提出变种的高秩攻击以利用循环 UOV 的特殊密钥结构。在性能上，我们提出并证明了有限域上随机循环矩阵的可逆概率公式，这从理论上保证了循环 UOV 的性能。同时我们引入减方法到循环 UOV 中以降低其公钥大小。最后我们将循环 UOV 与其他已知 UOV 变种方案进行对比。

在第四章中，我们首先分析了目前已知的稀疏密钥 Rainbow 签名方案的安全性，并提出一种针对 MB Rainbow 的新型攻击方法。同时我们对 MB Rainbow 和 NT Rainbow 的安全参数进行了修正，以保证其达到预定的安全级别。随后我们将循环 UOV 方案扩展为循环 Rainbow 方案。循环 Rainbow 比循环 UOV 拥有更快的签名速度、更短的公私钥大小和更短的签名长度。在安全性上，我们将目前所有对 Rainbow 进行攻击的方案运用上循环 Rainbow 上，同时提出了针对循环 Rainbow 的变种高秩攻击方法。在性能上，我们将循环 Rainbow 和其他数字签名方案进行对比。

在第五章中，我们提出了一种新型的 SRP 变种加密方案，我们称之为循环 SRP 加密方案。我们首先分析了 SRP 方案中心映射中冗余油醋多项式存在的原因，并提出新的方法除去这些冗余油醋多项式。这能够降低 SRP 方案的公私钥大小并提高其加解密速度。随后我们将旋转关系引入到 SRP 中心映射中，进一步以提高其解密速度。在安全性上，我们从理论和实验两个方向细致的分析了循环 SRP 方案的安全性。在性能上，我们将循环 SRP 方案与多个公钥加密方案进行对比。

在第六章中，我们主要关注于多变量公钥密码的在无线传感网络中的运用。我们

提出了适用于无线传感网络的在线离线循环 UOV 签名方案。并利用能量收集技术和预计算技术降低系统的整体能耗。我们从理论上讨论了方案的计算开销，并用仿真和实验的方法证实了在线离线循环 UOV 签名方案在具有能量收集功能的无线传感网络节点上的适用性。

最后我们对全文内容进行总结。并分享一些多变量公钥密码设计和分析相关的经验和想法，希望能够对关注多变量公钥密码的人有帮助。同时我们给出了未来工作计划和方向，希望更多的人能够参与到改进多变量公钥密码的研究上来。

第二章　多变量密码基础

在本章中，我们将介绍详细介绍多变量公钥密码的相关基础知识。具体的，在第2.1节中，我们将介绍目前多变量公钥密码方案主要基于的困难问题；在第2.2节中，我们将介绍多变量公钥密码方案的一般性构造方法和其分类；在第2.3节，第2.4节和第2.5节中，我们会分别介绍三个与本文相关的多变量公钥密码方案（UOV[37]，Rainbow[30] 和 SRP[32]），在介绍三个多变量公钥密码方案的同时，我们也会加入对这些方案的攻击方法的介绍以分析其安全性；在第2.6节中，我们将对本章进行小结。

2.1　基础定义

这里我们首先介绍多变量密码的一些基础定义和相关知识。

2.1.1　多变量多项式

在具体介绍多变量公钥密码的构造之前，我们先回顾一下多变量多项式的定义。

定义 2.1 **多变量多项式环**: 定义有限域 K 上有 n 个变量的多变量多项式环为

$$K[x_1,...,x_n] = \left\{ \sum_{i=1}^{s} c_i \cdot t_{a_i} | s \in \mathbb{N}, a_i = (a_{i,1},...,a_{i,n}) \in \mathbb{N}_0^n \right\}$$

我们称 $c_i \cdot t_{a_i}$ 为一个项，$c_i \in K$ 是这个项的系数，$t_{a_i} = x_1^{a_{i,1}} \cdot x_2^{a_{i,2}} \cdot ... \cdot x_n^{a_{i,n}}$ 是这个项的单项式。我们将所有在该多项式环中的单项式的集合记为 T^n。

定义 2.2 **单项式次数**: 单项式$t_{a_i} = x_1^{a_{i,1}} \cdot x_2^{a_{i,2}} \cdot ... \cdot x_n^{a_{i,n}}$ 的次数定义为

$$\deg(t_a) = \sum_{j=1}^{n} a_j.$$

而一个多项式$p = \sum_{i=1}^{s} c_i \cdot t_{a_i}$ 的次数定义为

$$\deg(p) = \max_{i \in \{1,...,s\}} \deg(t_{a_i}).$$

在多变量公钥密码中，中心映射系统和公钥系统通常是一个二次多变量多项式方程组。这里我们给出关于 n 个变量的 m 二次多变量多项式方程组系统 P 的形式:

$$\begin{aligned}
P_1(x_1,...,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{1,i,j} \cdot x_i x_j + \sum_{i=1}^{n} p_{1,i} \cdot x_i + p_{1,0} \\
P_2(x_1,...,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{2,i,j} \cdot x_i x_j + \sum_{i=1}^{n} p_{2,i} \cdot x_i + p_{2,0} \\
&\vdots \\
P_m(x_1,...,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{m,i,j} \cdot x_i x_j + \sum_{i=1}^{n} p_{m,i} \cdot x_i + p_{m,0}
\end{aligned}.$$

多项式 $P_i(x_1,...,x_n)$ 的矩阵形式可表示为

$$\begin{bmatrix} p_{i,1,1} & p_{i,1,2} & \cdots & p_{i,1,n} & p_{i,1} \\ 0 & p_{i,2,2} & \cdots & p_{i,2,n} & p_{i,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & p_{i,n,n} & p_{i,n} \\ 0 & 0 & \cdots & 0 & p_{i,0} \end{bmatrix}$$

我们有 $P_i(x_1,...,x_n)$=$(x_1,\cdots,x_n,1)\cdot P_i\cdot(x_1,\cdots,x_n,1)^T$。

定义 2.3 **二次齐次系统**: 如果一个二次多变量多项式系统 P 没有线性项和常数项，那么我们称 P 是一个二次齐次系统。

在多变量密码分析中，线性项和常数项通常不会影响系统安全性。为了方便，我们通常会除去线性项和常数项，认为 $P_i(x_1,...,x_n)$ 是一个二次齐次系统。其矩阵表示 P_i 为

$$P_i = \begin{bmatrix} p_{i,1,1} & p_{i,1,2} & \cdots & p_{i,1,n} \\ 0 & p_{i,2,2} & \cdots & p_{i,2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{i,n,n} \end{bmatrix}$$

且满足 $P_i(x_1,...,x_n) = (x_1,\cdots,x_n)\cdot P_i\cdot(x_1,\cdots,x_n)^T$。

多变量公钥密码的公钥系统 P 通常是一个关于 n 个变量的 m 个多变量二次方程的多变量系统。在针对多变量多项式的密码分析中，我们通常将每条多变量多项式写成矩阵形式，以方便使用一些线性代数的方法对其进行分析。但直接使用上三角矩阵来进行表示容易在运算中丢失掉如秩、核空间等关键信息。通常的，我们使用对称矩阵 $\hat{P_i}$ 来表示多项式 $P_i(x_1,...,x_n)$ 的二次部分。我们称 $\hat{P_i}$ 为多变量多项式 $P_i(x_1,...,x_n)$ 的相关矩阵，其定义为

$$\hat{P_i} = P_i + P_i^T = \begin{bmatrix} 2\cdot p_{i,1,1} & p_{i,1,2} & \cdots & p_{i,1,n} \\ p_{i,1,1} & 2\cdot p_{i,2,2} & \cdots & p_{i,2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{i,1,n} & p_{i,2,n} & \cdots & 2\cdot p_{i,n,n} \end{bmatrix} (i=1,\cdots,m).$$

2.1.2　MQ 问题

在进一步介绍多变量公钥密码的构造之前，我们先介绍其底层的困难问题。

多项式公钥密码系统的安全性核心是基于多项式系统求解问题。

定义 2.4 **多项式求解问题**：给定一个关于 n 个变量 m 条多项式的多项式系统 $P = (p_1, \cdots, p_m)$，要求找到一组解 $\hat{x}_1, \cdots, \hat{x}_n$ 满足

$$\begin{cases} {}_1(x_1, \cdots, x_n) = 0 \\ p_2(x_1, \cdots, x_n) = 0 \\ \quad \vdots \\ p_m(x_1, \cdots, x_n) = 0 \end{cases}$$

高次多变量多项式求解问题被证明是 NP 完全问题，即使是其最简单的形式（GF(2) 上二次多项式方程组求解）也是 NP 完全问题 [38]。虽然少部分多变量密码方案有使用到高次多变量多项式系统，但它们的效率都非常低。被人们所接受和认可的多变量公钥密码系统，通常都是使用二次多变量多项式系统。如果一个多项式求解问题中的多项式 $p_1(x_1, \cdots, x_n), \cdots, p_m(x_1, \cdots, x_n)$ 全部都是二次多项式，那么这样的多项式求解问题被我们称为 MQ 问题。

目前有很多针对 MQ 问题的攻击算法。当 $m \approx n$ 的时候，所有的这些攻击算法都具有指数级别复杂度。

2.1.3 IP 问题

由于多变量公钥密码的构造型式，大多数多变量公钥密码方案的安全性不但依赖于 MQ 问题（Multivariate Quadratic polynomial problem），同时也依赖于 IP 问题 [20]。这里我们给出 IP 问题的三种形式。

定义 2.5 **IP1S 问题**：给定非线性多变量系统 P 和 G 满足 $P = G \circ R$，其中 R 为线性映射或仿射，要求找到一个线性映射或仿射 R' 满足 $P = G \circ R'$。

定义 2.6 **IP2S 问题**：给定非线性多变量系统 P 和 G 满足 $P = S \circ G \circ R$，其中 S 和 R 为线性映射或仿射，要求找到线性映射或仿射 S' 和 R' 满足 $P = S' \circ G \circ R'$。

定义 2.7 **扩展 IP 问题**：给定非线性多变量系统 P 满足 $P = S \circ G \circ R$，其中 S 和 R 为线性映射或仿射，G 的构造属于某类特殊的非线性多项式系统 C。要求找到 P 的一个分解 $P = S' \circ G' \circ R'$ 满足 S' 和 R' 为线性映射或仿射且 $G' \in C$。

IP2S 问题主要被用于私钥的中心映射被公开的一些方案，如 C*[12]、HFE[20] 和 Square[39] 加密方案。在一些多变量公钥密码方案中，中心映射是私钥的一部分，这样的时候其安全性则依赖于扩展 IP 问题。UOV 和 Rainbow 签名方案就属于这一类方案。

与 MQ 问题不同，IP 问题的困难性并没有被证明是 NP 完全问题。实际上，一些

特殊构造的多变量密码方案 (如平衡油醋方案 [40]) 的公钥系统 P 的分解是非常容易的 [41]。这也导致很多多变量公钥密码方案并没有可证明安全。

2.2　多变量公钥密码的一般构造方法

前面介绍了多变量公钥密码的基础定义和相关知识，本节我们将介绍多变量公钥密码的一般性构造方法。包括双极型系统和其他系统的构造方法。

2.2.1　双极型系统

多变量公钥密码方案的一般性构造思路是先选取一个可逆中心映射 $G : K^n \to K^m$，然后选取两个可逆线性仿射 $S : K^m \to K^m$ 和 $R : K^n \to K^n$ 来隐藏中心映射 G 的结构。公钥多项式系统是一个复合二次多项式系统 $P = S \circ G \circ R$。因为 S 和 R 的作用，G 的特殊结构被隐藏了起来，攻击者很难区分 P 和随机的二次多变量系统，所以直接对 P 进行“求逆”非常困难。合法用户因为知道 S，R 和 G，所以可以对 P 进行“求逆”。

多变量公钥密码的标准加密/解密和签名/验证流程图如图 2-1所示。

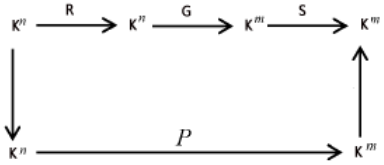

图 2-1 多变量密码双极型结构

在多变量公钥加密系统中，给定明文消息 $\mathbf{m} = (m_1, \cdots, m_n)$，计算

$$\mathbf{c} = P(m_1, \cdots, m_n) = (p_1(m_1, \cdots, m_n), \cdots, p_m(m_1, \cdots, m_n)).$$

$\mathbf{c} = (c_1, \cdots, c_m)$ 则为密文。给定密文的情况下，合法用户可以通过对 $\mathbf{c}$ 依次计算 S，G 和 T 的“逆”，就能得到明文 $\mathbf{m}$。需要说明的是，本文这里所说的“求逆”并不是严格数学意义上的“求逆”。因为公钥系统 P 本身并不一定是一个一对一的映射，所以并不存在逆元的概念。本文中所说的对 P 进行“求逆”是指，给定向量 $\mathbf{y}$ 的值，求解出一个向量 $\mathbf{x}$ 满足 $P(\mathbf{x}) = \mathbf{y}$。

多变量公钥签名系统中，给定一个消息 $\mathbf{m}$ 要计算消息的签名 $\mathbf{s} = (s_1, \cdots, s_n)$。签名者首先计算消息的摘要 $\mathbf{w} = (w_1, \cdots, w_m) = Hash(\mathbf{m})$，然后对 $\mathbf{w}$ 依次计算 S，G 和 T 的“逆”，得到对消息 $\mathbf{m}$ 的签名 $\mathbf{s} = (s_1, \cdots, s_n)$。当验证者拿到消息签名对 $(\mathbf{m}, \mathbf{s})$ 后，计算 $hash(\mathbf{m}) = P(\mathbf{s})$ 是否成立。如果成立，则认为签名正确，否则认为签名错误。

目前绝大多数多变量公钥密码方案都是双极型系统，根据他们中心映射构造的不

同，可以被分成三种类型：

小域方案： 在小域方案中，中心映射 G 的所有组成元素和计算都在小域 K 上，利用特殊的结构来使其容易“求逆”。公钥生成时通过作用上 S 和 R 来隐藏 G 的结构计算出公钥。UOV 和 Rainbow 方案就属于小域方案。本文中我们将主要关于 MPKC 小域方案的改进。

大域方案： 与小域方案不同，大域方案的中心映射通常构造在小域 K 的 n 次扩域 E 上。其具体构造通常是大域上的一个单变量多项式方程 $G : E \to E$，可以使用如柏林坎普算法等相关算法对其进行“求逆”[42]。生成公钥的时候通过 K^n 与 E 之间的同构映射将大域上的中心映射写成小域上的映射，最后复合上仿射 S 和 R 得到公钥系统。常见的大域方案有 C* 系列方案和 HFE 系列方案和 Square 方案等方案。

中域方案： 中域方案的中心映射构造介于小域方案和大域方案之间。它使用到小域 K 到扩域 E 的映射，但是其构造的扩域并非 n 次扩域，而是 $l = \frac{n}{k}$ 次扩域（k 和 l 为整数）。所以此时 E 是一个中等大小的域，我们称之为中域。中域方案利用了特殊的结构来使中域上的中心映射可逆。目前最有名的多变量公钥密码中域方案是 MFE 方案[43]。

2.2.2 其他构造

多变量公钥密码方案除了使用双极型构造以外，还有一些其他构造方式。比较典型的有混合型构造、基于 IP 问题的身份认证和基于 MQ 问题的身份认证。

混合型构造的代表是 Patarin 等人提出的 Dragon 方案 [44]，Shen 等人也提出了混合 UOV 方案（RGB 方案）[45]。混合型构造的核心思想是通过将明文 $\mathbf{x} = (x_1, \cdots, x_n)$ 与密文 $\mathbf{y} = (y_1, \cdots, y_m)$ 混合作用到一个映射 $P : K^{m+n} \to K^l$ 中。我们有 $P(x_1, \cdots, x_n, y_1, \cdots, y_m) = (p_1, \cdots, p_l)$。混合型方案本质上的构造思路还是和双极型相似，需要用在中心映射上作用上仿射来隐藏中心映射的具体结构。所以其安全性也是依赖于 MQ 问题和 IP 问题。由于混合型多变量公钥密码方案比较少，我们这里不深入进行介绍。

在基于 IP 问题的身份认证方案 [20] 中，证明者随机生成两个仿射 $S : K^m \to K^m$ 和 $R : K^n \to K^n$ 作为密钥，并随机生成一个二次多变量中心映射系统 G。计算 $P = S \circ G \circ R$，P 和 G 构成公钥。证明者通过一个基于 IP 问题的零知识论证协议向验证者证明自己拥有密钥 S 和 R，从而实现身份认证。使用 Fiat-Shamir 转换，可以将基于 IP 问题的身份认证方案转换成一个随机预言机模型下可证明安全的数字签名方案，

其安全性依赖于 IP2S 问题。

在基于 MQ 问题的身份认证方案 [33] 中，证明者随机选择二次多变量系统 $P : K^n \to K^m$ 和一个秘密向量 **s**，计算 $\mathbf{v} = P(\mathbf{s})$。$P$ 和 **v** 为公钥，**s** 为私钥。证明者通过一个基于 MQ 问题的零知识论证向验证者证明自己拥有密钥 **s**，从而实现身份证明。使用 Fiat-Shamir 转换，可以将基于 MQ 问题的身份认证方案转换成一个随机预言机模型下可证明安全的数字签名方案，其安全性可以规约到 MQ 问题。

2.3　UOV 签名方案

本节中，我们将介绍 UOV 签名方案和它的安全性分析。同时我们将介绍 UOV 方案的一些变种方案。

2.3.1　基础 UOV 方案

UOV（非平衡油醋）签名方案本质上是 OV（平衡油醋）签名方案 [40][29] 的一个扩展方案。它可以抵抗平衡油醋攻击 [41]，具有很强的安全性 [46]，从提出到现在接近 20 年的时间里仍然保持安全。

为了理解清楚 UOV 签名方案，我们首先介绍油醋多项式

$$g = \sum_{i=1}^{v}\sum_{j=1}^{v} a_{ij}x'_i x'_j + \sum_{i=1}^{o}\sum_{j=1}^{v} b_{ij}\hat{x}_i x'_j + \sum_{j=1}^{v} \beta_j x'_j + \sum_{i=1}^{o} \alpha_i \hat{x}_i + c\,.$$

油醋多项式的变量被分两类：油变量 $\hat{x}_i$ 和醋变量 x'_i。在油醋多项式中，油变量的个数为 o，醋变量的个数为 v。中心映射 $G : K^n \to K^m$ 是由基域 K 上的 o 条油醋多项式组成的。由于油醋多项式系统的特殊结构，我们可以对油醋多项式组成的中心映射进行“求逆”操作。随机的选择 v 个醋变量的值带入到 o 条油醋多项式中的时候，我们能够生成关于 o 个油变量的 o 条线性方程。我们可以使用高斯消元来求解出油变量的值。

当中心映射 $G : K^n \to K^m$ 被确定后，我们可以随机生成可逆线性仿射 $R : K^n \to K^n$，然后计算公钥 $P = G \circ R : K^n \to K^m$。

假设待签名的消息是 **m**，我们按以下步骤生成其对应的签名：

步骤 1　计算消息 **m** 的哈希值 $\mathbf{y} = hash(\mathbf{m}) \in K^o$。

步骤 2　随机选取醋变量值 $v_1, \cdots, v_v \in K$。

步骤 3　用醋变量值 $v_1, \cdots, v_v$ 替换掉 $x'_1, \cdots, x'_v$ 带入到 $P(x'_1, \cdots, x'_v, \hat{x}_1, \cdots, \hat{x}_o)$，我们能够得到关于油变量 $\hat{x}_1, \cdots, \hat{x}_o$ 的 o 条线性方程。使用高斯消元求解出油变量的值（如果得到的线性系统无解，则跳回步骤 2）。令 $(x_1, \cdots, x_n) =$

$(x'_1, \cdots, x'_v, \hat{x}_1, \cdots, \hat{x}_o)$。

步骤 4 作用 R 的逆映射到 $(x_1, \cdots, x_n)$，得到签名 $(s_1, \cdots, s_n)$

定义 $d = v - o$，当 $d = 0$ 的时候，该方案被称为平衡油醋 (OV) 签名。当 $d > 0$ 时，该方案被称为非平衡油醋 (UOV) 签名。

这里我们给出 UOV 方案的一个一般性描述：

私钥： UOV 方案的私钥由中心映射 G: $K^n \to K^m$ 和线性仿射 $R : K^n \to K^n$ 组成。

公钥: 公钥则为复合映射 $P = G \circ R$: $K^n \to K^m$。

签名生成: 假设要进行签名的消息为 **m**。我们用按如下步骤对消息进行签名：

1) 计算消息摘要 $\mathbf{w} \in K^n$。

2) “求逆”中心映射得到 $\mathbf{x} = G^{-1}(\mathbf{y})$。

3) 计算 R 的逆得到 $\mathbf{s} = R^{-1}(\mathbf{x})$ 作为签名。

签名认证: 签名者发送文档 -签名对 **(m,s)** 给验证者。验证者判断等式 $P(\mathbf{s}) = Hash(\mathbf{m})$ 是否成立。如果成立，则认为签名是正确的；如果不成立，则认为签名是伪造的。

2.3.2 UOV 等价密钥

本节中我们将介绍 UOV 的等价密钥。令 $((G, R), P)$ 为 UOV 的一个公私钥对，$\Omega : K^n \to K^n$ 为一个仿射。我们有：

$$P = G \circ R = \underbrace{G \circ \Omega}_{\text{G'}} \circ \underbrace{\Omega^{-1} \circ R}_{\text{R'}}.$$

令 (G, R) 是一个 UOV 密钥，$\Omega : K^n \to K^n$ 是具有如下形式线性项的仿射

$$\Omega_{lin} = \begin{bmatrix} \Omega^1_{v\times v} & 0_{v\times o} \\ \Omega^3_{o\times v} & \Omega^4_{o\times o} \end{bmatrix}.$$

那么满足 $G' = G \circ \Omega$ 和 $R' = \Omega^{-1} \circ R$ 的 G', R' 是一个等价 UOV 密钥。

对每一个 UOV 公钥系统 P，以极大的概率存在一个 UOV 等价密钥 (G', R') 满足 [47]：

$$R'_{lin} = \begin{bmatrix} 1_{v\times v} & R'_{v\times o} \\ 0_{o\times v} & 0_{o\times o} \end{bmatrix} \tag{2-1}$$

我们可以将 R'_{lin} 写成 $R'_{v+1} \cdot \cdots \cdot R'_n$ 的乘积，其中

$$R_i' = \begin{bmatrix} 1 & & 0 & 0 & r_{1i}' & 0 \\ & \ddots & & \vdots & \vdots & \vdots \\ 0 & & 1 & 0 & r_{vi}' & 0 \\ 0 & \cdots & 0 & 1 & 0 & 0 \\ \vdots & & \vdots & & \vdots & \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}; \quad (i = v+1, \cdots, n).$$

矩阵 R_i' 对应位置上的非零元素和 R_{lin}' 中的相等。

2.3.3　直接攻击

攻击 UOV 方案最直接的攻击方法是尝试直接求解公钥多项式系统 $P(\mathbf{s}) = \mathbf{w}$ 来计算出消息 $\mathbf{m}$（$\mathbf{w} = hash(\mathbf{m})$）的签名 $\mathbf{s}$。由于 UOV 公钥多项式系统是一个欠定系统，所以攻击者可以选择事先随机固定一些变量来创建一个非欠定系统，然后使用基于 Gröbner 的方法进行求解。

2.3.4　UOV 协调攻击

令 $((G, R), P)$ 是一个 UOV 密钥对，并且 R_{lin} 具有方程(2-1)的形式。那么我们可以将 R_{lin}^{-1} 写成

$$R_{lin}^{-1} = R_n^{-1} \cdot \cdots \cdot R_{v+1}^{-1}.$$

由此我们有

$$G_k = R_{v+1}^{-T} \cdot \cdots \cdot R_{n-1}^{-T} \cdot R_n^{-T} \cdot P_k \cdot R_n^{-1} \cdot R_{n-1}^{-1} \cdot \cdots \cdot R_{v+1}^{-1} \quad (k = 1, \cdots, o).$$

定义

$$P_k^i = R_{i+1}^{-T} \cdot \cdots \cdot R_{n-1}^{-T} \cdot R_n^{-T} \cdot P_k \cdot R_n^{-1} \cdot R_{n-1}^{-1} \cdot \cdots \cdot R_{i+1}^{-1} \quad (k = 1, \cdots, o \quad i = v, \cdots, n-1).$$

其中 R_i^{-1} 具有与 R_i 一样的形式，所以矩阵 P_k^i 有如下形式

$$P_k^i = \begin{bmatrix} *_{i\times i} & *_{i\times(n-i)} \\ *_{(n-i)\times i} & 0_{(n-i)\times(n-i)} \end{bmatrix} (k = 1, \cdots, o \quad i = v, \cdots, n-1).$$

UOV 协调攻击 [47] 的目标，就是要从 P_k 开始，对所有的 $(k = 1, \cdots, o)$，按顺序计算出 $P_k^i (i = n-1, \cdot, v)$。当最后我们计算得到 P_k^v 的时候 $(k = 1, \cdots, o)$，我们同时还能得到 $R^{-1} = R_n^{-1} \cdot \cdots \cdot R_{v+1}^{-1}$。实际上这时我们相当于拥有了一个 UOV 等价密钥。

UOV 协调攻击的复杂度主要来源于从 P_k^{i+1} 中计算出 P_k^i $(i = n-1, \cdots, n)$，这个过程中我们需要求解关于 v 个变量的 $o \cdot (n-i)$ 条二次方程。所以整个 UOV 协调攻击的

复杂度主要被求解 v 个变量的 o 条二次多项式所决定。

2.3.5 UOV 攻击

UOV 攻击 [41] 的目的只要找到一个等价仿射 R 使中心映射保持 UOV 结构。UOV 攻击最初是被用来攻击平衡油醋方案 [29]，但也能够对 UOV 产生效果。下面我们将介绍 UOV 攻击的相关细节。

定义 2.8 **油空间:** 我们定义 K^n 上油空间为 $O = \{\mathbf{x} = (x_1, \cdots, x_n) \in K^n : x_1 = ... = x_v = 0\}$.

定义 2.9 **醋空间:** 我们定义 K^n 上醋空间为 $V = \{\mathbf{x} = (x_1, \cdots, x_n) \in K^n : x_{v+1} = ... = x_n = 0\}$.

UOV 攻击的本质就是要找到在油空间在映射 R 下的前像空间。令 E 是一个具有如下形式的线性变换:

$$E = \begin{bmatrix} *_{v\times v} & *_{v\times o} \\ *_{o\times v} & 0_{o\times o} \end{bmatrix}$$

那么我们可以得出 $E(O)$ 是醋空间 V 的一个 o 维子空间。如果 E 是一个可逆线性变换，$E^{-1}(V)$ 是 K^n 的一个 v 维子空间，且 O 是它的子空间。在平衡油醋攻击中，我们有 $E(O) = V$ 和 $E^{-1}(V) = O$，这样我们能够在多项式时间范围内找到一个等价仿射 R。在非平衡油醋攻击中，令 $H = \sum_{i=1}^{o} \lambda_i \cdot \hat{G}_i$ 是 UOV 中心映射多项式相关矩阵的一个线性组合。不难找到一个中心映射多项式相关矩阵 $\hat{G}_k$ 是一个可逆矩阵。那么以不低于 q^{o-v} 的概率，$\hat{G}_k^{-1} \cdot H$ 拥有一个非平凡不变子空间，且该空间是油空间 O 的子空间。对应到公钥多项式相关矩阵上，令 $W = \sum_{i=1}^{o} \hat{P}_i$ 是 UOV 公钥系统多项式相关矩阵的一个线性组合，且 $\hat{P}_k$ 可逆。那么有不低于 q^{o-v} 的概率，$\hat{P}_k^{-1} \cdot W$ 拥有一个非平凡不变子空间，且该空间是 $R^{-1}(O)$ 的子空间。

那么通过不断组合尝试，我们可以找到足够多 $P_k^{-1} \cdot W$ 的子空间。为了测试一个子空间 D 是否是 $R^{-1}(O)$ 的子空间，我们可以随机取 $\mathbf{x}, \mathbf{y} \in D$，然后测试 $\mathbf{x}^T \cdot P_i \cdot \mathbf{y} = 0$ $(i = 1, \cdots, o)$ 是否成立。当进行了足够多的尝试之后，我们就能够找到 $R^{-1}(O)$ 的一组基，这样我们就能够构造出等价仿射 R 并得到 P 的一组等价密钥。UOV 攻击的复杂度大约为 $q^{v-o-1} \cdot o^4$。

2.3.6 UOV 变种方案

虽然 UOV 具有良好的安全性，但是它并没有被广泛使用。一个比较重要的原因是它的密钥太大了，另外一个原因就是它并不是已知最快的多变量签名方案。在文

章 [48] [49] [50] 中，作者们提出了 Cyclic UOV 签名方案。Cyclic UOV 签名方案的核心思想是将一些特殊的序列插入到 UOV 的公钥中来降低存储和提高验证的速度。相比较于普通 UOV，Cyclic UOV 的公钥大小降低了 84%，同时 Cyclic UOV 还具备和 UOV 一样的安全性质。

Cyclic UOV 可以被用来降低公钥大小和提高验证速度。目前也有一些 UOV 变种方案关注于降低 UOV 私钥大小和提高签名速度。TTS[51][52] 和增强的 TTS[53] 就是这样的系统。尽管它们已经被攻破，但是使用稀疏密钥来降低 UOV 私钥大小的方法仍然被大家使用。总的来说，目前大概有三种降低 UOV 私钥大小的方法：伪随机数生成器法、基于矩阵的方法和 NT 法。

相比较于降低 UOV 公钥大小，要降低 UOV 的私钥大小实际上并不算困难。最常见的方法是使用一个伪随机数生成器 (PRNG) 来将 UOV 的公钥压缩成常数大小，这极大了降低了 UOV 签名的私钥大小。但是这种方法会严重影响 UOV 的签名速度。我们需要找到能够同时降低私钥大小和提高签名效率的方法。基于矩阵的方法的核心思想是将 UOV 中心映射多项式矩阵使用对角矩阵形式分成多个小的块 [54]，这样在计算签名的时候步骤 3 中所需要求解的线性系统就会变小很多，从而降低了签名计算的复杂度。NT 法将一些旋转关系引入到 UOV 中心映射的醋变量中，从而降低密钥的大小 [55]。同时因为旋转关系的存在，在进行签名运算的一些中间结果能够被重复利用，这样就能提高签名的效率。

基于矩阵的方法和 NT 法最初都是为 Rainbow 设计的。因为 UOV 可以被视为是一个单层的 Rainbow 签名方案，所以这两种方法都能够直接运用到 Rainbow 上。Tan 等人将这两种方法结合提来，提出了 MB UOV （Matrix based UOV）方案 [56]，能够进一步降低 UOV 私钥大小提高签名效率。但是我们发现 MB UOV 的公钥多变量系统的多项式矩阵 $\hat{P}_i$ 实际上会泄露 $R^{-1}(O)$ 空间的一个 $o-d$ 维子空间，攻击者能够利用这个子空间构造一个 MB UOV 的等价密钥来进行签名伪造。

2.4　Rainbow 签名方案

Rainbow 是由 Ding 等人在 [30] 中提出的一种多变量签名方案。它的核心思想还是基于 UOV 的油醋多项式，但是拥有像彩虹一样的多层结构，故命名为 Rainbow。相比较于 UOV 签名方案，Rainbow 签名方案拥有更短的密钥和签名。本节我们将对 Rainbow 签名方案进行介绍。

2.4.1 基础 Rainbow 方案

Rainbow 签名方案的中心映射是一个多层 UOV 结构的中心映射。令 t 为 Rainbow 中心映射的层数，$v_1,\ldots,v_{t+1}$ 为满足 $0=v_0<v_1<v_2<\cdots<v_{t+1}=n$ 的 $t+1$ 个整数。对于 i=1,...,t，第 i 层 Rainbow 中心映射的醋变量个数为 v_i，油变量个数为 $o_i=v_{i+1}-v_i$。Rainbow 方案总的方程数为 $m=\sum_{i=1}^{t}o_i$，总的变量数为 n。我们称 (v_1,o_1,...,o_t) 为 Rainbow 方案的参数。Rainbow 签名方案的层数 t 可以为任意大于 1 的整数，不过在实际运用中考虑到性能和安全性的折中，通常会取 $t=2$。

Rainbow 的中心映射 G=($g_{v_1+1},\ldots,g_n$) 是一个 K^n 到 K^m 的映射，每个多项式 g_h 具有如下结构：

$$g_h=\mathbf{x}^T A_h\mathbf{x}+\mathbf{b}_h\mathbf{x}+c_h,\mathbf{x}=(x_1,\ldots,x_n)^T.$$

令 $h=v_i+j$ ($i\in[1,\ldots,t],j\in[1,\ldots,o_i]$)。$A_{v_i+j}$ 是一个 n 维方阵，可表示为

$$A_{v_i+j}=\begin{bmatrix} VV_{v_i+j} & VO_{v_i+j} & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},$$

其中 VV_{v_i+j} 是一个随机选取的 v_i 维方阵，VO_{v_i+j} 是一个随机选取的 $v_i\cdot o_i$ 矩阵。$\mathbf{b}_h$ 是有如下形式的一个 n 维向量

$$\mathbf{b}_{v_i+j}=(\mathbf{b}'_{v_i+j},\overbrace{0,\ldots,0}^{n-v_{i+1}}),$$

其中 $\mathbf{b}'_{v_i+j}$ 是一个随机选取的 v_{i+1} 维向量。c_h 是随机选取的常数项。

由于 G 是由多层 UOV 结构组成的，所以我们可以很轻松对它进行“求逆”。对任何向量 **y**=($y_1,\ldots,y_m$)$\in K_m$，我们可以使用算法2-1对其进行“求逆”。

这里我们给出 Rainbow 方案的一个一般性描述：

私钥： Rainbow 方案的私钥由中心映射 G: $K^n\to K^m$ 和两个线性仿射 S: $K^m\to K^m$ 和 $R:K^n\to K^n$ 组成。

公钥: 公钥则为复合映射 $P=S\circ G\circ R$: $K^n\to K^m$。

签名生成: 假设要进行签名的消息为 **m**。我们用按如下步骤对消息进行签名：

1) 计算消息摘要 $\mathbf{w}\in K^n$。

2) 计算 S 的逆 $\mathbf{y}=S^{-1}(\mathbf{w})$。

3) 使用算法2-1“求逆”中心映射得到 $\mathbf{x}=G^{-1}(\mathbf{y})$。

4) 计算 R 的逆得到 $\mathbf{s}=R^{-1}(\mathbf{x})$ 作为签名。

算法 2-1: Rainbow 中心映射“求逆”：$G^{-1}(\mathbf{y})$

输入: $\mathbf{y} = (y_1, \ldots, y_m) \in K^m$.

输出: $\mathbf{x} = (x_1, \ldots, x_n) \in K^n$.

1: 令 $i = 1$，并随机选择初始醋变量 $s_1, \ldots, s_{v_1} \in K$。

2: 令 $(x_1, \ldots, x_{v_1})$=$(s_1, \ldots, s_{v_1})$，并带入到多项式 $g_{v_i+1}, \ldots, g_{v_i+o_i}$ 中得到一组关于 o_i 个变量的线性方程 $L\mathbf{x} = \mathbf{u}$ (如果这组系统无解，则回到步骤 1)。

3: 使用高斯消元对线性系统进行求解得到 $(x_{v_i+1}, \ldots, x_{v_i+o_i})$=$(s_{v_i+1}, \ldots, s_{v_i+o_i})$。

4: 令 $i = i + 1$。如果 $i < t + 1$，回到步骤 2。

5: 返回 $(x_1, \ldots, x_n)$。

签名认证: 签名者发送文档 -签名对 **(m**,**s)** 给验证者。验证者判断等式 $P(\mathbf{s}) = Hash(\mathbf{m})$ 是否成立. 如果成立，则认为签名是正确的；如果不成立，则认为签名是伪造的。

2.4.2　Rainbow 等价密钥

令 S, G, R 为一个双层 Rainbow 的密钥。我们定义两个线性仿射 $\Omega : K^n \to K^n$ 和 $\Sigma : K^m \to K^m$ 具有如下形式：

$$\Sigma_{lin} = \begin{bmatrix} \Sigma^1_{o_1 \cdot o_1} & 0o_1 \cdot o_2 \\ \Sigma^3_{o_2 \cdot o_1} & \Sigma^4_{o_2 \cdot o_2} \end{bmatrix}, \quad \Omega_{lin} = \begin{bmatrix} \Omega^1_{v_1 \cdot v_1} & 0_{v_1 \cdot o_1} & 0_{v_1 \cdot o_2} \\ \Omega^4_{o_1 \cdot v_1} & \Omega^5_{o_1 \cdot o_1} & 0_{o_1 \cdot o_2} \\ \Omega^7_{o_2 \cdot v_1} & \Omega^8_{o_2 \cdot o_1} & \Omega^9_{o_2 \cdot o_2} \end{bmatrix}$$

计算 $S' = S \circ \Sigma^{-1}$，$G' = \Sigma \circ G \circ \Omega$，$R' = \Omega^{-1} \circ R$，则 (S', G', R') 是该双层 Rainbow 方案的一个等价密钥。值得注意的是，Σ^{-1}_{lin} 和 Ω^{-1}_{lin} 与 Σ_{lin} 和 Ω_{lin} 具有相同的形式。

令 P 是一个双层 Rainbow 方案的公钥。那么以极大的概率存在一个对应的 Rainbow 密钥 S, G, R 满足 $S \circ G \circ R = P$ 且

$$S_{lin} = \begin{bmatrix} 1_{o_1 \cdot o_1} & S_{o_1 \cdot o_2} \\ 0_{o_2 \cdot o_1} & 1_{o_2 \cdot o_2} \end{bmatrix}, \quad R_{lin} = \begin{bmatrix} 1_{v_1 \cdot v_1} & R_{v_1 \cdot o_1} & R_{v_1 \cdot o_2} \\ 0_{o_1 \cdot v_1} & 1_{o_1 \cdot o_1} & R_{o_1 \cdot o_2} \\ 0_{o_2 \cdot v_1} & 0_{o_2 \cdot o_1} & 1_{o_2 \cdot o_2} \end{bmatrix}. \tag{2-2}$$

2.4.3　直接攻击

攻击 Rainbow 方案最直接的攻击方法是尝试直接求解公钥多项式系统 $P(\mathbf{s}) = \mathbf{w}$ 来计算出消息 **m**（$\mathbf{w} = hash(\mathbf{m})$）的签名 **s**。由于 Rainbow 公钥多项式系统是一个欠定系

统，所以攻击者可以选择事先随机固定一些变量来创建一个非欠定系统，然后使用基于 Gröbner 的方法进行求解。

2.4.4 低秩攻击

小秩攻击的目的是要找到一个具有很低的秩的公钥多项式相关矩阵 $\hat{P}_k$ $(v_1 \leq k \leq n)$ 的线性组合矩阵（在 Rainbow 中，这个秩的大小是 v_2）。这样的矩阵实际上对应的是 Rainbow 中心映射第一层多项式相关矩阵的一个线性组合。要找到这样的矩阵，实际上我们是要解决一个最小秩问题。

定义 2.9 **最小秩问题：** 给定 m 个 $n \cdot n$ 矩阵 $\hat{P}_{v_1+1}, \cdots, \hat{P}_n$，要求找到一个线性组合 $H = \sum_{i=v_1+1}^{n} \lambda_i \hat{P}_i$ 满足 $Rank(H) \leq v_2$。

我们可以用算法2-2来解决该问题。

算法 2-2: 针对 Rainbow 的最小秩攻击

输入： 矩阵 $\hat{P}_{v_1+1}, ..., \hat{P}_n$。

输出： 线性组合 $C = \sum_{i=v_1+1}^{n} c_i \cdot \hat{P}_i$ 满足 $Rank(C) \leq v2$。

1: 随机选择向量 $\lambda \in K^m$ 并计算 $H = \sum_{i=v_1+1}^{n} \lambda_i \cdot \hat{P}_i$。

2: 如果 $Rank(H) > 1$ 且 $Rank(H) < n$，则随机选择向量 $\gamma \in Ker(H)$ 并计算 $C = \sum_{i=v_1+1}^{n} \gamma_i \cdot \hat{P}_i$。

3: 如果 $Rank(C) > v_2$，回到步骤 1。

4: 返回 C。

当攻击者成功将第一层 Rainbow 中心映射分离之后，分离剩下的中心映射就变得容易了。整个最小秩攻击的复杂度主要在于分离第一层 Rainbow 中心映射，其复杂度公式为 $o_1 \cdot q^{v_1+1}$。

2.4.5 高秩攻击

在介绍高秩攻击之前，我们首先定义 Rainbow 中各层的油空间。令

$$O_i = \{x \in K^n : x_1 = ... = x_{v_i} = 0\}(i = 1, ..., u).$$

我们称 O_i 为 Rainbow 中心映射第 i 层油空间。在 Rainbow 中心映射中，变量 $x_{v_t+1}, ..., x_n$ 只在最后一层二次多项式 $(g_{v_t+1}, ..., g_n)$ 的二次项中存在。由此我们可知 $O_t \subset ker(\sum_{k=v_1+1}^{v_t} a_k \cdot \hat{P}_k)$，其中 a_k 为基域上任意元素。这意味着 $R^{-1}(O_t)$ 在 $\hat{P}_k$ $(k = 1, \cdots, n)$ 的一些线性组合的核空间之中。我们可以通过随机生成足够多的线性组

合，来找到 $R^{-1}(O_t)$ 的一组基。算法2-3给出了求解 $R^{-1}(O_t)$ 的具体步骤。

算法 2-3: 针对 Rainbow 的高秩攻击

输入: 矩阵 $\hat{P}_{v_1+1}, \cdots, \hat{P}_n$。

输出: $R^{-1}(O_t)$。

1: 随机选择向量 $\lambda \in K^m$ 并计算 $H = \sum_{i=v_1+1}^{n} \lambda_i \cdot \hat{P}_i$ 和 $V = ker(H)$。

2: 如果 $dim(V) > 1$，令 $(\sum_{k=v_1+1}^{n} \lambda_k \hat{P}_k)V = 0$ 并求解。看解空间的维度是否为 $m - o_t$。

3: 每次以 $\frac{1}{q^{o_t}}$ 的概率，我们能找到 $V \subset R^{-1}(O_t)$。持续这个过程直到我们找到整个 $R^{-1}(O_t)$ 空间。

4: 返回 $R^{-1}(O_t)$。

当攻击者获得 $R^{-1}(O_t)$ 之后，通过对 $R^{-1}(O_t)$ 子空间的研究。它可以很容易找到各层 $R^{-1}(O_i)$。所以我们可以得到高秩攻击的复杂度为 $q^{o_t} \cdot \frac{n^3}{6}$。

2.4.6　彩虹带分离攻击

彩虹带分离攻击 [57] 可以看成是 UOV 协调攻击的扩展方案。假设 (S, G, R) 为 Rainbow 一个等价密钥，且 S_{lin} 和 R_{lin} 具有等式(2-2)中的形式。我们可以将 R_{lin} 写成 $R_n \cdot ... \cdot R_{v_1+1}$，则有

$$G_k = \sum_{l=1}^{m} S_{kl}^{-1}(R_{v_1+1}^{-T} \cdot ... \cdot R_n^{-T} \cdot P_l \cdot R_n^{-1} \cdot ... \cdot R_{v_1+1}^{-1}),$$

其中 S_{kl} $(k, l = 1, ..., m)$ 表示矩阵 S_{lin}^{-1} 对应位置上的元素。

和 UOV 协调攻击类似的，彩虹带分离攻击实际上也是找到一系列矩阵 $P_k^{n-1}, ..., P_k^{v_1}$ 最后使 $P_k^{v_1}$ 具有 G_k 的格式。这样攻击者拥有 $P_k^{v_1}$ $(k = v_1 + 1, ..., n)$ 就能够像合法用户一样生成签名。

令矩阵 S'_n 只有前两个块的最后一列存在任意元素，其值等于 S' 中对应位置的值。对应的，矩阵 R'_n 中只有第二块的第 $o - 1$ 行中存在任意元素，其值等于 R' 中对应位置的值。当普通 Rainbow 中心映射 P 作用上好密钥 S'_n 和 R'_n 之后，$(S_n'^{-1} \circ P) \circ R_n'^{-1}$ 将会具有如图 4-3的形式。

令 S'_n 和 R'_n 中的未知元素为未知数，从图 4-3中我们可以得到 n 个变量的一条三次多项式方程，$m + n - 2$ 条二次多项式方程。我们可以使用求解 Gröbner 基的方法来求解这个方程组。其直接攻击复杂度可以通过计算该方程组的正则度 d_{reg} [58] 来估量。其

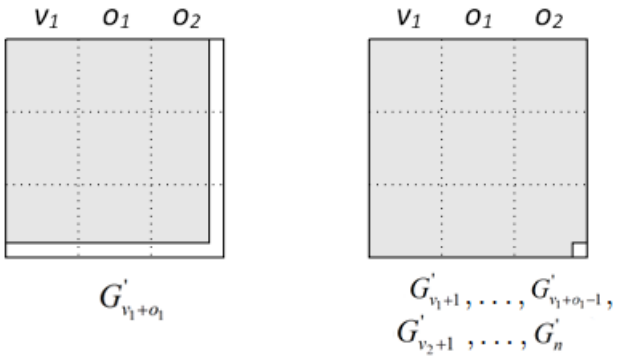

图 2-2 普通 Rainbow 作用了好密钥之后的中心映射

正则度为希尔伯特序列

$$S_{m,n} = \frac{\prod_{i=1}^{m}(1-z^{d_i})}{(1-z)^n},$$

的第一个非负项的索引。使用 F5 算法来求解这个系统的复杂度为

$$O(\binom{n+d_{reg}}{d_{reg}}^{\omega}),$$

其中 n 是变量数，m 是方程数量，ω 是线性代数常量，在密码分析中我们通常取 $\omega = 2$。

2.4.7 UOV 攻击和 UOV 协调攻击

Rainbow 可以看成是一个多层 UOV 结构的多变量签名方案。令 $P = S \circ G \circ R$ 为一个 Rainbow 签名方案的公钥，定义 $\hat{G} = S \circ G$。我们可以看出 $\hat{G}$ 实际上可以看成一个单层 UOV 中心映射，其具有 v_t 个醋变量和 o_t 个变量。则 Rainbow 公钥可以写成 $P = \hat{G} \circ R$，这明显是一个 UOV 公钥形式。所以我们可以得知，所有针对 UOV 的攻击实际上也是能用到 Rainbow 上面的。由前面我们对 UOV 攻击的分析可知，对 Rainbow 的 UOV 攻击复杂度为 $q^{v_t-o_t-1} \cdot o_t^4$。而对 Rainbow 的 UOV 协调攻击的复杂度主要被求解关于 v_t 个变量的 o_t 条二次多项式的复杂度所决定。

2.5 SRP 加密方案

SRP 加密方案本质上是 Square 加密方案和 Rainbow 签名方案和加方法 (Plus method) 的一个组合性方案。通过将 Square 加密方案和 Rainbow 签名方案合并到一起，一些针对单个方案的攻击方法不再奏效。因为 Square 加密方案和 Rainbow 签名方案效率都很高，所以其组合方案也具有很高的效率。

2.5.1 基础 SRP 加密方案

这里我们先介绍 SRP 方案的一般构造。为了简单起见，我们用 UOV 来代替 Rainbow。令 K 为一个奇特征域且阶 q 满足 $q \equiv 3 \bmod 4$。令 d 为一个奇数，扩域 E=K^d。

令 $\phi : E \to K^d$ 是一个 E 与向量空间 K^d 的同构。令 o，r，s，l 为非负整数。SRP 方案的一般性描述如下。

密钥生成： 令 $n = d + o - l$，$n' = d + o$ 且 $m = d + o + r + s$，中心映射 G 是三个影射 G_S，G_R 和 G_P 的组合。公钥则为 $P = S \circ G \circ R$，其中 $S : K^m \to K^m$ 和 $R : K^n \to K^{n'}$ 是随机满秩线性仿射，G_S，G_R 和 G_P 构造如下：

1) Square 部分 $G_S : K^{n'} \to K^d$ 的构造：

$$K^{d+o} \overset{\pi_d}{\to} K^d \overset{\phi^{-1}}{\to} E \overset{X \to X^2}{\to} E \overset{\phi}{\to} K^d.$$

其中 $\pi_d : K^{d+o} \to K^d$ 为取前 d 个坐标值的投影映射。

2) UOV（Rainbow）部分 $G_R = (g_1, ..., g_{o+r}) : K^{n'} \to K^{o+r}$ 是一个普通 UOV(Rainbow) 方案的中心映射。

3) 加映射部分 $G_P = (p_1, ..., p_s) : K^{n'} \to K^s$ 是由 s 个随机多项式 $p_1, ..., p_s$ 组成的。

4) 随机生成满秩仿射 $R : K^n \to K^{n'}$ 和可逆仿射 $S : K^{n'} \to K^{n'}$，计算公钥映射 $P = S \circ (G_S || G_R || G_P) \circ R$。

加密： 给定消息 $\mathbf{m} \in K^n$，计算密文 $\mathbf{c} = P(\mathbf{m})$。

解密： 给定密文 $\mathbf{c} = (c_1, ..., c_m)$ 解密算法如下：

1) 求逆 S 仿射: $\mathbf{x} = (x_1, ..., x_m) = S^{-1}(\mathbf{c})$。

2) 计算 $X = \phi^{-1}(x_1, ..., x_d)$。

3) 计算 $Z_{1,2} = \pm X^{\frac{q^d+1}{4}}$，令 $\mathbf{v^i} = (v_1^i, ..., v_d^i) = \phi(Z_i)(i = 1, 2)$。

4) 给定醋变量 $v_1^i, ..., v_d^i (i = 1, 2)$，求解 $n' - d = o$ 个变量的 $o + r$ 个线性方程：$g_k(y_1^i, ..., y_d^i, u_{d+1}, ..., u_{n'}) = x_{d+k}\ (k = 1, ..., o + r; i = 1, 2)$。解表示为 $y_{d+1}, ..., y_{n'}$。

5) 计算消息 $\mathbf{m} = R^{-1}(\mathbf{y})$。

2.5.2　解密失败概率

为了保证解密算法能够得到正确的明文。我们需要保证公钥系统是一个单射系统。在 ABC[31] 加密方案中，解密错误的概率是不可忽略的，原因就是公钥系统并不是一个单射系统。但优化后的 ABC 加密方案变成了一个几乎单射系统 [59]。因此，ABC 加密方案的解密错误率可以通过选择合适的参数降到极小。SRP 方案具有类似的调优办法，其主要思路是通过在中心映射中添加冗余的油醋多项式，从而提高 Rainbow 部分线性系统能够确定唯一解的概率。假设中心映射中油 r 条冗余油醋多项式，则 SRP 公钥系统不是单射的概率是低于 q^{-r}，所以 SRP 中解密错误率低于 q^{-r}。

2.5.3 直接攻击

SRP 加密方案的公钥系统 $P : K^n \to K^{n+l+s}$ 是一个超定系统。且变量数 n 与多项式数目 $n+l+s$ 大致相近。此时要对 SRP 公钥系统 $P(\mathbf{m}) = \mathbf{c}$ 进行直接攻击的最佳方法是使用 F4/F5[60] 等计算 Gröbner 基进行直接求解。但总的来说 SRP 加密方案直接攻击的复杂度为指数级别。

2.5.4 差分攻击

给定公钥系统 P，差分方程的定义如下：

$$DP(\mathbf{a}, \mathbf{x}) = P(\mathbf{x} + \mathbf{a}) - P(\mathbf{x}) - P(\mathbf{a}) + P(\mathbf{0}).$$

当我们用 K^n 上某个点替代 $\mathbf{a}$，$DP(\mathbf{a}, \mathbf{x})$ 将变成一个从 K^n 到 K^m 的线性仿射 M。文章 [61] 给出了针对简单 Square 方案的差分攻击方法。。

在 SRP 方案中，Plus 部分是对差分攻击免疫的，它并不存在明显的差分不变性。针对 Rainbow 部分的 UOV 攻击实质上是差分攻击的一种，因为其本质上是想找到公钥多项式相关矩阵 $\hat{P}_i$ 的线性组合的一个小的不变子空间。但一次搜索找到这样空间的概率小于 $q^{v_t - o_t}$。在加上加方法的扰动，攻击者无法从公钥中找到关于 Rainbow 部分的差分性质。

Square 部分实际上是很容易受到差分攻击的。但 Square 部分的差分性质会被 Rainbow 多项式和 Plus 部分的随机多项式完全打乱。攻击者想要对 Square 部分进行差分必须先将其从公钥系统中分离出来，这复杂度不低于最小秩攻击。故差分攻击并不能影响 SRP 方案的安全性。

2.5.5 秩攻击

Thommae 等人针对 Square 及其变种方案的最小秩攻击 [62]。实际上，针对 Square 变种方案的最小秩攻击可以被分为两类。一类是小域上的最小秩攻击，这与前面我们提高的针对 Rainbow 和 UOV 的最小秩攻击本质上是一样的。另一类是针对大域上系统的最小秩攻击，这种攻击的一个典型就是针对 HFE 加密方案的 KS 攻击 [21]。

在 SRP 加密方案中，中心映射结构实际上是一个双层结构。带入到 Square 映射中的变量只是中心映射 $d+o+s$ 个变量中的前 d 个变量，此时 KS 攻击很难对该系统其作用。但即使是双层变量结构，SRP 方案也容易被针对中心映射多项式小域上相关矩阵的小秩攻击所威胁。其攻击方法类似于 Thommae 等人针对双层 Square 方案所提出的小秩攻击，复杂度大约为 $q^{l+1}d(n+l+s)^3$

2.5.6　线性方程攻击

在 SRP 中，Square 部分的中心映射 $X \to X^2$ 实际上是 C* 中心映射 $X \to X^{q^\theta+1}$ 的一个特殊实例 ($\theta = 0$)。但针对 C* 的线性方程攻击实际上无法对 Square 方案进行攻击。因为 C* 中的线性关系 $XY^{q^\theta} = YX^{q^{2\theta}} = 0$ 在 Square 系统中变成了平凡关系 $XY = YX$。所以 Square 的明密文间不存在简单的双线性关系。

SRP 中 Rainbow 和 Plus 部分的明密文之间显然也并不存在双线性关系，故 SRP 方案对线性方程攻击免疫。

2.6　小结

本章中，我们主要介绍了多变量公钥密码的一些基础知识。包括多变量公钥密码所基于的困难问题和多变量公钥密码的基本构造方法。同时本章介绍了 UOV、Rainbow 和 SRP 三个多变量公钥加密和签名方案的基本构造和基本安全性分析。方便读者理解本文后面章节的内容。

第三章　一种基于循环矩阵的 UOV 签名方案

UOV 是目前最重要的 MPKC 签名方案之一。它具有很强的安全性，并且被认为是抗量子计算机攻击的。但是 UOV 签名方案并没有被广泛的使用。一个原因是它的密钥比较大，另外一个原因是它并不是目前已知的最快的多变量签名方案。很多多变量签名方案如 Gui[24] 和 Rainbow 都在速度上优于 UOV 签名方案。为了让 UOV 签名方案更加实用，我们必须降低 UOV 的密钥大小并提高其签名效率。本章中，我们提出一个新的 UOV 变种方案：基于循环矩阵的 UOV 签名方案。我们分析并用实验证实了循环 UOV 的性能和安全性。相对于目前所有的 UOV 变种，它拥有更短的私钥和更高的签名效率。

3.1　基本思路

我们提出的基于循环矩阵的 UOV 签名方案名字上很像文章 [50] 中的 Cyclic UOV, 但是他们的基础思想差别很大。在 Cyclic UOV 中，作者主要目的是降低 UOV 的公钥大小和签名验证复杂度。而在我们的循环 UOV 中，我们主要关注于降低私钥大小和签名生成复杂度。在本文中，循环 UOV(Circulant UOV) 指的是本文中的 UOV 变种方案，而 Cyclic UOV 指的是文章 [50] 的方案。这里我们首先介绍循环 UOV 的基本思想。

循环 UOV 的基本思想，是想要利用循环矩阵的性质加速第2.3.1节中 UOV 签名过程中的步骤 3。步骤 3 是 UOV 签名生成过程中最慢的部分。在步骤 3 中，我们需要求解一个线性方程组$L\mathbf{x} = \mathbf{u}$，

其中 L 是一个 $o \cdot o$ 矩阵，$\mathbf{u}$ 是一个 o 维向量，$\mathbf{x}$ 是一个 o 维未知数向量。一般情况下，我们使用高斯消元计算出 $\mathbf{x}$ 的值。在循环 UOV 中，我们在 UOV 的中心映射中加入一些旋转关系，这会让矩阵 L 变成一个循环矩阵。循环矩阵的逆矩阵可以被很快计算出来。这可以大大加速步骤 3 的签名速度。同时，我们在中心映射中引入的旋转关系，也可以降低 UOV 密钥大小。

3.2　循环 UOV 的中心映射

本节我们将介绍循环 UOV 的具体中心映射构造和其相关性质。

3.2.1　循环 UOV 的中心映射结构

首先，我们给出循环 UOV 的中心映射多项式的矩阵表示。我们保留中心映射多项式的常数和线性部分，所以每个中心映射多项式可以用一个 $(n+1) \cdot (n+1)$ 的矩阵来

表示。如图 3-1.

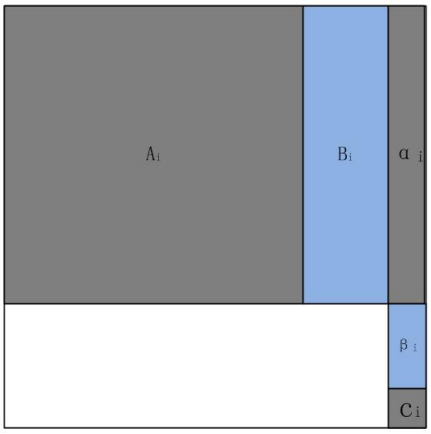

图 3-1 循环 UOV 中心映射多项式的矩阵表示

白色表示该部分为零元素，灰色表示该部分可以取基域上的任意元素，蓝色表示该部分和其他中心映射多项式的矩阵具有旋转关系。矩阵A_i 是一个 $v \cdot v$ 大小的方阵，用来表示醋醋变量系数。矩阵B_i 是一个 $v \cdot o$ 大小的矩阵，用来表示油醋变量交叉项系数。最后一列中的β_i 表示油变量的线性系数，α_i 表示醋变量的线性系数，c_i 表示常数项。循环 UOV 的每个中心映射多项式的矩阵表示看上去都和普通 UOV 的一样。

实际上，循环 UOV 的中心映射中并不存在循环矩阵。而是在中心映射多项式相关矩阵的子矩阵中，存在一些旋转关系。这些旋转关系，能够帮我们在签名过程中得到一个循环矩阵。

3.2.2 循环 UOV 的中心映射大小

基于前面的描述，我们这里给出构造循环 UOV 中心映射的具体参数，并计算其大小。

1) B_1: 初始的 $v \cdot o$ 矩阵，用来表示醋变量和油变量的交叉项系数。

2) β_1: 初始的 o 维向量，用来表示油变量的线性项系数。

3) A_k，$k \in [1,\cdots,o]$: o 个随机的 $v \cdot v$ 矩阵，用来表示醋变量的二次系数。

4) α_k，$k \in [1,\cdots,o]$: o 个 v 维向量，用来表示油变量的线性系数。

5) c_k，$k \in [1,\cdots,o]$: o 个中心映射多项式的常数项。

从上面可以看出，循环 UOV 的中心映射大概需要用

$$o \cdot (\frac{v \cdot v + 1}{2} + v + 1) + o \cdot v + o$$

个基域元素表示。而普通 UOV 的中心映射大概需要用

$$o \cdot (\frac{v \cdot (v+1)}{2} + o \cdot v + n + 1)$$

个基域元素来表示。对于实际参数而言，循环 UOV 的中心映射大小大概比普通 UOV 小 45%.

3.3 循环 UOV 中心映射“求逆”

在节中，我们主要介绍如何快速地“求逆”循环 UOV 的中心映射。

假设我们要“求逆”的消息是 **y**. 我们随机的选取醋变量向量 **v**. 用 $(v_1, \cdots, v_v)$ 替换醋变量 $(x_1, \cdots, x_v)$，我们会得到一个关于 o 个变量的线性方程组。对每个中心映射多项式 P_k，我们能够得到方程:

$$\underbrace{\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k}_{\text{常数}} + \underbrace{\mathbf{v}^T * B_k * \mathbf{o} + \beta_{\mathbf{k}} \cdot \mathbf{o}}_{\text{线性项}} = y_k.$$

其中**o**=$(o_1, \cdots, o_o)$ 表示油变量向量$(x_{v+1}, \cdots, x_{v+o})$。令$u_k = y_k - (\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k)$, 那么我们能够得到一个线性系统:

$$\underbrace{\begin{pmatrix} \mathbf{v}^T * B_1 + \beta_1 \\ \mathbf{v}^T * B_2 + \beta_2 \\ \vdots \\ \mathbf{v}^T * B_{o-1} + \beta_{o-1} \\ \mathbf{v}^T * B_o + \beta_o \end{pmatrix}}_{\mathrm{L}} \begin{pmatrix} o_1 \\ o_2 \\ \vdots \\ o_{o-1} \\ o_o \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{o-1} \\ u_o \end{pmatrix}.$$

矩阵B_i 和向量β_i 之间拥有旋转关系。当我们将 **v** 带入后，矩阵 L 会变成一个 $o \cdot o$ 大小的循环矩阵。这样我们就可以很快的求逆矩阵 L 了。

3.3.1 计算矩阵 L

在具体介绍如何求逆矩阵 L 之前，我们首先讨论如何高效的计算出矩阵 L。一般情况下，要计算出矩阵 L，我们需要计算出 $\mathbf{v}^T * B_k + \beta_k$ （$k \in [1, \cdots, o]$）。不过在循环 UOV 中，因为 B_k 实际是 B_1 旋转 k 次得到的，β_k 实际是 β_1 旋转 k 次得到的。所以其实我们只需要计算出矩阵 L 的第一行。剩下的部分直接可以通过旋转得到。相比于普通的 UOV，循环 UOV 中计算 L 的时间复杂度降低了 o 倍。

3.3.2 L 的可逆概率

在一次签名计算中，如果我们计算得到的矩阵 L 是不可逆的。那么我们就必须重新选择一个醋向量 **v** 来计算出一个可逆的矩阵 L。为了能够得到一个更快的签名方案，我们必须保证矩阵 L 可逆的概率非常高。这要求我们选择合适的参数，使基域上的随机循环矩阵的可逆概率很高。对于不同的基域和参数 o，这里我们采用实验的方法随机

生成 10^5 个随机循环矩阵，然后计算他们的平均可逆概率。

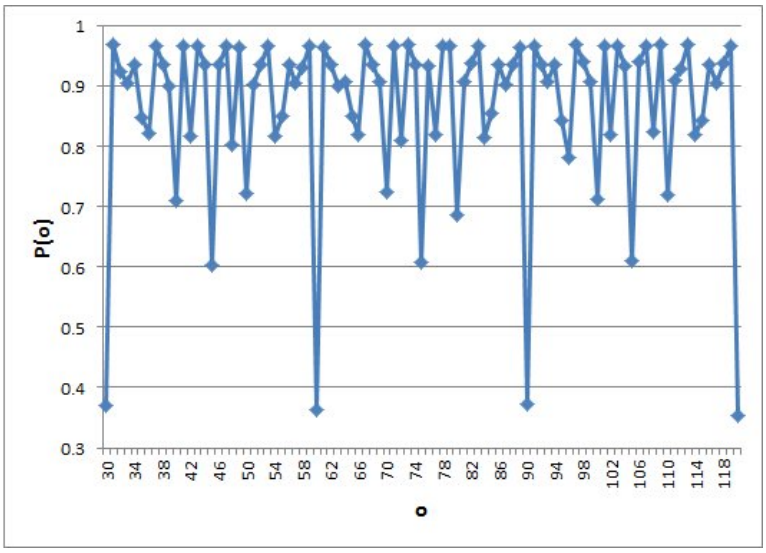

图 3-2 GF(31) 上随机循环矩阵的可逆概率

从实验结果来看，以 GF(31) 作为循环 UOV 的基域看上去是不错的选择。当 GF(31) 为循环 UOV 基域时，$P(o)$ 看上去是一个周期函数。对于大多数 o，$P(o)$ 的值接近 $\frac{30}{31}$。这是 GF(31) 上一个随机矩阵的可逆概率。不过需要注意的是，如果 o 是 5 或 6 的倍数的时候，$P(o)$ 的值会显著的降低。所以在参数选择的时候，我们要避免选择这样的参数。这里我们是对 GF(31) 域进行了实验分析。为了更好的选择参数，本章后面将提出循环矩阵可逆概率的相关公式。

3.3.3 求逆矩阵 L

我们使用扩展欧几里得算法计算循环矩阵 L 的逆矩阵 [63]。这里我们给出其简单的描述。假设矩阵 L 是一个有限域 K 上的可逆循环矩阵。我们希望计算出满足 $LJ = I$ 的循环矩阵 J（循环矩阵的逆矩阵也是一个循环矩阵）。令 $(l_0, l_1, \cdots, l_{o-1})$ 是矩阵 L 的第一行。我们可以定义循环矩阵 L 在多项式环 $K[x]$ 上的相关多项式 $f(x) = \sum_{i=0}^{o-1} l_i x^i$。计算循环矩阵 L 的逆等价于要计算 $F_q[x]$ 上找到满足 $f(x) * g(x) = 1 \mod x^o - 1$ 的多项式 $g(x)$。使用扩展欧几里得算法计算这个问题大概需要 $O(o^2)$ 个基域上的代数运算。这比高斯消元快大约 o 倍。所以，求逆循环 UOV 的矩阵 L 比普通 UOV 快很多。表 3-1给出了一个相关对比。

3.4 循环 UOV 的一般性描述

本节我们给出循环 UOV 签名方案的一般性描述。相对于普通 UOV 签名方案，我们在循环 UOV 签名中心映射左边加上了可逆仿射 S 来增强方案安全性。仿射运算可以

表 3-1 “求逆”循环 UOV 和普通 UOV 中心映射的相关对比

	循环 UOV	UOV
计算 L	$O(o^2)$	$O(o^3)$
计算 L^{-1}	$O(o^2)$	$O(o^3)$

被高效地被计算。尤其是当我们使用 SIMD 相关指令 [64] 时，仿射的运算开销会大大降低。所以仿射 S 基本上不会影响签名生成的速度。

密钥生成: 根据所需的安全级别，选取包括基域 $K = GF(q)$, 油变量数 o 和醋变量数 v 等合适的参数。令 $n = o + v$，按以下步骤生成密钥:

1) 随机生成 (B_1,β_1,R,S) 和 (A_k,α_k,c_k $k \in [1,\cdots,o]$).

2) 用 (B_1,β_1) 旋转生成 (B_k,β_k) ($k \in [2,\cdots,o]$) 来构造中心映射 G.

3) 计算复合映射 P=$S \circ G \circ R$: $K^n \to K^o$ 作为公钥.

4) 存储 (B_1,β_1,R^{-1}, S^{-1}) 和 (A_k,α_k,c_k)，$k \in [1,\cdots,o]$ 作为私钥.

签名生成: 假设待签名的文档是 **m**，我们按一下步骤生成其对应的签名：

1) 计算消息 **m** 的哈希值 $\mathbf{y} \in K^o$。

2) 作用逆仿射 S^{-1} 到 **y** 上，得到 $\hat{\mathbf{y}} = S^{-1}(\mathbf{y})$.

3) 使用第3.3节中提到的方法“求逆”中心映射: $\hat{\mathbf{x}} = G^{-1}(\hat{\mathbf{y}})$.

4) 作用逆仿射 R^{-1}，得到$\mathbf{s} = R^{-1}(\hat{\mathbf{x}})$. 输出签名 $\mathbf{s} \in K^n$.

签名验证: 签名者将消息签名对 $(\mathbf{m}, \mathbf{s})$ 发送给验证者。验证者检查 $P(\mathbf{s})$ 是否等于 $Hash(\mathbf{m})$ 来验证签名正确性。如果相等，则签名合法。否则，签名非法。

3.5 循环 UOV 安全性分析

在本节中，我们将已知的针对 MPKC 签名方案的攻击方法运用上循环 UOV 上来分析循环 UOV 的安全性。

在循环 UOV 中，公钥 P=$S \circ G \circ R$ 是一个复合映射。因为中心映射 G 中所有的多项式全部都是油醋多项式，所以 $S \circ G$ 也将会是油醋多项式形式，但 $S \circ G$ 中的多项式并不会有中心映射多项式的旋转结构。这意味着循环 UOV 实际上是 UOV 的一个子集，所有针对 UOV 的攻击方法都可以运用到循环 UOV 上来。不过本节我们将证明，当参数选择合适时，循环 UOV 有着普通 UOV 一样的安全性质。

3.5.1　直接攻击

攻击 UOV 签名方案最直接的方法就是直接求解多变量二次方程组 $P(\mathbf{x})=\mathbf{y}$。UOV 的公钥 P 是一个欠定二次多项式方程组。求解欠定多项式非线性方程组的时候，一个比较好的策略是通过先随机猜测一些变量的值来构造一个超定系统，再使用求解 Gröbner 基 [65] 的代数方法来求解超定系统。这种方法被称为混合求解法。这里我使用混合求解法来对 UOV 和循环 UOV 公钥系统进行直接求解，其中我们选择 MAGMA[66] 上的 F4 算法 [67] 来进行 Gröbner 基的求解。对于每组被测参数，我随机生成 100 个循环 UOV 和 UOV 的实例，然后计算混合求解法的平均耗时。表 3-2给出了在基域 GF(5) 和 GF(2^2) 对循环 UOV 和 UOV 进行直接攻击的时间。

表 3-2 循环 UOV 和 UOV 的直接攻击对比

参数 (K,n,m)	UOV	循环 UOV
(GF(2^2),9,3)	0.064s	0.065s
(GF(2^2),12,4)	1.999s	1.961s
(GF(2^2),15,5)	77.334s	75.315s
(GF(5),9,3)	0.289s	0.311s
(GF(5),12,4)	10.812s	10.804s
(GF(5),15,5)	547.251s	545.741s

从表 3-2中，我们可以看出。使用混合求解法直接攻击循环 UOV 和 UOV，攻击时间非常接近。我们可以认为循环 UOV 和 UOV 几乎具有相同的抗直接攻击性质。所以我们可以认为循环 UOV 和普通 UOV 的公钥系统具有几乎相同的正则度。循环 UOV 的正则度 d_{reg}[68] 等于满足多项式 $\frac{(1-z^2)^m}{(1-z)^n}$ 的 z^D 项的系数小于或等于 0 最小的次数 D 的值。循环 UOV 的直接攻击复杂度公式为：

$$\min_{k\geq 0} q^k \cdot O(m \cdot \binom{m-k+d_{reg}+1}{d_{reg}}^{\omega}).$$

3.5.2　UOV 协调攻击

在 UOV 签名方案中，中心映射多项式相关矩阵 $\hat{G}_i$ 的右下角的油油项系数一定是 0。UOV 协调攻击就是利用这个性质来生成一些二次方程。攻击者通过求解关于 v 个变量的 $(n-j)\cdot o\,(j=n-1,...,v)$ 个二次方程来等到一个等价密钥。如果我们将 v 和 o 选的足够大，UOV 协调攻击的复杂度会大于直接攻击的复杂度。这样 UOV 协调攻击就

不会影响到方案的安全性。因为循环 UOV 也拥有 UOV 的密钥结构，所以 UOV 协调攻击也可以被用来攻击循环 UOV。

有人可能认为循环 UOV 中的旋转关系能够会帮攻击者在 UOV 协调攻击中获得更多的方程。比如，我们有 $B_1[1,1]-B_2[1,2]=0$。但是这是在没有考虑 S 的影响的情况下才成立的。当 S 作用到中心映射 G 上后，$S\circ G$ 已经不存在 G 中的旋转关系了。攻击者想利用 G 中的旋转关系，它就必须引入了更多变量的解决三次方程。攻击者实施这样的攻击需要求解关于 v^2+o-v 个变量的 $v\cdot o+o-1$ 个三次方程和 o 个二次方程。当 o 和 v 足够大的时候，这样的攻击复杂度远大于直接攻击。

3.5.3 彩虹带分离攻击

彩虹带分离攻击原本是用来攻击 Rainbow 签名方案的。它的核心思想是想要利用 Rainbow 的稀疏密钥结构来生成等价密钥。攻击者想要找到一个满足 $P=S'\circ G'\circ R'$ 的等价密钥 (S',G',R')，同时 G' 还必须拥有常规 Rainbow 中心映射的形式。在循环 UOV 中，我们有用稠密的油醋交叉项系数。每个中心映射多项式相关矩阵 $\hat{G}'_i$ 看上去都是完全随机的。攻击者无法确定其中的 0 元素来进行彩虹带分离攻击。

有人可能认为旋转关系也许能够帮助攻击者在这个攻击中获得更多的方程。但这并没有考虑到在彩虹带分离攻击中 S' 和 R' 具有特殊的形式，在这种情况下 G' 就已经失去了旋转关系了。所以彩虹带分离攻击是无法作用到循环 UOV 上的。

3.5.4 UOV 攻击

UOV 攻击的目的要通过 UOV 公钥系统的差分对称性来找到油空间 O 在仿射 R^{-1} 作用下的前像空间 $R^{-1}(O)$。攻击者随机的对公钥多项式相关矩阵进行线性组合得到 $W=\sum_0^{o-1}\lambda_i\hat{P}_i$，用他们乘以某个 $\hat{P}_j$ 的逆。并求解其不变子空间。对普通 UOV，UOV 攻击的复杂度公式为 $q^{v-o-1}\cdot o^4$。循环 UOV 和 UOV 具有相同的前像空间 $R^{-1}(O)$。$W\cdot\hat{P}_j$ 拥有非平凡不变子空间的概率也等于 q^{o-v}。所以从理论上来看，循环 UOV 在 UOV 攻击下应该具有相同的性质。为了证明我们的猜想，我们随机生成了 100 组循环 UOV 和 UOV 实例，然后用 UOV 攻击来计算油空间的前向空间 $R^{-1}(O)$。表3-3记录了 UOV 攻击恢复循环 UOV 和 UOV 的前向空间 $R^{-1}(O)$ 所需要的时间。

从表 3-3中我们可以看出。UOV 攻击对循环 UOV 和普通 UOV 具有相同的攻击复杂度。所以我们可以得到循环 UOV 的 UOV 攻击复杂度公式：$q^{v-o-1}\cdot o^4$。当 v 和 o 的值比较接近的时候，UOV 攻击具有很高的效率。但是如果我们选择 $v\approx 2o$ 时，UOV

表 3-3 循环 UOV 和 UOV 的 UOV 攻击对比

(GF(31),o,v)	(8,12)	(10,14)	(8,14)	(10,16)
UOV	1.98s	2.42s	934.30s	1025.27s
循环 UOV	1.99s	2.39s	941.2s	1197.88s

攻击的复杂度是 o 的指数级别的。

3.5.5 小秩攻击

在小秩攻击中，攻击者搜索公钥多项式相关矩阵 $\hat{P}_i$ 的线性组合，找到其中秩小于等于 r 的矩阵 (r 为中心映射多项式相关矩阵的最小的秩)。在循环 UOV 中，$\hat{G}_i$ 中每个子矩阵 A_i 都是一个随机选取的 $v \cdot v$ 大小的矩阵。所以其中心映射多项式相关矩阵的最小秩 r 以极大的概率大于 v。要攻击这样的系统，我们需要找到一个秩小于等于 v 的 o 个 $n \cdot n$ 公钥多项式相关矩阵的线性组合。因为 $o < v < n$，所以找到这样的组合的最佳方法是暴利搜索。所以循环 UOV 最小秩攻击的复杂度大约为 $O(q^o n^3)$。

3.5.6 高秩攻击

高秩攻击本来是用来攻击 Rainbow 和类 Rainbow 的签名方案的。它可以被认为是最小秩攻击的对偶攻击。攻击者的目标是要找到一个被很多公钥多项式相关矩阵 $\hat{P}_i$ 的线性组合所共享的小核空间，进而找到 Rainbow 的子空间 $R^{-1}(O_t)$。这样攻击者就能够分离 Rainbow 中心映射的各层结构，然后像合法用户一样生成签名。高秩攻击无法攻击常规 UOV，因为常规 UOV 的中心映射多项式相关矩阵并不存在大多数多项式相关矩阵共享的小核空间。

如果循环 UOV 参数选择不恰当，其安全性会受到高秩攻击影响。令 $\hat{P}_h = \sum_0^{o-1} \lambda_i \hat{P}_i$，其中 λ_i 为基域上的随机元素。为了方便分析，我们首先将 R 仿射带来的影响消除。我们有：

$$R^{-T} \hat{P}_h R^{-1} = \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix},$$

其中 A 是 $v \cdot v$ 大小的矩阵，B 是 $v \cdot o$ 大小的矩阵。假设 A 是可逆的，那么我们有 $Rank(\hat{P}_h) = Rank(A) + Rank(B)$。在这个攻击场景中，UOV 和循环 UOV 的区别是：在 UOV 中矩阵 B 是一个随机矩阵，而在循环 UOV 中矩阵 B 可以被写成

$B = B_1 \sum_0^{o-1} \lambda_i' T^i$。其中 λ_i' 是基域上的随机元素，矩阵 T 是 $o \cdot o$ 大小的旋转矩阵:

$$T = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & 1 & 0 \\ 0 & \ddots & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

为了分析出循环 UOV 的高秩攻击复杂度，我们必须分析清楚矩阵 B 的秩。因为 B_1 是一个 $v \cdot o$ 大小的随机矩阵，并且 $v \approx 2o$，那么以极大概率我们有 $Rank(B_1) = o$。所以矩阵 B 的秩主要取决于矩阵 $\sum_0^{o-1} \lambda_i' T^i$。要分析矩该矩阵的秩，其实可以等价的分析矩阵 T 的特征多项式。

矩阵 T 在基域上的特征多项式是 $x^o - 1$。$\sum_0^{o-1} \lambda_i' T^i$ 的秩有如下等价关系:

$$Rank(\sum_0^{o-1} \lambda_i' T^i) = o - degree(gcd(x^o - 1, \sum_0^{o-1} \lambda_i' x^i))$$

对于一般的 q 和 o，攻击者通过暴力搜索可以找到一组 λ_i' 来让矩阵 B 不满秩。不满秩的 B 将会泄露出 $R^{-1}(O)$ 的一个 $o - Rank(B)$ 维子空间。使用高秩攻击的方法来找到一个这样的 d 维子空间的复杂度大概是 $O(q^d \frac{n^3}{6})$。攻击者通过找到很多这样的子空间，并将它们组合起来得到一个更大的 $R^{-1}(O)$ 的子空间 U。子空间 U 能够帮助攻击者降低“求逆” $\mathbf{y} = P(\mathbf{x})$ 的复杂度。

一个随机 $\sum_0^{o-1} \lambda_i' T^i$ 组合的秩，主要取决于特征多项式 $x^o - 1$ 在基域上的因式分解。假设 $x^o - 1$ 在基域上可以被分解成 $x^o - 1 = \sum_{k=0}^{t} f_k$。令 d_k 为多项式 f_k 的次数。假设 $d_o \leq d_1 \cdots \leq d_t$，对每个 d_k，攻击者可以使用如下方法攻击循环 UOV。

1) 令 $D_k = \{d_1, ..., d_k\}$。

2) 从 D_k 中取出第一个元素 d_j。随机生成线性组合 $\hat{P}_h = \sum_{t=0}^{o-1} \lambda_i \hat{P}_i$，求解其核空间 $U_j = ker(\hat{P}_h)$。

3) 如果 $dim(U_j) \geq 1$，令 $(\sum_{i=0}^{o-1} \lambda_i \hat{P}_i) U_j = 0$。如果解空间维度为 $o - d_j$，将 d_j 从 D_k 中删除。如果 $D_k \neq \emptyset$ 则跳回步骤 2。

4) 令 $o_k = \sum_{i=1}^{k} d_i$。将前面所得到的 $U_o, ..., U_k$ 合并成一个更大的空间 U，并使用 U 空间将循环 UOV 的公钥多项式转换成有 o_k 个油变量和 $n - o_k$ 个醋变量的油醋多项式形式。

5) 对于一个要“求逆”的向量 $\mathbf{y}$，随机选醋变量 $v_1, ..., v_{n-o_k} \in K$ 将其带入系统中。

此时攻击者能够获得关于 o_k 个变量的 o 个线性方程，这样的随机线性方程有解的概率为 q^{o-o_k}。所以攻击者每次有 q^{o-o_k} 的概率能够找到一个 $\mathbf{x}$ 满足 $P(\mathbf{x})=\mathbf{y}$。

从上面的攻击步骤我们可以看出，攻击者选择不同的 d_k 具有不同的攻击复杂度。总的攻击复杂度可以为:

$$HighRank(q,o)=\min_{d_k}(\frac{n^3}{6}\cdot(q^{d_1}+\cdots+q^{d_k}+q^{o-o_k})).$$

对于不同的参数 o 和 q，高秩攻击的复杂度差别很大。以基域 GF(31) 为例，当 $o=29$ 的时候高秩攻击的复杂度大约为 31^{28}。因为在域 GF(31) 上 $x^{29}-1=(x+30)(x^{28}+x^{27}+\cdots+x+1)$，随机的多项式很难与 $x^{29}-1$ 有一个高次公因式，所以其攻击复杂度高。但是如果我们选择 $o=28$ 作为参数，其高秩攻击的复杂度大约为 31^{16}。这是因为在域 GF(31) 上 $x^{28}-1=(x+1)(x+30)(x^2+1)(x^6+x^5+x^4+x^2+x+1)(x^6+10x^5+3x^4+10x^3+3x^2+10x+1)(x^6)$，一个随机的多项式有很大的概率与 $x^{29}-1$ 有高次公因式。为了防止高秩攻击，我们必须非常小心的选取 o 和 q。

3.5.7　其他攻击

从上面的分析中，我们可以得出结论：如果参数选择合适，循环 UOV 能够抵抗目前已知的所有针对 UOV 的攻击。也许有人会认为，可能存在一种特殊的攻击方法，能够利用循环 UOV 中心映射中的旋转结构来降低攻击复杂度。实际上，旋转关系在多变量公钥密码和一些其他公钥密码方案中广泛被使用并且非常难被利用 [69][70][71]，而且在循环 UOV 私钥中只有一小部分内容存在旋转关系。当仿射 S 作用到循环 UOV 中心映射上之后，这种旋转关系变得更加难利用了。

不幸的是，我们并不能给出循环 UOV 方案能规约到一个常见密码学困难问题的形式化证明 [72]。不过普通 UOV 签名方案提出二十多年以来，也没有可证明安全依然被认为是安全的。类似的，在基于格的密码方案中，NTRU 方案 [73] 的安全性依赖于格问题，但是也不能规约到一个常见的格上困难问题。存在一些可证明安全的 NTRU 变种方案 [74] 和一些其他可证明安全的格密码方案 [75]，但是这些方案的性能比 NTRU 差很多。在 MPKC 中，也存在一些可证明安全但是效率很低的方案。在本文中，我们选择通过研究具体的密码分析方法来确定循环 UOV 的安全性。

3.6　循环 UOV 的性能

本节我们将对循环 UOV 的性能进行理论分析和实验验证。首先我们将会对循环矩阵的可逆概率进行论证，然后基于循环 UOV 的性质和循环矩阵的可逆性质，为循环

UOV 选择合适的参数，并将其与普通 UOV 进行性能对比。

3.6.1 循环矩阵的可逆概率

为了更好的为循环 UOV 选取参数，我们必须了解清楚有限域上循环矩阵的可逆性质。目前对循环矩阵可逆性质的研究，基本都是在复数域上的。公开的文献中并没有关于有限域上循环矩阵的可逆性质的相关结论。本小节中，我们从理论上对循环矩阵进行研究，并给出其可逆概率的公式。

令 $Circ(o)$ 为域 K 上 $o \cdot o$ 的循环矩阵的全体集合。对任意 $L \in Circ(o)$，我们可以定义其多项式表示

$$l(x) = \sum_{t=0}^{o-1} l_i x^i$$

其中 $l_0, l_1, \cdots, l_{o-1}$ 为循环矩阵 L 的第一行元素。$Circ(o)$ 实际上是一个环 [76]，且与 $K[x]/(x^o - 1)$ 同构。其同构映射为

$$\Sigma : Circ(n) \to K[x]/(x^o - 1)$$

即将一个循环矩阵转换成其多项式表示形式。

由此我们可知，域 K 上随机 o 维循环矩阵可逆的概率等价于多项式环 $K[x]/(x^o-1)$ 上随机元素是乘法可逆元的概率。$Circ(o)$ 上的一个可逆矩阵，等价对应于 $K[x]/(x^o-1)$ 上的一个可逆元。以下我们探讨多项式环 $K[x]/(x^o - 1)$ 上元素可逆性质。

$K[x]/(x^o - 1)$ 上元素可逆性，取决于多项式 $x^o - 1$ 在域 K 上的分解。令 p 为域 K 的特征，这里我们要分两种情况讨论。

当 p 与 o 互素时，$x^o - 1$ 可分解为

$$x^o - 1 = \prod_{d|o} \Phi_d(x)$$

其中 $\Phi_d(x)$ 为域 K 上的 d 次分圆多项式 $\Phi_d(x) = \prod_{gcd(j,d)=1}(x - \zeta_d^j)$，$\zeta_d$ 为域 K 上的 d 次原根。$\Phi_d(x)$ 的次数为 $\phi(d)$ 且系数全部属于域 K 的 d 次分裂域。

同时由于当 p 与 o 互素的时候，p 与 o 的所有的因子也互素。由此我们可知 p 与上面所有的 d 都互素。据书籍 [77] 第 2.4 节中结论，此时 $\Phi_d(x)$ 可被分解为 $\frac{\phi(d)}{r_d}$ 个 $K[x]$ 上不同的 r_d 次首一不可约多项式 f_i。其中 r_d 为整数模 d 剩余环中 q 的乘法阶，满足 $q^{r_d} \equiv 1 \bmod n$。由此我们可以将 $(x^o - 1)$ 的分解写为

$$(x^o - 1) = \prod_{d|o} \prod_{i=1}^{\frac{\phi(d)}{r_d}} f_i(x)$$

且所有的 f_i 为不同的不可约多项式。由多项式中国剩余定义，我们可以推出

$$K[x]/(x^o-1) \cong \prod_{d|o} \prod_{i=1}^{\frac{\phi(d)}{r_d}} K[x]/f_i(x).$$

要求 $K[x]/(x^o-1)$ 中可逆元的个数，等价于求环 $\prod_{d|o} \prod_{i=1}^{\frac{\phi(d)}{r_d}} K[x]/f_i(x)$ 中可逆元的个数。$\prod_{d|o} \prod_{i=1}^{\frac{\phi(d)}{r_d}} K[x]/f_i(x)$ 实际是多个有限域进行笛卡尔乘积，故我们可计算的，$K[x]/(x^o-1)$ 中可逆元的个数为

$$\prod_{d|o} (q^{r_d}-1)^{\frac{\phi(d)}{r_d}}.$$

故可得出结论，当 p 与 o 互素时，$K[x]/(x^o-1)$ 元素可逆的概率为 $\frac{\prod_{d|o}(q^{r_d}-1)^{\frac{\phi(d)}{r_d}}}{q^o}$。

当 p 与 o 不互素时，我们可以类似的进行分解。(x^o-1) 可分解为

$$x^o-1 = x^{mp^t}-1 = (x^m-1)^{p^t} = \prod_{d|o} \Phi_d(x)^{p^t} = \prod_{d|m} \prod_{i=1}^{\frac{\phi(d)}{r_d}} f_i(x)^{p^t}.$$

由此我们可以得到环同构

$$K[x]/(x^o-1) \cong \prod_{d|m} \prod_{i=1}^{\frac{\phi(d)}{r_d}} K[x]/f_i(x)^{p^t}.$$

要求解 $K[x]/(x^o-1)$ 中可逆元的个数，等价于求解环 $\prod_{d|m} \prod_{i=1}^{\frac{\phi(d)}{r_d}} K[x]/f_i(x)^{p^t}$ 中可逆元的个数。$\prod_{d|m} \prod_{i=1}^{\frac{\phi(d)}{r_d}} K[x]/f_i(x)^{p^t}$ 实际是多个有限域进行笛卡尔乘积，故我们可计算的，$K[x]/(x^o-1)$ 中可逆元的个数为

$$\prod_{d|m} [(q^{r_d}-1) \cdot q^{d(p^t-1)}]^{\frac{\phi(d)}{r_d}}.$$

顾可得出结论，当 p 与 o 不互素时，$K[x]/(x^o-1)$ 元素可逆的概率为

$$\frac{\prod_{d|m} [(q^{r_d}-1) \cdot q^{rd\cdot(p^t-1)}]^{\frac{\phi(d)}{r_d}}}{q^o}.$$

由此我们可以计算出以常见 UOV 参数中 K 和 o 为参数的循环矩阵的可逆概率。图3-3给出了域为 GF(8)、GF(16)、GF(31)、GF(32) 和 GF(256) 时，o 维随机循环矩阵是可逆矩阵的概率 $P(o)$。

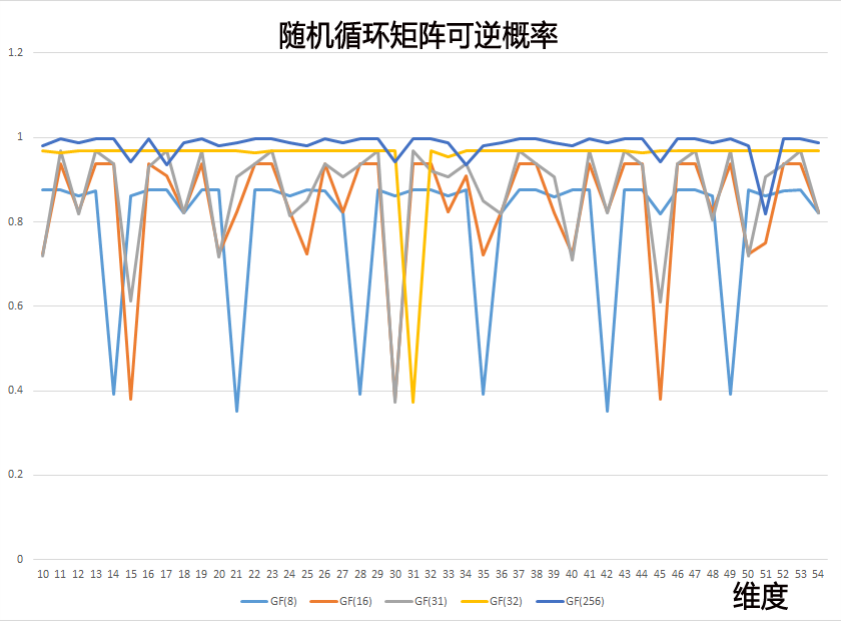

图 3-3 常见域上随机循环矩阵的可逆概率

由图3-3，我们可以看出大多数 o 的取值都能保证 $P(o)$ 的值很大。当 K 的阶越大的时候，$P(o)$ 的值越大。

3.6.2 循环矩阵与普通矩阵对比

上一小节，我们研究了有限域上循环矩阵的可逆性质。这一小节我们将对有限域上循环矩阵和普通矩阵的可逆性质进行对比。

令 q 为有限域 K 阶，则域 K 上秩为 r 的 $o \cdot o$ 矩阵的个数为 $\frac{q^{r(r-1)/2}\prod_{i=o-r+1}^{o}(q^i-1)^2}{\prod_{i=1}^{r}(q^i-1)}$。令 A 为域 K 上的 $o \cdot o$ 维随机矩阵，A 矩阵秩为满秩的概率为 $\frac{q^{o(o-1)/2}\prod_{i=o-r+1}^{o}(q^i-1)^2}{q^{o^2}\prod_{i=1}^{r}(q^i-1)}$。对每个固定的域该值为常数。这里我们对域 GF(31) 和 GF(256) 上的循环矩阵和普通矩阵的可逆概率进行对比。图3-4和图3-5分别给出了 GF(31) 和 GF(256) 上循环矩阵和普通矩阵的可逆概率对比。

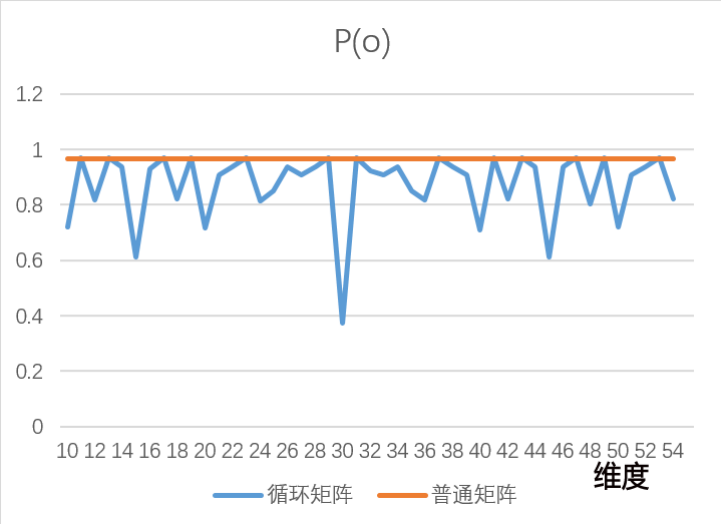

图 3-4 GF(31) 上循环矩阵和普通矩阵的可逆概率对比

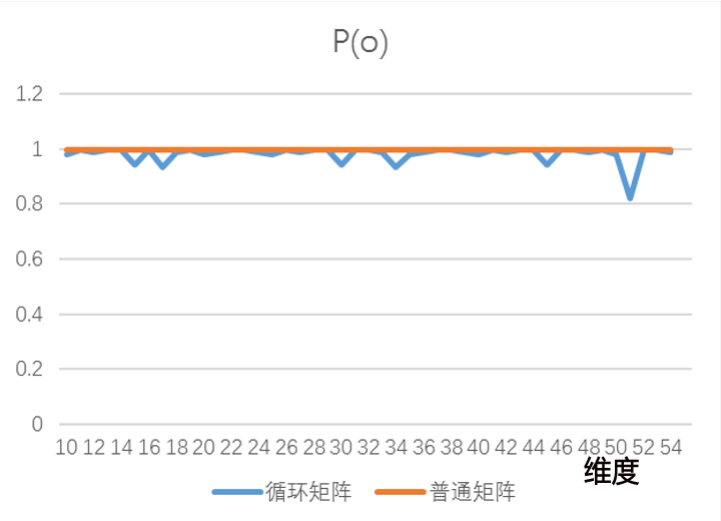

图 3-5 GF(256) 上循环矩阵和普通矩阵的可逆概率对比

由图3-4和3-5我们可以看出，有限域上普通矩阵是满秩矩阵的概率是一个常数，在多数情况下随机的普通矩阵是可逆的概率比随机的循环矩阵可逆概率要大。但当我们参数选择合适的时候，随机的循环矩阵可逆概率和随机的普通矩阵可逆概率差别极小，当选取一些特殊的 o 的时候，随机的循环矩阵的可逆概率甚至会大于普通矩阵的可逆概率。在进行 UOV 参数选择的时候，我们可以挑选可逆概率更大的 o 值。GF(31) 和 GF(256) 两个域对比来看，我们发现当域的阶更大的时候，随机循环矩阵的可逆概率和随机普通矩阵的可逆概率越接近。GF(256) 上，随机循环矩阵的可逆概率和随机普通矩阵的可逆概率差别很小。总的来说，循环 UOV 的参数选择不能像普通 UOV 一样灵活，尤其是在如 GF(31) 这样的小域上，选择参数的时候要格外注意。

3.6.3 减方法

在文章 [78] 中，作者提出了基域为 GF(31) 的 UOV 方案的安全参数。从上面的讨论，我们可以知道循环 UOV 可以使用这些参数来防止直接攻击。但是直接使用这些参数并不合适，因为为了防止高秩攻击和提高矩阵 L 的可逆概率，循环 UOV 对 q 和 o 的选择有严格的限制。我们需要适当的增大 o 来适应这些要求。这样会导致 v 与 o 的比值稍微降低，影响 UOV 攻击的复杂度。不过由于 UOV 攻击的复杂度为 $q^{v-o-1} \cdot o^4$，相比于其他攻击方法的复杂度要高很多，所以我们的稍微增大 o 的大小并不会影响安全性。

o 的增大，会增大循环 UOV 的公钥大小和签名验证时间。为了解决这个问题，我们引入减方法到循环 UOV 中。减方法首次在 [79] 中被提出。其主要思路是通过删掉多变量公钥密码公钥系统中的部分多项式，来减少多变量密码公钥系统中的一些线性性质，这样可以增加 MPKC 方案的安全性。减方法被广泛运用在如 C*、HFE 等多种大域 MPKC 方案中 [80]。在循环 UOV 中，我们删除掉 1 到 2 个公钥多项式来使循环 UOV 和普通 UOV 具有相同的多项式数量。因为直接攻击的复杂度主要取决于多项式的数目，所以这不会影响直接攻击的安全性。当运用了减方法之后，攻击者实际上获得了更少的关于私钥的信息，所以减方法并不会让其他利用循环 UOV 结构的攻击变得容易。

3.6.4 与普通 UOV 对比

根据前面我们对循环 UOV 安全性的讨论，我们为循环 UOV 选择安全级别为 2^{80}、2^{100} 和 2^{128} 的参数并将其与普通 UOV 进行试验对比。

在选择了合适的参数后，我们对循环 UOV 和 UOV 进行了 C++ 代码的实现。我们所有的实验都是在 Intel Core i7-4790@3.6Ghz CPU 上进行。AVX2 指令集被用来提速多项式求值和矩阵向量乘法操作 [81]。对每组参数我们进行了 10^4 次测试并记录他们的平均性能。表3-4 记录了具体的实验数据。在每组对比数据中，循环 UOV 和普通 UOV 具有相同的变量数 n 和公钥多项式数 m，所以他们具有相同的签名验证速度，公钥大小和签名长度。此外，循环 UOV 的私钥大小比普通 UOV 降低了大约 45%，签名速度更是明显优于 UOV。

为了进一步理解循环 UOV 速度上的优势，我们给出循环 UOV 和普通 UOV 的签名速度比值图 3-6。相比于普通 UOV，循环 UOV 在现代 CPU 上的签名速度大约是普通 UOV 的 8 倍。在我们的实验中，扩展欧几里得算法几乎是高斯消元算法速度的 25 倍。当 o 和 v 比较小的时候，扩展欧几里得算法相对于高斯消元的提速相对较小，系统总的

表 3-4 GF(31) 上循环 UOV 与普通 UOV 性能对比

	安全级别 (bit)	参数 (o,v)	私钥大小 size(kB)	签名周期 (10^3 cycles)	密钥生成周期 (10^3 cycles)
UOV	80	(33,66)	96.5	1,893	44,964
	100	(41,82)	181.7	3,394	107,532
	128	(52,104)	364.9	4,093	299,160
循环 UOV	80	(34,65)	53.9	252	149,472
	100	(43,80)	99.6	336	373,932
	128	(53,103)	196.5	454	732,888

提速也相对较小。当 o 和 v 适中，普通 UOV 签名算法的主要复杂度取决于高斯消元，这个时候循环 UOV 对普通 UOV 的签名提速最大。当 o 和 v 变大的时候，UOV 签名算法中其他部分的负载变大了，这导致循环 UOV 相对于普通 UOV 的签名提速被降低。

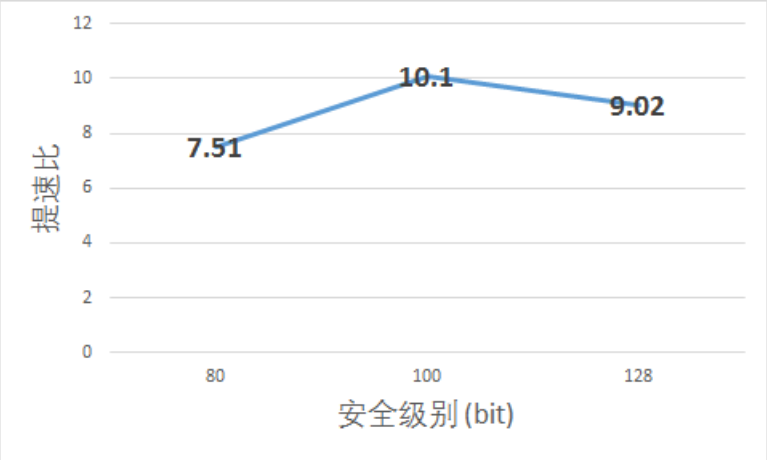

图 3-6 循环 UOV 相对于 UOV 的签名提速

3.6.5 与其他 UOV 变种对比

本小节中，我们将循环 UOV 与其他 UOV 变种进行对比。在文章 [78] 中，作者在 2.53 GHz CPU 主频的 CPU 上实现了 Cyclic UOV。直接将他的运行时间与本文的运行时间进行对比是不公平的，因为本文中我们使用的 CPU 架构具有 AVX2 指令集 (SIMD 硬件加速功能)。不过因为我们都对普通 UOV 进行了实现，我们可以用普通 UOV 的实验数据作为基础，得出 Cyclic UOV 在我们的计算模型中的性能结果。

尽管基于矩阵的方法（MB 法）和 NT 法最初是为 Rainbow 签名方案设计的，不过因为 UOV 可以看成是一个单层 Rainbow 方案，这些方案也可以用到 UOV 上。虽然其在安全性上存在一定问题，我们仍对其进行实现并记录性能。我们的实验结果记录在

表 3-5中。表3-5中记录了 2^{80} 安全性下，循环 UOV 和其他 UOV 变种的性能数据。

表 3-5 循环与其他 UOV 变种进行对比

方案	参数 (o,v)	公钥 (kB)	私钥 (kB)	签名 (bit)	密钥生成周期 (10^3 cycles)	签名周期 (10^3 cycles)	验证周期 (10^3 cycles)
UOV	(33,66)	101.7	96.5	495	44,964	1,893	43
Cyclic UOV	(33,66)	17.1	96.5	495	22,291,200	1,893	10
MB UOV	(33,66,d=3)	101.7	54.1	495	151,284	1,242	43
NT UOV	(33,66)	101.7	67.9	495	136,814	1,287	43
循环 UOV	(34,65)	101.7	53.9	495	149,472	252	43

很明显我们可以看出，循环 UOV 在签名时间、私钥大小上是最优的。而 Cyclic UOV 则是在验证时间、公钥大小这两方面更优。普通 UOV 在密钥生成时间上有优势。从表3-5中我们可以得出结论，循环方法在降低密钥大小和提高签名速度上要优于 MB 法和 NT 法。

由于 Rainbow 可以被认为是一个多层的 UOV 结构，人们可能很自然的会想到想要把循环 UOV 的方法运用到 Rainbow 上。但是由于 Rainbow 的多层特殊结构，在参数选择和安全性分析上我们需要更细致的处理。我们将在下一章关注这个问题。

3.7 小结

在本章中，我们提出了一种新的 UOV 签名变种方案。通过在普通 UOV 的中心映射矩阵中引入一些旋转关系，它可以拥有更快的签名速度和更短的私钥。

在安全性上，我们对有所的已知攻击方法都进行了理论分析，并用实验结果验证了理论分析的结论。同时我们引入减方法到循环 UOV 中，并为其选择了合适的安全参数。通过对相同安全级别循环 UOV 和 UOV 的实验比较，我们可以看到循环 UOV 的私钥大小比普通 UOV 大约减少了 45%，其签名速度达到普通 UOV 的 7 到 10 倍。

第四章　一种基于循环矩阵的 Rainbow 签名方案

Rainbow 是 UOV 的一种多层扩展方案。其具有相当于 UOV 的安全性，同时相比于 UOV 具有更高的签名效率，更短的公私钥和更短的签名。尽管如此，Rainbow 还是由于密钥太大而没有被广泛使用。降低 Rainbow 的密钥大小，是 MPKC 的一个重要研究方向。

由于 Rainbow 的特殊性质，目前已经有不少 Rainbow 的变种方案。在文章 [71][49] 中，作者提出了 Cyclic Rainbow 来降低 Rainbow 的公钥大小和验证时间。另外一些 Rainbow 的变种方案如 [82][54][53]，通过引入稀疏密钥结构来降低 Rainbow 的密钥大小并提高签名速度。尽管加强版 TTS 已经被攻破了 [83]，但是使用稀疏密钥的 Rainbow 方案还存留至今 [55][54][82]。本章中，我们将分析 MB Rainbow 和 NT Rainbow 两种 Rainbow 变种的安全性，并提出一个新的基于循环矩阵的 Rainbow 变种方案。我们将其命名为循环 Rainbow。

4.1　Rainbow 变种方案

在文章 [71] 中，作者提出在 Rainbow 的公钥中插入了一些特殊的序列来节省存储空间。Cyclic Rainbow 是其中一种特殊的实例，它将具有循环关系的序列插入到 Rainbow 的公钥中来降低公钥的大小。同时，通过使用循环关系，Cyclic Rainbow 还能极大的提高签名验证的速度。

另外一些 Rainbow 的变种方案则关注于降低私钥大小和提高签名速度。增强版 TTS 被 Yang 和 Chen 与 2005 年在文章 [53] 中被提出。其总的思路是要使用稀疏的多层 UOV 陷门结构来提高签名速度并降低私钥大小。它可以被认为是一种变形的 Rainbow 签名方案。不过它因为缺乏足够多的油变量和醋变量交叉项而被一种变种的彩虹带分离攻击方法所攻破 [83]。尽管如此，使用稀疏密钥来降低 Rainbow 密钥大小并提高签名速度的方法仍然被很多人使用。2013 年，Yasuda 等人提出了基于矩阵的 Rainbow(MB Rainbow)[54]。通过将 Rainbow 中心映射的每一层中心映射相关矩阵变成能用准对角矩阵表示更小的块，私钥的大小可以被降低 40%，签名的速度能够提高 40%。2014 年，Yasuda 等人又提出了 NT Rainbow[82]。其主要思想是通过将旋转关系引入到 Rainbow 中心映射的醋醋项中来降低 Rainbow 的私钥大小。NT Rainbow 的基本原理可以结合和 MB Rainbow 的基本原理一起使用以获得更高的效率 [55]。然而，我们发现 MB Rainbow

容易受到一种变种的彩虹带分离攻击的攻击，而 NT Rainbow 在原文中的安全参数 [55] 不足以抵抗彩虹带分离攻击。为了让 NT Rainbow 和 MB Rainbow 达到 2^{80} 和 2^{100} 的安全级别，我们必须修正它们的参数。

4.1.1 对 MB Rainbow 和 NT Rainbow 的彩虹带分离攻击

对于参数为 (K,v_1,o_1,o_2) 的 Rainbow 签名方案，彩虹带分离攻击的目的是要找到一个等价密钥 S' 和 R' 满足

$$F = S \circ G \circ R = S' \circ G' \circ R',$$

其中 G' 是一个可用的 Rainbow 中心映射，S' 和 R' 具有图 4-1 所有的特殊结构。

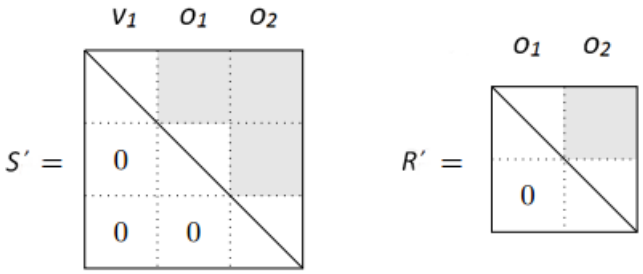

图 4-1 普通 Rainbow 的等价密钥

在图4-1中，白色部分表示零元素，灰色部分表示任意元素，对角线元素为 1。

在彩虹带分离攻击中，存在满足图 4-2形式的“好密钥”(good keys) S'_n 和 R'_n 。

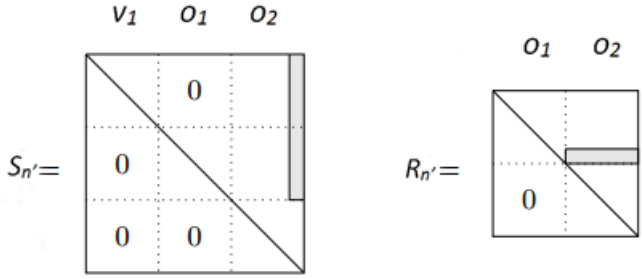

图 4-2 普通 Rainbow 的好密钥

矩阵 S'_n 只有前两个块的最后一列存在任意元素，其值等于 S' 中对应位置的值。对应的，矩阵 R'_n 中只有第二块的第 $o-1$ 行中存在任意元素，其值等于 R' 中对应位置的值。当普通 Rainbow 公钥系统 P 作用上好密钥 S'_n 和 R'_n 之后，$(S'^{-1}_n \circ P) \circ R'^{-1}_n$ 将会具有如图 4-3的形式。

令 S'_n 和 R'_n 中的未知元素为未知数，从图 4-3中我们可以得到 n 个变量的一条三次方程，$m+n-2$ 条二次多项式方程。我们可以使用求解 Gröbner 基的方法来求解这个方程组。这里我们使用 F5 算法来求解，其复杂度可以通过计算该方程组的正则度

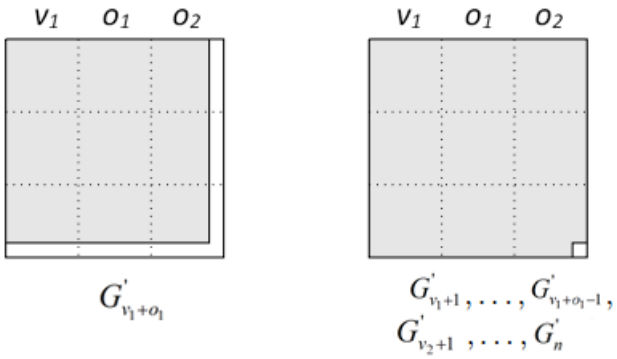

图 4-3 普通 Rainbow 作用了好密钥之后的中心映射

d_{reg}[58] 来估量。其正则度 d_{reg} 是希尔伯特序列

$$S_{m,n} = \frac{\prod_{i=1}^{m}(1-z^{d_i})}{(1-z)^n},$$

的第一个非负项的索引。使用 F5 算法来求解这个系统的复杂度为

$$O(\binom{n+d_{reg}}{d_{reg}}^{\omega}),$$

其中 n 是变量数，m 是方程数量，ω 是线性代数常量，在密码分析中我们通常取 $\omega = 2$。

在 NT Rainbow 文章中 [82]，作者建议 2^{80} 安全级别 NT Rainbow 参数为 (GF(256),18,14,14)。运用彩虹带分离攻击，我们可以得到关于 46 个变量的一个三次方程和 72 个二次方程。求解这个方程组的复杂度大约为 2^{70}，这实际上低于作者的预期。

对于 MB Rainbow 文章中 [54]，作者建议 2^{100} 安全级别 MB Rainbow 的参数为 (GF(256),31,19,2*12)。运用彩虹带分离攻击，我们可以得到关于 74 个变量的一个三次方程和 155 个二次方程。求解这个方程组的复杂度大约为 2^{110}，这看上去已经满足了安全要求。但其实常规的彩虹带分离攻击，并没有充分利用 MB Rainbow 的特殊结构。在参数为 $(K, v_1, o_1, d * o_2')$ 的 MB Rainbow 中心映射中，最后一层多项式相关矩阵有更多的零列。当我们将好密钥 $S_n'^{-1}$ 和 $R_n'^{-1}$ 作用到 MB Rainbow 的公钥系统 P 上的时候，$(S_n'^{-1} \circ P) \circ R_n'^{-1}$ 有如图 4-5一样的结构。

对比图 4-5和图 4-3，我们可以看出图 4-5的左边矩阵的最后一行和最后一列全部为零元素。这样在进行彩虹带分离攻击的时候，我们就可以获得更多的二次方程，从而降低求解 Gröbner 基的复杂度。在对参数为 $(K, v_1, o_1, d * o_2')$ 的 MB Rainbow 的彩虹带分离攻击中，我们可以得到关于 n 个变量的一个三次方程和 $(n-1)*(m-o_1-o_2'+1)+(m+1)$ 个二次方程。具体的，对参数为 (GF(256),31,19,2*12) 的 MB Rainbow 进行我们的变种彩虹带分离攻击，我们可以得到关于 74 个变量的一个三次方程和 993 个二次方程。使

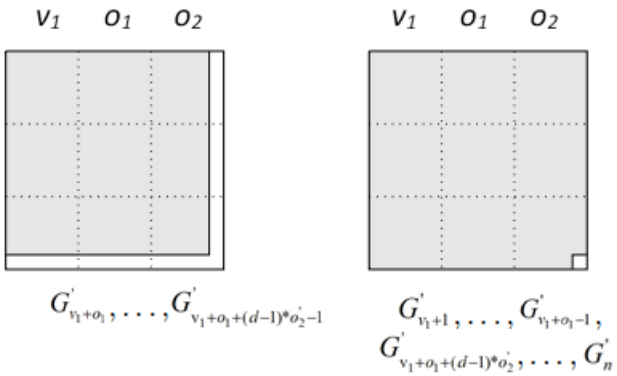

图 4-4 MB Rainbow 作用了好密钥之后的中心映射

用 F5 算法求解这个系统的复杂度大约是 2^{33}，这远低于原本的彩虹带分离攻击的复杂度。

讨论：MB Rainbow 的中心映射最后一层的多项式相关矩阵相比于普通 Rainbow 有更多的零列，这样的稀疏结构导致攻击者在进行彩虹带分离攻击的时候能够获得更多的等式。为了防止这样的攻击，我们应该避免在 MB Rainbow 中心映射的最后一层中引入基于矩阵的稀疏结构。幸运的是，MB Rainbow 中心映射的其他层仍然能够使用基于矩阵的结构来加速签名过程。我们可以重新选择 MB Rainbow 的参数来达到想要的安全级别。NT Rainbow 目前已有的参数不足以抵抗彩虹带分离攻击，我们将在第4.7.2节修正 MB Rainbow 和 NT Rainbow 的参数选择。

4.2 循环 Rainbow

在本节中，我们将提出一种新型的 Rainbow 签名方案。我们将其命名为循环 Rainbow[84]。在上一章中，我们提出了循环 UOV。因为 Rainbow 是多层 UOV 签名方案，所以很自然的，我们会想到把循环矩阵方法也运用到 Rainbow 上。循环 Rainbow 可以看成循环 UOV 的扩展方案，它拥有循环 UOV 的中心映射结构。相比于普通的 Rainbow，循环 Rainbow 具有更短的私钥和更快的签名速度。相比较于循环 UOV，循环 Rainbow 具有更短的签名和公私钥、更快的签名速度和验证速度。

4.2.1 基本思路

循环 Rainbow 是要构造一个多层的循环 UOV 签名方案。其核心思想是修改 Rainbow 签名生成算法 2-1中线性方程的形式，以获得更快的签名生成速度。正如算法 2-1中所描述的，在 Rainbow 签名生成中我们需要对 Rainbow 中心映射的每一层进行求解。假设在求解第 i 层的时候，我们需要求解方程 $L\mathbf{x} = \mathbf{u}$。L 是一个 $o_i \cdot o_i$ 大小的方

阵，$\mathbf{u}$ 是 o_i 维列向量，$\mathbf{x}$ 是 o_i 维的位置数向量。一般情况下，我们需要使用高斯消元来求解方程组。其复杂度大约为 $O(o_i^3)$。

4.3　循环 Rainbow 的中心映射

在循环 Rainbow 中，我们引入了旋转关系到 Rainbow 中心映射的每层多项式相关矩阵之中，这样的旋转关系会使得矩阵 L 变成一个循环矩阵，我们可以使用扩展欧几里得法来对 $L\mathbf{x} = \mathbf{u}$ 进行快速求解。其复杂度大约为 $O(o_i^2)$。此外，旋转结构的引入还能够降低私钥大小和算法 2-1中其他部分的计算负载。

4.3.1　循环 Rainbow 的中心映射结构

为了让 L 变成循环矩阵，我们在 Rainbow 中心映射每一层的多项式相关矩阵中加入旋转关系。这里我们给出循环 Rainbow 中心映射的 i 层多项式相关矩阵表示形式。我们保留常数项和线性项，i 层多项式相关矩阵为 $(v_i + o_i + 1) \cdot (v_i + o_i + 1)$ 的方阵如图4-5所示。

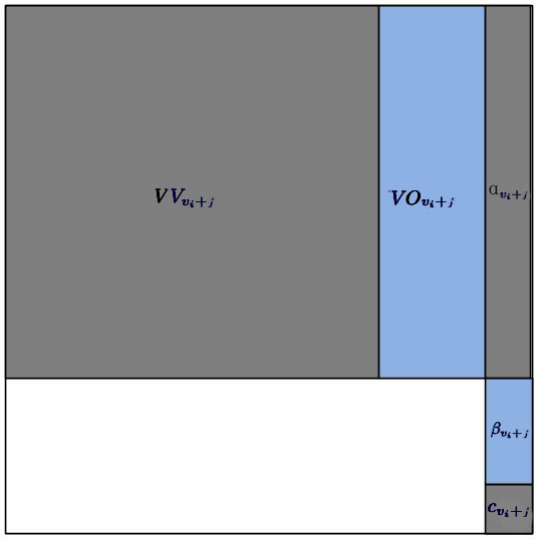

图 4-5　MB Rainbow 中心映射第 i 层多项式的矩阵表示

白色的部分表示零元素，灰色的部分表示基域上的任意元素，蓝色的部分表示该部分与同层的其他多项式相关矩阵存在旋转关系。其中 VV_{v_i+j} 是 $v_i \cdot v_i$ 的方阵，表示第 i 层醋醋项的系数。VO_{v_i+j} 是 $v_i \cdot o_i$ 的矩阵，表示第 i 层油醋交叉项的系数。最后一列的 α_{v_i+j} 表示醋变量的线性系数，β_{v_i+j} 表示油变量的线性系数。c_{v_i+j} 表示多项式的常数项。实际上，循环 Rainbow 中心映射的每一层多项式相关矩阵表示都是一个循环 UOV 的多项式相关矩阵表示。

循环 Rainbow 中心映射本身是不存在循环矩阵的，但是我们在每层多项式相关矩

阵之中引入的旋转关系能够帮我们在“求逆”中心映射的时候得到多个循环矩阵。这里我们将矩阵 VO_{v_i+j} 写成列向量形式，VO_{v_i+j} 和 β_{v_i+j} 有如下旋转关系：

$$\begin{aligned}
VO_{v_i+1} &= (\mathbf{vo}_{i,1}, \mathbf{vo}_{i,2}, \cdots, \mathbf{vo}_{i,o_i}) \\
VO_{v_i+2} &= (\mathbf{vo}_{i,o_i}, \mathbf{vo}_{i,1}, \cdots, \mathbf{vo}_{i,o_i-1}) \\
&\vdots \\
VO_{v_i+o_i} &= (\mathbf{vo}_{i,2}, \mathbf{vo}_{i,3}, \cdots, \mathbf{vo}_{i,1}) \\
\beta_{v_i+1} &= (\beta_{i,1}, \beta_{i,2}, \cdots, \beta_{i,o_i})^T \\
\beta_{v_i+2} &= (\beta_{i,o_i}, \beta_{i,1}, \cdots, \beta_{i,o_i-1})^T \\
&\vdots \\
\beta_{v_i+o_i} &= (\beta_{i,2}, \beta_{i,3}, \cdots, \beta_{i,1})^T.
\end{aligned}$$

4.3.2 循环 Rainbow 中心映射大小

基于前面的描述，我们这里给出构造循环 Rainbow 中心映射的具体参数，并计算其大小。

1) VO_{v_i+1}: $i \in [1, \cdots, t]$，每层初始的 $v_i \cdot o_i$ 矩阵，用来表示该层中醋变量和油变量的交叉项系数。

2) β_{v_i+1}: $i \in [1, \cdots, t]$，每层初始的 o_i 维向量，用来表示该层中油变量的线性项系数。

3) VV_{v_i+j}：$i \in [1, \cdots, t]$ $j \in [1, \cdots, o_i]$，每层 o_i 个随机的 $v_i \cdot v_i$ 矩阵，用来表示醋变量的二次系数。

4) α_{v_i+j}：$i \in [1, \cdots, t]$ $j \in [1, \cdots, o_i]$，每层 o_i 个 v_i 维向量，用来表示油变量的线性系数。

5) c_{v_i+j}：$i \in [1, \cdots, t]$ $j \in [1, \cdots, o_i]$，每层 o_i 个中心映射多项式的常数项。

相比较于普通 Rainbow，循环 Rainbow 主要是油醋交叉项和油变量线性项上能够节约部分存储空间。对于实际参数，其私钥大小大概降低 45%。

4.4 循环 Rainbow 中心映射“求逆”

因为循环 Rainbow 中心映射 G 实际上普通 Rainbow 中心映射的一种特例，所以我们同样可以通过算法 2-1的方法来“求逆”循环 Rainbow。在本节中，我们主要介绍如何快速“求逆”循环 Rainbow 的中心映射。

假设要“求逆”的向量是 **y**，总的来讲，依然是使用选定初始醋变量 $\mathbf{v_1}$ 然后逐层求解 Rainbow 中心映射。不过在求解每层油变量 $\mathbf{o_i}$ 的时候，我们利用了循环 UOV 中的快速“求逆”陷门来提高计算速度。以下是具体的介绍。

假设第 i 层的醋变量是 $\mathbf{v} = (v_1, \ldots, v_{v_i})$。我们用 $(v_1, \ldots, v_{v_i})$ 替换变量 $(x_1, \ldots, x_{v_i})$ 带入到第 i 层的中心映射多项式中，这使我们会得到关于 o_i 个变量的线性系统。对于第 i 层的每个多变量多项式 g_{v_i+j}，我们能够得到一个方程：

$$\underbrace{\mathbf{v}^T \cdot VV_{v_i+j} \cdot \mathbf{v} + \mathbf{v}^T \cdot \alpha_{v_i+j} + c_{v_i+j}}_{\text{常数项}} + \underbrace{\mathbf{v}^T \cdot VO_{v_i+j} \cdot \mathbf{o} + \beta_{v_i+j} \cdot \mathbf{o}}_{\text{线性项}} = y_{v_i+j},$$

其中向量 $\mathbf{o} = (x_{v_i+1}, \ldots, x_{v_i+o_i})$ 表示该层的油变量。令

$$u_{v_i+j} = y_{v_i+j} - (\mathbf{v}^T \cdot VV_{v_i+j} \cdot \mathbf{v} + \mathbf{v}^T \cdot \alpha_{v_i+j} + c_{v_i+j})$$

($j \in [1, \cdots, o_i]$)，我们能够得到一个线性系统：

$$\underbrace{\begin{pmatrix} \mathbf{v}^T \cdot VO_{v_i+1} + \beta_{v_i+1} \\ \mathbf{v}^T \cdot VO_{v_i+2} + \beta_{v_i+2} \\ \vdots \\ \mathbf{v}^T \cdot VO_{v_i+o_i-1} + \beta_{v_i+o_i-1} \\ \mathbf{v}^T \cdot VO_{v_i+o_i} + \beta_{v_i+o_i} \end{pmatrix}}_{\mathrm{L}} \begin{pmatrix} x_{v_i+1} \\ x_{v_i+2} \\ \vdots \\ x_{v_i+o_i-1} \\ x_{v_i+o_i} \end{pmatrix} = \begin{pmatrix} u_{v_i+1} \\ u_{v_i+2} \\ \vdots \\ u_{v_i+o_i-1} \\ u_{v_i+o_i} \end{pmatrix}.$$

因为矩阵 VO_{v_i+j} 之间和向量 β_{v_i+j} 之间存在旋转关系，当我们带入醋变量向量 **v** 之后，矩阵 L 会变成一个循环矩阵，可以被快速求逆。

4.4.1　计算 L

实际上“求逆”每一层循环 Rainbow 中心映射的过程，都像是在“求逆”一个循环 UOV 中心映射。因为 L 矩阵是一个循环矩阵，所以求解 L 矩阵实际上只需要将其第一行 $\mathbf{v}^T \cdot VO_{v_i+1} + \beta_{v_i+1}$ 求解出来就可以了。这只是一个矩阵向量乘法，相比较于普通 Rainbow 进行多项式求值计算，速度提高了 o_i 倍。

4.4.2　L 的可逆概率

在 Rainbow 签名算法中，如果某一层的 L 矩阵不可逆，那签名者就必须重新选择初始醋向量并重新进行计算。为了保证能够获得更快的签名算法，就必须要求矩阵 L 的可逆概率足够大。

对于不同的基域和参数 o，我们随机生成 10^5 个随机循环矩阵，然后计算他们的平均可逆概率。

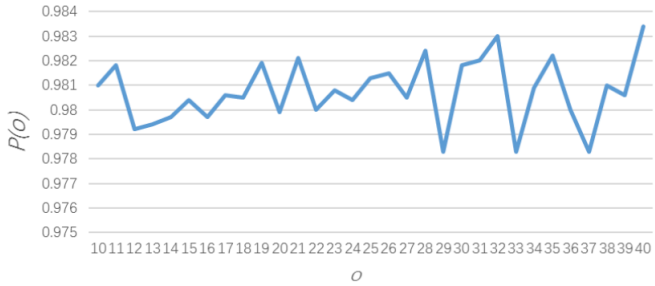

图 4-6 GF(256) 上随机循环矩阵的可逆概率

从图 4-6中我们可以看出，基域为 GF(256) 的时候矩阵 L 可逆的概率非常接近于 1。这个值也与我们在第3.6.1节所给出的公式吻合。故当我们选择合适的参数时，循环 L 的可逆概率很高。

4.5 循环 Rainbow 的一般性描述

使用上一节构造的中心映射 G，这里我们给出循环 Rainbow 的一般性描述。

密钥生成：根据所需的安全级别，选择合适的参数 $(K,v_1,o_1,\ldots,o_t)$。然后按第4.3节中介绍的方法构造一个二次中心映射 G，和两个可逆仿射 S: $K^m \to K^m$ 和 R: $K^n \to K^n$。计算出公钥 $P = S \circ G \circ R : K^n \to K^m$，密钥则为 (S,G,R)。

签名生成：假设要签名的消息为 **m**，我们用如下步骤对 **m** 进行签名：

1) 计算消息摘要：$hash(\mathbf{m}) = \mathbf{w} \in K^n$。

2) 求逆 S 仿射：$\mathbf{y} = S^{-1}(\mathbf{w})$。

3) 使用循环矩阵方法“求逆”中心映射：$\mathbf{x} = G^{-1}(\mathbf{y})$。

4) 求逆 R 仿射 $\mathbf{s} = R^{-1}(\mathbf{x})$。

签名验证: 签名者发送文档 -签名对 **(m,s)** 给验证者。验证者判断等式 $P(\mathbf{s}) = Hash(\mathbf{m})$ 是否成立. 如果成立，则签名认为是正确的；如果不成立，则签名认为签名是伪造的。

4.6 循环 Rainbow 安全性分析

在本节中，我们将已知的针对 MPKC 签名方案的攻击方法运用到循环 Rainbow 上来分析循环 Rainbow 的安全性。

在循环 Rainbow 中，公钥 P=$S \circ G \circ R$ 的取值实际上是普通 Rainbow 公钥的一个子集。所有能够攻击普通 Rainbow 的方法都能被用来攻击循环 Rainbow。但是如果我们选择参数合适，循环 Rainbow 将具有和普通 Rainbow 一样的安全性。

4.6.1　直接攻击

Rainbow 的公钥是一个欠定二次多变量多项式系统，对其进行直接攻击的最佳方法是先随机选取一些变量来构成一个超定系统，然后使用基于 Gröbner 基的方法来求解系统。这里我使用混合求解法来对 Rainbow 和循环 Rainbow 公钥系统进行直接攻击，其中我们选择 MAGMA 上的 F4 算法来进行 Gröbner 基的求解。对于每组被测参数，我随机生成 100 个循环 Rainbow 和 Rainbow 的实例，然后计算混合求解法的平均耗时。图3-2给出了在基域 GF(256) 上对循环 Rainbow 和 Rainbow 进行直接攻击的时间。

表 4-1 GF(256) 上循环 Rainbow 和普通 Rainbow 的直接攻击耗时

参数 (v_1,o_1,o_2)	Rainbow	循环 Rainbow
(4,2,2)	0.01 s	0.01 s
(4,3,3)	0.07 s	0.07 s
(6,4,4)	3.12 s	3.10 s
(6,5,5)	133.73 s	134.41 s
(6,6,6)	5743.53 s	5709.81 s

从表4-1中我们可以看出，对循环 Rainbow 和 Rainbow 进行直接攻击的复杂度并没有明显区别。所以我们可以得出循环 Rainbow 的直接攻击复杂度公式为

$$\min_{k\geq 0} q^k \cdot O(m \cdot \binom{m-k+d_{reg}+1}{d_{reg}}^{\omega}),$$

其中正则度 d_{reg} 为希尔伯特序列

$$S_{m,n} = \frac{\prod_{i=1}^{m}(1-z^{d_i})}{(1-z)^n},$$

的第一个非负项的索引。

4.6.2　最小秩攻击

针对 Rainbow 的最小秩攻击是要找到一个秩小于 v_2 的 Rainbow 公钥多项式相关矩阵的线性组合。在 Rainbow 中心映射中，第 i 层中心映射多项式相关矩阵的秩小于等于 v_{i+1}。其中第一层中心映射多项式相关矩阵的秩小于等于 v_2。由于仿射 R 作用到中心映射 G 上本质上只是对变量进行线性变换，并不会改变中心映射多项式相关矩阵的秩。存在一些公钥多项式相关矩阵的线性组合的秩小于等于 v_2，而这些矩阵的线性组合实际上对应的就是 Rainbow 中心映射的第一层多项式相关矩阵。攻击者通过最小秩攻击找到这样的线性组合，它就能够分离出 Rainbow 的第一层中心映射。剩余的

Rainbow 密钥可以通过类似的方法提取。针对 Rainbow 的最小秩攻击的复杂度大约为 $O(o_1 * q^{v_1+1})$。

要分析循环 Rainbow 的最小秩攻击复杂度，我们必须分析其中心映射多项式相关矩阵的秩。与普通 Rainbow 中心映射多项式相关矩阵不同的是，循环 Rainbow 中心映射多项式相关矩阵子矩阵 VO_{v_i+j}（$i \in [1,\cdots,t]$ $j \in [1,\cdots,o_i]$），之间存在旋转关系。虽然这并不会直接影响中心映射多项式相关矩阵的秩，这种旋转关系在进行线性组合的时候，可能导致线性组合所得矩阵的秩变低。但在这里并不会影响最小秩攻击的复杂度。因为在循环 Rainbow 第 i 层中心映射矩阵 A_{v_i+j} 的左上角存在一个醋醋项系数矩阵 VV_{v_i+j}（$i \in [1,\cdots,t]$ $j \in [1,\cdots,o_i]$）。所有的醋醋项系数矩阵都是随机选取的，其秩小于等于 v_i，他们之间并不存在任何关系。所以矩阵 A_{v_i+j} 的线性组合后醋醋项系数矩阵的位置仍然是一个完全随机的矩阵。所以这里即使存在秩的降低，但是从最小秩攻击的角度来讲，只能降低随机搜索方法的复杂度。随机搜索的复杂度大约 $O(\frac{n^2}{6} * q^{v_1})$，这并不比传统的针对 Rainbow 的最小秩攻击复杂度低 [85]。所以我们可以得出结论：循环 Rainbow 的最小秩攻击的复杂度为 $O(o_1 * q^{v_1+1})$。

4.6.3 高秩攻击

在高秩攻击中，攻击者想要找到被很多公钥多项式相关矩阵的线性组合所共享的小的核空间。这实际上和最小秩攻击的思路是刚好相反的。在最小秩攻击中，攻击者是想找到一个被少数公钥多项式相关矩阵线性组合所共享的大的核空间。高秩攻击对 Rainbow 有效是因为除了最后一层以外的 Rainbow 中心映射多项式相关矩阵的秩都不满秩。第 i 层中心映射多项式相关矩阵 A_{v_i+j}（$j \in [1,\ldots,o_i]$）的秩小于等于 v_{i+1}。中心映射最后一层油空间是除最后一层的多项式以外所有多项式相关矩阵的核空间。所以我们使用高秩攻击找到一个被很多公钥多项式相关矩阵的线性组合所共享的小的核空间的时候，相当于能够找到最后一层中心映射的油空间在 R^{-1} 作用下的像空间。对于普通 Rainbow，高秩攻击的复杂度为 $O(q^{o_t} * \frac{n^3}{6})$，其中 o_t 是 Rainbow 中心映射最后一层油变量个数。

对于循环 Rainbow，如果参数选择不当高秩攻击可能会变成一个很危险的攻击方法。为了研究循环 Rainbow 高秩攻击复杂度，我们需要研究清楚循环 Rainbow 最后一

层中心映射多项式相关矩阵的秩。我们首先定义 $o_t \cdot o_t$ 大小的旋转矩阵：

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & 1 & 0 \\ 0 & \ddots & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

为了能够更清楚的进行分析，我们先消除仿射 R 对中心映射的影响。对公钥多项式作用上 R^{-1} 仿射，等价于将矩阵 R^{-T} 和 R 作用到中心映射多项式相关矩阵 A_h 两边：

$$R^{-T} A_h R = \begin{bmatrix} E & H \\ H^T & 0 \end{bmatrix},$$

其中 E 是 $v_t \cdot v_t$ 大小的矩阵，H 是 $v_t \cdot v_o$ 大小的矩阵。如果 E 是一个可逆矩阵，那么矩阵 $R^{-T} A_h R$ 的秩为 $Rank(R^{-T} A_h R) = Rank(E) + Rank(H)$。在循环 Rainbow 中，$H$ 可以表示成 $H = VO_{v_t+1} \sum_{i=0}^{o_t-1} \lambda_i M^i$，其中 λ_i 是基域上的随机元素。因为矩阵 VO_{v_t+1} 是一个 $v_t \cdot o_t$ $(v_t > o_t)$ 的随机矩阵，所以 VO_{v_t+1} 以极大概率等于 o_t。那么 H 的秩主要取决于矩阵线性组合 $\sum_{i=0}^{o_t-1} \lambda_i M^i$ 的秩。矩阵 $\sum_{i=0}^{o_t-1} \lambda_i M^i$ 的秩的公式为

$$Rank(\sum_{i=0}^{o_t-1} \lambda_i M^i) = o_t - degree(gcd(x^{o_t} - 1, \sum_{i=o}^{o_t-1} \lambda_i x^i)).$$

一个随机的矩阵线性组合 $\sum_{i=0}^{o_t-1} \lambda_i M^i$ 的秩主要取决于 $x^{o_t} - 1$ 在基域上的因式分解。

假设 $x^{o_t} - 1$ 在基域上可以分解为 $x^{o_t} - 1 = \sum_{k=0}^{z} f_k$。令 d_k 为多项式 f_k 的度，且 $d_0 \leq d_1 \leq \cdots \leq d_z$。算法4-1描述了如何利用高秩攻击来求解循环 Rainbow 的 $R^{-1}(O_t)$ 的一组基。

由算法4-1可得：循环 Rainbow 的高秩攻击复杂度为

$$HighRank(q, n, o_t) = \frac{n^3}{6}(q^{d_0} + q^{d_1} + \cdots + q^{d_z}).$$

其主要参数 d_i 由多项式 $x^{o_t} - 1$ 在基域上的因式分解所决定。对于不同的 q 和 o，多项式 $x^{o_t} - 1$ 的因式分解差别很大。对于参数为 (GF(256),19,18,19) 的循环 Rainbow 签名方案，其高秩攻击复杂度大约为 2^{87}。但是如果我们稍微调整参数为 (GF(256),19,18,18)，其复杂度就只有 2^{41}。这正是 $x^{o_t} - 1$ 在基域 GF(256) 的因式分解所决定的。为了防止高秩攻击，我们在选择循环 Rainbow 参数的时候要额外小心。这导致循环 Rainbow 的参数选择不如普通 Rainbow 灵活。但总的来讲这并不会影响循环 Rainbow 的性能，因为高秩攻击目前并不是决定 Rainbow 参数选取的最主要因数。

算法 4-1: 针对循环 Rainbow 的高秩攻击

输入: 循环 Rainbow 的参数 (K,v_1,o_1,...,o_t) 和公钥多项式相关矩阵 A_1,...,A_m。

输出: $R^{-1}(O_t)$ 的一组基.

1: 在基域上将 $x^{o_t}-1$ 分解成 $x^{o_t}-1=\sum_{k=0}^{z} f_k$。令 d_k 为多项式 f_k 的度且 $d_0 \leq d_1 \leq \cdots \leq d_z$。初始化 $D=\{d_0,\ldots,d_z\}$，$B=\{\}$。

2: 从 D 中取第一个元素 d_k 并随机选取基域上元素 $\lambda'_{v_1+1},\ldots,\lambda'_n$。计算 $A_h=\sum_{i=v_1+1}^{n}\lambda'_i A_i$ 并找到 A_h 的核空间 $U=ker(A_h)$。

3: 如果和空间 U 的维度大于 0，构造方程组 $(\sum_{i=v_1+1}^{n}\lambda'_i A_i)U=0$。如果方程组解空间维度为 o_t-d_k，将空间 U 的基加入 B 中并从集合 D 中移除 d_k。如果 D 不为空集，则回到步骤 2

4: 返回 B.

4.6.4 彩虹带分离攻击

彩虹带分离攻击的核心思想是要利用 Rainbow 中心映射多项式相关矩阵中的零列来构造等式求解等价密钥。其可以被认为是 UOV 协调攻击的扩展版本。攻击者能够找到一个等价密钥 (S',R',G') 满足 $S\circ G\circ R=P=S'\circ G'\circ R'$ 且 G' 具有 Rainbow 中心映射结构。彩虹带分离攻击的主要复杂度在于恢复第一层好密钥时需要求解关于 n 个变量的一个三次多变量方程和 $m+n-2$ 个二次多变量方程。

在参数为 $(K,v_1,o_1,d*o'_2)$ 的 MB Rainbow 的中心映射多项式相关矩阵中，有更多的零列存在，攻击者在进行彩虹带分离攻击的时候能够得到更多的方程，从而降低了攻击的复杂度。在循环 Rainbow 中，中心映射多项式相关矩阵和普通 Rainbow 一样稠密。攻击者进行彩虹带分离攻击的时候无法定位到更多的零元素。也许有人会认为循环 Rainbow 中心映射多项式相关矩阵中的旋转关系可能会被攻击者在彩虹带分离中利用来获得更多的方程，因为在中心映射多项式相关矩阵中我们有 $OV_h[1,1]-OV_{h+1}[1,2]=0$。实际上这样的想法并不正确，因为彩虹带分离攻击中要找的等价密钥 G' 匹配的是特殊的 S' 和 R'，G' 实际上已经不存在旋转关系了。攻击者无法利用 G 中的旋转关系来确定 G' 的内容。所以循环 Rainbow 具有和普通 Rainbow 面对彩虹带分离攻击有相同的安全性。循环 Rainbow 的彩虹带分离攻击的复杂度等价于求解关于 n 个变量的一条三次多变量方程和 $m+n-2$ 条二次多变量方程。

4.6.5　UOV 攻击

UOV 攻击 [41] 的主要目的是要找到在 UOV 中心映射的油空间在 R^{-1} 作用下的像。针对 UOV 签名方案的 UOV 攻击的复杂度公式为 $O(q^{v-o-1} * o^4)$。在循环 Rainbow 中，仿射 S 将中心映射的所有的多项式相关矩阵进行了线性组合，所以循环 Rainbow 的每个公钥多项式可以被认为是有 v_t 个醋变量 o_t 个油变量的 UOV 公钥多项式。如果我们选择 $v_t \geq 2o_t$，循环 Rainbow 的 UOV 攻击复杂度是 o_t 的指数级别。

为了验证循环 Rainbow 的 UOV 攻击的复杂度公式，我们对循环 Rainbow 和 Rainbow 参数生成 100 个实例，然后对它们应用 UOV 攻击。表 4-2证明了循环 Rainbow 和普通 Rainbow 对应 UOV 攻击有相同的复杂度。根据文章 [37] 中的分析，我们可以得出循环 Rainbow 的 UOV 攻击的复杂度为 $O(q^{v_t-o_t-1} * o_t^4)$。

表 4-2 GF(256) 上循环 Rainbow 和普通 Rainbow 的 UOV 攻击耗时

参数 (v_1,o_1,o_2)	Rainbow	循环 Rainbow
(4,2,2)	0.03 s	0.03 s
(4,3,3)	0.41 s	0.40 s
(4,4,4)	1.97 s	1.98 s
(5,4,4)	409.41 s	405.96 s
(6,4,5)	947.65 s	943.40 s

4.6.6　UOV 协调攻击

UOV 协调攻击最初是用来攻击 UOV 签名方案的。在 UOV 中，中心映射多项式相关矩阵的油油项系数都为 0，UOV 协调攻击利用这个特性来产生 v 个变量的 o 个二次方程。因为 Rainbow 公钥系统可以看成是一个 v_t 个醋变量 o_t 个油变量的 UOV 公钥系统，所以 UOV 协调攻击同样适用于循环 Rainbow 和 Rainbow。在 UOV 协调攻击中，攻击者是要找到一个等价密钥 (S', R', G') 满足 $S \circ G \circ R = P = S' \circ G' \circ R'$ 且 G' 具有 UOV 陷门形式。如第 4.6.4节一样，G 中的旋转关系并不能帮助攻击者获得更多的方程，因为等价密钥 G' 中已经不存在旋转关系了。所以我们可以得出循环 Rainbow 的 UOV 协调攻击复杂度不低于求解关于 v_t 个变量的 o_t 个多变量二次方程。

4.7 循环 Rainbow 的性能

本章中，我们将讨论循环 Rainbow 参数选择和性能，并将其与其他 Rainbow 变种方案进行对比。

4.7.1 安全参数选择和安全性对比

从第 4.1.1节我们可以知道，参数为 $(K, v_1, o_1, d * o_2')$ 的 MB Rainbow 容易被彩虹带分离攻击攻破因为其最后一层中心映射多项式相关矩阵有更多的零列。攻击者能够得到更多的方程从而降低了求解复杂度。幸运的是，这种变种彩虹带分析攻击并没有完全攻破 MB 方法。Rainbow 中心映射除了最后一层之外，其他层中心映射仍然可以使用 MB 结构。例如，我可以使用参数为 $(K, v_1, d * o_1', o_2)$ 的 MB Rainbow 来防止第 4.1.1节的攻击。

为了更进一步理解循环 Rainbow 和其他 Rainbow 变种的安全性，我们在表 4-3和表 4-4中给出基域为 GF(256) 的循环 Rainbow 和其他 Rainbow 变种在不同攻击下的复杂度。

表 4-3 GF(256) 上循环 Rainbow 和其他 Rainbow 变种方案的攻击复杂度

攻击方法	普通 Rainbow (19,18,19)	MB Rainbow (19,2*9,19)	NT Rainbow (19,18,19)	循环 Rainbow (19,18,19)
直接攻击	2^{113}	2^{113}	2^{113}	2^{113}
高秩攻击	2^{166}	2^{166}	2^{166}	2^{87}
彩虹带分离攻击	2^{80}	2^{80}	2^{80}	2^{80}
小秩攻击	2^{164}	2^{164}	2^{164}	2^{164}
UOV 攻击	2^{152}	2^{152}	2^{152}	2^{152}
UOV 协调攻击	2^{145}	2^{145}	2^{145}	2^{145}
安全级别 (bit)	80	80	80	80

从表 4-3和表 4-4中，我们可以看出循环 Rainbow 的高秩攻击复杂度比普通 Rainbow 要低，这导致循环 Rainbow 的参数选择没有普通 Rainbow 那么灵活。但是这对循环 Rainbow 的性能影响并不大，因为目前决定 Rainbow 和其变种的主要攻击方法是彩虹带分离攻击。对于循环 Rainbow、NT Rainbow 和普通 Rainbow，我们可以选择参数 (GF(256),19,18,19) 和 (GF(256),26,23,23) 来达到 2^{80} 和 2^{100} 安全级别。为了满足 MB

表 4-4 GF(256) 上循环 Rainbow 和其他 Rainbow 变种方案的攻击复杂度

攻击方法	普通 Rainbow (26,23,23)	MB Rainbow (26,2*12,23)	NT Rainbow (26,23,23)	循环 Rainbow (26,23,23)
直接攻击	2^{144}	2^{148}	2^{144}	2^{144}
高秩攻击	2^{199}	2^{199}	2^{199}	2^{104}
彩虹带分离攻击	2^{104}	2^{104}	2^{104}	2^{104}
小秩攻击	2^{220}	2^{220}	2^{220}	2^{220}
UOV 攻击	2^{218}	2^{226}	2^{218}	2^{218}
UOV 协调攻击	2^{186}	2^{190}	2^{186}	2^{186}
安全级别 (bit)	100	100	100	100

Rainbow 的构造要求，我们适当调整参数至 (GF(256),19,2*9,19) 和 (GF(256),26,2*12,23)。

4.7.2　与其他 Rainbow 变种对比

本节中我们通过实验对比来证明循环 Rainbow 的性能优势。我们首先将循环 Rainbow 与其他 Rainbow 变种方案进行对比，然后我们再将循环 Rainbow 与目前主流的多款数字签名方案进行对比。本节中所有的实验都是在 Intel Core i7-4790@3.6Ghz CPU 上运行，并使用 AVX2 指令集对多项式求值和矩阵向量乘法进行加速。在选择了合适的参数后，我们对循环 Rainbow 和其他 Rainbow 变种进行了 C++ 代码的实现。并记录了他们平均签名生成时间和验证时间在表 4-5中。

从表 4-5中我们可以看出，循环 Rainbow 在签名上大约是普通 Rainbow 速度的 3 倍并能够降低私钥 45% 的大小。在所有的 Rainbow 变种方案中，循环 Rainbow 是签名速度最快、密钥最小的。

4.7.3　与其他签名方案对比

为了进一步证明循环 Rainbow 的高效性，我们将循环 Rainbow 与目前主流的数字签名方案如 RSA，ECDSA 等进行对比。对于 RSA 和 ECDSA，我们选择 OpenSSL[86] 中的实现。在 OpenSSL 中，RSA 和 ECDSA 的实现已经运用了 SSE 等 x64 CPU 上 SIMD 指令进行加速。我们根据 NIST 密钥管理推荐手册 [87] 来为 ECDSA 和 RSA 选择合适安全参数，并将 OpenSSL 的 x64 版本在本机上运行以获得更加公平的对比。同时我们将第三章中的循环 UOV 和 Gui 签名方案加入对比。对于 Gui 方案，我们直接选取文章

表 4-5 循环 Rainbow 和其他 Rainbow 变种对比

安全性	方案	参数 (K,v_1,o_1,o_2)	公钥 (kB)	私钥 (kB)	签名 (bit)	签名周期 (10^3 cycles)	验证周期 (10^3 cycles)
2^{80}	Rainbow	(GF(256),19,18,19)	59.7	42.0	448	616.7	20.0
	MB Rainbow	(GF(256),19,2*9,19)	59.7	34.2	448	523.0	20.0
	NT Rainbow	(GF(256),19,18,19)	59.7	26.5	448	334.2	20.0
	循环 Rainbow	(GF(256),19,18,19)	59.7	23.0	448	210.1	20.0
2^{100}	Rainbow	(GF(256),26,23,23)	121.3	84.7	576	966.8	65.2
	MB Rainbow	(GF(256),26,2*12,23)	127.5	77.2	584	816.3	68.2
	NT Rainbow	(GF(256),26,23,23)	121.3	50.8	576	669.7	65.2
	循环	(GF(256),26,23,23)	121.3	46.1	576	327.8	65.2

[24] 中的实验数据。表4-6给出了几个数字签名方案的性能对比。

表 4-6 循环 Rainbow 和其他数字签名方案对比

方案	参数	安全级别 (bit)	公钥 (kB)	私钥 (kB)	签名 (bit)	签名周期 (10^3 cycles)	验证周期 (10^3 cycles)
Gui	(95,5,6,6)	80	61.6	3.1	128	238	62
RSA	1024	80	0.13	0.13	1024	475	54
ECDSA	nistk163	80	0.04	0.02	326	720	1140
循环 UOV	(GF(31),65,34)	80	101.7	53.9	495	252	43
循环 Rainbow	(GF(256),19,18,19)	80	57.9	42.0	448	210	33

从表 4-5中我们可以看出，循环 Rainbow 在签名速度和验证速度上相对于其他方案都具有较为明显的优势。相对于号称目前最快的 MPKC 数字签名方案的 Gui。循环 Rainbow 和循环 UOV 具有更快的签名速度，同时他们具有十分相近的签名速度。相对于传统的数字签名方案，MPKC 数字签名方案明显具有更快的签名生成和验证速度。但其公私钥大小却也存在明显的劣势。考虑到在一些运用场景中，密钥不需要频繁更换，这样的结果在 MPKC 领域中已经相对可以令人接受。总而言之，我们的实验结果证明循环 UOV 和循环 Rainbow 相对于其他数字签名方案具有很强的竞争力。

4.8　小结

本章中，我们将循环 UOV 扩展为循环 Rainbow 方案。循环 Rainbow 相比较于循环 UOV 拥有更好的性能的同时也拥有更短的密钥和签名。但其参数选择和安全性分析比循环 UOV 要求更严格。

安全性上，我们对所有针对 Rainbow 的攻击方法都进行了理论分析，并用实验验证了理论分析的结论。同时我们对比分析了其他 Rainbow 变种方案的安全性，并选择了合适的安全参数。

在性能上，实验显示循环 Rainbow 在相同安全级别上，相比较于其他 Rainbow 变种方案具有更快的签名速度和更小的密钥。

第五章　一种新型的 SRP 变种加密方案

前面两章分别介绍了两种基于循环矩阵的多变量的签名方案，本章我们介绍一种新型的多变量加密方案。第一个备受关注的多变量公钥密码方案 C* [12] 实际上就是一个加密方案，但是 Patarin 在文章 [13] 中提出了针对 C* 的线性方程攻击。从此之后，许多多变量加密方案如 HFE [20]，STS [88]，ZHEF [25]，ABC [31]，SRP [32] 等被提出。但是大多数多变量加密方案都已经被攻破，目前鲜有安全的多变量加密方案。SRP 是目前少有的安全的多变量加密方案。

SRP 方案本质上是 Square 加密方案、Rainbow 签名方案和加方法的一个组合性方案。其本质上是对 Square 方案在安全性和性能上的一个修补方案。Square 方案最由 Ding 等人先在文章 [32] 中被提出。但随后 Billet 等人在文章 [61] 中提出了 Square 加密系统的差分攻击，在多项式时间内能够攻破 Square 的实用参数。为了修复该问题，Ding 等人在 Square 中心映射中加入了加方法，即在 Square 中心映射中加入随机二次多变量多项式，从而扰动其公钥系统的差分性质，抵抗差分攻击。该方案被称为 Square+ 方案。不幸的是，Square+ 方案很容易被最小秩攻击攻破。双层 Square 方案的提出，能够在一定程度上降低最小秩攻击对 Square 方案的威胁，但这要求增大参数 l 的取值。SRP 方案从密钥结构上可以看成是 Square+ 方案和双层 Square 方案的结合，它可以抵抗针对 Square 的差分攻击和最小秩攻击。同时其公钥大小只有 ABC 加密方案的 1/30，解密速度接近于 ZHFE。不幸的是，SRP 签名的密钥大小还是太大。Petzoldt 等人将 Cyclic 法运用到 SRP 中，以降低 SRP 的公钥大小并提高其加密速度。在本章中，我们通过除去 SRP 方案中冗余多项式来降低 SRP 公私钥大小，并提高其加解密速度。同时我们将前面两章的旋转关系引入到 SRP 方案中得到循环 SRP 加密方案，进一步提高循环 SRP 的解密速度。

5.1　优化 SRP 方案

SRP 加密方案本质上是 Square 加密方案、Rainbow 签名方案和加方法的一个组合性方案。通过将 Square 加密方案和 Rainbow 签名方案合并到一起，一些针对单个方案的攻击方法不再奏效。因为 Square 加密方案和 Rainbow 签名方案效率都很高，所以其组合方案也具有很高的效率。但 SRP 方案在设计上任存在缺陷，本节中我们分析 SRP 方案中存在缺陷，并对其进行优化。

5.1.1　r 条冗余油醋多项式

在 SPR 方案解密过程中，中心映射的求逆可以被分成两个部分：Square 部分和 Rainbow 部分。Square 方案的求逆需要在大域上计算 $X^{q^d+1/4}$，其中 $(q \equiv 3 \bmod 4, d \equiv 1 \bmod 2)$。

针对 Rainbow 部分求逆的提速。很自然的我们可以考虑将第三章和第四章中的循环 UOV 和循环 Rainbow 运用在这里，提高 SRP 的解密速度并降低其私钥大小。

但是这里循环矩阵的构造看上去并不能很好的满足 SRP 方案中 Rainbow 部分的结构。因为在 SRP 方案中为了提高解密成功率，中心映射的 Rainbow 部分相对于普通 Rainbow 方案多出了 r 条额外的油醋多项式。这样导致 Rainbow 求逆过程中生成的线性方程组 $L\mathbf{x} = \mathbf{u}$ 中的矩阵 L 并不是一个方阵。一个比较常规的想法是利用引入旋转关系的方法，将每层需要求解的系统变成一个超定 Topelitz 线性系统 [89]，然后利用变种 Bareiss 算法 [90] 的快速求解超定 Toeplitz 系统。从而提高解密效率。事实上，我们最初对 SRP 进行改进的设计就是想利用 Toeplitz 结构进行。但经过分析后，我们发现 SRP 有更大的改进空间。

在 SRP 中心映射中 r 条多余油醋多项式的引入，是为了提高解密正确率。我们将 Square 部分求解出来后会有两个醋变量备选值。如果将 r 条冗余油醋多项式删除，我们将这两个备选值带入到 Rainbow 部分求解的时候，会面临两个问题。

1) 正确解判定问题：因为有两个初始醋向量待验证，我们需要先后将两个备选醋变量带入 Rainbow 部分。我们必须在系统中引入足够多的冗余，来帮助确定正确的初始向量，从而正确求逆中心映射。因此 SRP 原作者们会引入 r 条冗余的油醋多项式到中心映射中，使其成为超定系统。

2) L 矩阵不可逆问题（线性方程无唯一解）：在进行 Rainbow 部分求逆的时候。SRP 方案与传统的 Rainbow 签名方案并不一样。在传统 Rainbow 签名方案中，我们随机选择醋变量带入系统中求解，当遇到 L 矩阵不可逆的时候，我们可以重新选择一个初始醋变量进行求解。而在 SRP 中，初始醋变量是只能从两个备选值里面选。在求解的时候，当我们遇到 L 矩阵不可逆的情况时，就必须求解部分二次方程。这将极大影响其解密效率。

事实上，Plus 部分的随机二次多变量多项式在求逆过程中可以负责对问题一进行辅助判断。但要利用 Plus 部分解决问正确解判定问题，我们需要引入的随机多项式条数

l 满足 $q^l > 2^{80}$。这将极大的增加 SRP 的公钥大小。此外，在原本的 SRP 解密中，私钥拥有者实际上并不需要存储 Plus 部分。Plus 部分的引入主要是针对差分攻击和秩攻击进行的扰乱，在解密的时候 SRP 的原设计是将 Plus 部分直接丢弃掉的。

r 条冗余多项式的引入，会增大中心映射 Rainbow 部分求解线性方程的维度，也会增大了 SRP 加密方案的公私钥大小，严重影响 SRP 方案性能。我们必须解决上面两个问题，尽量减小 r。

5.1.2 解决办法

我们注意到在 SRP 方案中，左边仿射 R 实际上是一个 $K^n \to K^{n+l}$ 的扩张映射。合法用户实际上可以将中心映射的求逆和左边仿射 R 的求逆一起进行，这样可以帮助我们在中心映射求逆中引入一些线性约束。在常见的 SRP 参数中，通常有 $l = 16$。即引入了 16 条线性约束到我们待求解的系统中。因为 $31^{16} \approx 2^{80}$，这些约束可以大致帮助我们解决正确解判定问题和 L 矩阵不可逆问题。这样我们就可以将 SRP 中的 r 条冗余油醋多项式删掉，从而降低 SRP 公私钥大小，提高 SRP 加解密速度。本节中我们将细致地介绍我们解决办法。

为了简单起见，本章中我们提出的 SRP 方案的 Rainbow 部分都是单层 Rainbow 结构（UOV）。这是因为 UOV 结构能够放便的进行分析，也容易将 R 仿射融合到 UOV 的求逆运算中来。令 **m** 是加密的明文消息。**v** 为中心映射的醋变量向量，**o** 为中心映射的油变量向量。$L\mathbf{o} = \mathbf{y}$ 为醋变量带入到油醋后获得线性方程系统。这里我们假设中心映射的冗余油醋多项式数目为 $r = 0$。则 L 矩阵为 $o \cdot o$ 方阵，**y** 向量为 o 维向量。我们有以下线性关系（这里为了表达简约我们只考虑仿射 R 的线性项，加入常数项目后线性关系同样存在）

$$\begin{bmatrix} \mathbf{v} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & L \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v} \\ \mathbf{o} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & L \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}$$

由此我们可以得到一个关于向量 **v**，**y** 和明文消息向量 **x** 之间的一个简单的关系：

$$\begin{bmatrix} \mathbf{v} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}$$

即超定线性方程 $(\mathbf{v}, \mathbf{y})^T = A \cdot \mathbf{x}^T$。

对于有限域 K 上一个随机的 $m \cdot n$ 矩阵（$m > n$）其为满秩的概率为:

$$\prod_{i=0}^{n-1}(1-q^i/q^m)$$

在我们的参数中，矩阵 A 为 $(n+l)\cdot n$ 矩阵，故其为满秩（秩为 n）的概率为

$$\prod_{i=0}^{n-1}(1-q^i/q^{n+l})$$

因此，对于正确的 $(\mathbf{v},\mathbf{y})$，我们以极大概率可以使用简单线性代数计算出对应的正确解 $\mathbf{x}$。

对于非正确的 $(\mathbf{v},\mathbf{y})$ 的情况，由于矩阵 A 可认为是随机的矩阵。我们可认为错误的解向量有解 $\mathbf{x}$ 需要满足额外 l 条线性约束，其满足的概率大约为 q^{-l}。故通过该线性约束，我们可以以大约 $1-q^{-l}$ 的概率确定出唯一正确解。

5.1.3　性能对比

我们优化后的 SRP 方案与原 SRP 方案相比减少了中心映射中 r 条冗余多项式。这能降低公私钥的大小，同时提高加解密速度。这里我们将优化后的 SPR 方案与未优化的 SRP 方案进行一个简单的对比，并记录在表5-1中。

表 5-1 优化后的 SPR 方案与未优化的 SRP 方案性能对比

方案	参数 q,d,o,r,s,l	m,n	安全性	公钥 (kB)	私钥 (kB)	加密周期 (10^3 cycles)	解密周期 (10^3 cycles)
SRP	31,33,32, 16,5,16	86, 49	80	66.9	55.9	35	1873
优化后 SRP	31,33,32, 0,5,16	70, 49	80	54.5	37.9	28	1351

表5-1中两个方案的解密错误概率大约都是 2^{-80}。其中优化后的 SRP 方案比未优化的 SRP 公钥大小降低 18%，私钥大小降低 31%。加密速度大约为原方案的 1.27 倍，解密速度大约为原方案的 1.39 倍。但总的来说，优化后的 SRP 方案在解密速度和密钥大小方面还是不够理想。本章后面我们将使用循环矩阵的方法，进一步降低 SRP 方案的私钥大小，提高其解密速度。

5.2　循环 SRP

本章中我们将循环矩阵的方法引入到优化后的 SRP 加密方案中以进一步降低私钥大小，提高解密速度。

5.2.1 基本思想

我们在 SRP 中心映射的油醋多项式中引入部分旋转关系，从而让矩阵 L 成为循环矩阵。

这里我们给出循环 SRP 中心映射油醋多项式的相关矩阵。

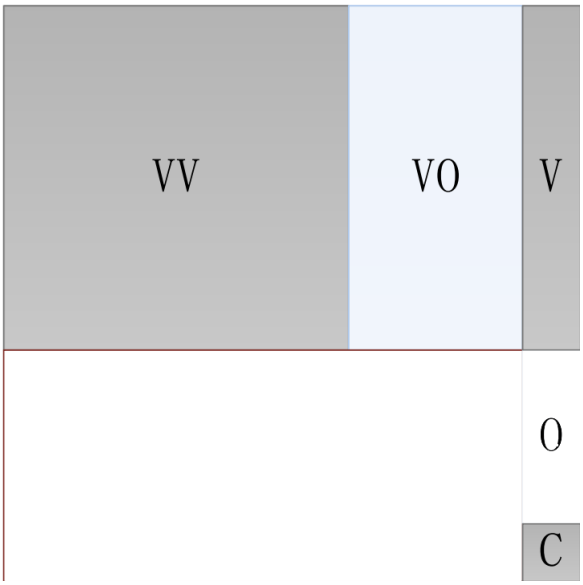

图 5-1 循环 SRP 中心映射油醋多项式的相关矩阵

VV 表示醋醋项系数，VO 表示油醋交叉项系数，V 表示醋变量线性项系数，C 是常数项。其中白色表示该部分为零元素，灰色表示该部分可以取基域上的任意元素，蓝色表示该部分和其他中心映射多项式的矩阵具有旋转关系。从中心映射矩阵表示图上来看，循环 SRP 中心映射中的油醋多项式和循环 UOV 中的油醋多项式都在 VO 部分引入了旋转关系。但不同的是，循环 UOV 中，油变量线性项的系数也存在旋转关系。而在循环 SRP 中，油醋多项式的油变量线性系数被我们置为零。这是为了进一步提高中心映射的求逆速度。线性项系数的缺失并不会对多变量公钥密码的安全性造成影响。

UOV 部分中心映射，第一条油醋多项式是完全随机选择的。换句话说，该多项式的相关矩阵所有的非零元素都是完全随机的。具体如下：

$$VV_1 = (\mathbf{vv}_1, \mathbf{vv}_2, \cdots, \mathbf{vv}_v)$$

$$VO_1 = (\mathbf{vo}_1, \mathbf{vo}_2, \cdots, \mathbf{vo}_o)$$

$$V_1 = v_{1,1}, \cdots, v_{1,v}$$

$$O_1 = \mathbf{0}$$

$$C_1 = c_1$$

对于剩下的 $o-1$ 条油醋多项式，我们首先随机选择 VV_i，V_i 和 C_i $(i=2,\cdots,o)$。然后令：

$$VO_2 = (\mathbf{vo}_o, \mathbf{vo}_1, \cdots, \mathbf{vo}_{o-1})$$

$$VO_3 = (\mathbf{vo}_{o-1}, \mathbf{vo}_o, \cdots, \mathbf{vo}_{o-2})$$

$$\vdots$$

$$VO_o = (\mathbf{vo}_2, \mathbf{vo}_3, \cdots, \mathbf{vo}_1)$$

和

$$O_k = \mathbf{0} \quad (i=2,\cdots,o).$$

5.2.2　中心映射求逆

循环 SRP 中心映射求逆可分为两个部分：Square 部分求逆和 UOV 部分求逆。这里我们讨论的中心映射求逆实际上是对 $G \circ R : K^n \to K^{d+o+s}$ 进行求逆。因为仿射 R 能够在求逆中心映射的过程中给予很多线性约束，我们要将其利用起来。

令待求逆向量值为 $\mathbf{b} = (\mathbf{s}, \mathbf{r}, \mathbf{p}) \in K^{d+o+s}$，其中 $\mathbf{s} \in K^d$，$\mathbf{r} \in K^o$，$\mathbf{p} \in K^s$。我们按以下步骤对循环 SRP 中心映射进行求逆。

首先利用 Square 部分对 **s** 进行求逆。

Square 部分求逆： 作用 $\phi^{-1}(s_1,\cdots,s_d) \to B : K^d \to E$ 同构映射将 **s** 映射到 K 的 d 次扩域 E 上。然后在扩域上计算 $Z_{1,2} = \pm B^{\frac{q^d+1}{4}}$。令 $\mathbf{v^i} = (v_1^i, ..., v_d^i) = \phi(Z_i)(i=1,2)$。我们获得两个醋变量向量的可能解 $\mathbf{v}^1$ 和 $\mathbf{v}^2$，且 $\mathbf{v}^1$ 和 $\mathbf{v}^2$ 刚好是正负关系。

Square 部分求逆的主要复杂度在于计算 $Z_{1,2} = \pm B^{\frac{q^d+1}{4}}$，其中 $q \equiv 3 \bmod 4$ 且 $d \equiv 1 \bmod 2$。这种指数运算的最常见方法是使用平方乘算法 (Square and Multiply algorithm)。但是这里实际上存在更简单的计算方法。由于 $\frac{q^d+1}{4}$ 存在 q-adic 扩张：

$$\begin{aligned}\frac{q^d+1}{4} = \frac{q+1}{4} + \frac{3q-1}{4}\cdot q + \frac{q-3}{4}\cdot q^2 + \frac{3q-1}{4}\cdot q^3 + \frac{q-3}{4}\cdot q^4 \\ + \frac{3q-1}{4}\cdot q^5 + \frac{q-3}{4}\cdot q^6 + \cdots + \frac{3q-1}{4}\cdot q^{d-2} + \frac{q-3}{4}\cdot q^{d-1},\end{aligned}$$

由于计算有限域上任何元素的 Frobenius 映射的开销非常小。所以我们可以使用多指数技术快速计算 $B^{\frac{q^d+1}{4}}$。算法5-1 给出其具体步骤。

UOV 部分求逆： 这里 UOV 部分的求逆，我们同时引入了仿射 R 中的线性约束。将 $\mathbf{v}^1$ 其作为醋变量 **v** 带入到 UOV 部分进行求逆。我们能得到一个关于 o 个变量 o 条方

算法 5-1: Square 中心映射快速求逆

输入: 扩域 E 上元素 B。

输出: $B^{\frac{q^d+1}{4}}$ 的值。

1: 计算二元扩张 $(d-1)/2 = [b_k, b_{k-1}, \cdots, b_0]_2$。

2: $\beta = 1, m = 0, \epsilon = B^q$。

3: 初始化 $\epsilon = \epsilon^{q^{2^i}+1}$。对于 $i = 0, \cdots, k$，计算 $\epsilon = \epsilon^{q^{2^i}+1}$，如果 $b_i = 1$，计算 $\beta = \beta \cdot \epsilon^{q^m}$，$m = m + 2^{i+1}$。

4: $\gamma = B_0 \cdot \beta^{q+1}$。

5: $\beta = \beta^{\frac{q+1}{2}}$，$\gamma = \gamma^{\frac{q-3}{4}}$。

6: 返回 $\beta = \beta \cdot \gamma \cdot B$。

程的线性方程组。

$$\underbrace{\mathbf{v}^T \cdot VV_i \cdot \mathbf{v} + \mathbf{v}^T \cdot V_i + C_i}_{\text{常数}} + \underbrace{\mathbf{v}^T \cdot VO_i \cdot \mathbf{o} + O_i \cdot \mathbf{o}}_{\text{线性}} = b_{d+i} (i = 1, \cdots, o)$$

令

$$u_i = r_i - (\mathbf{v}^T \cdot VV_i \cdot \mathbf{v} + \mathbf{v}^T \cdot V_i + C_i),$$

我们有

$$\underbrace{\begin{bmatrix} \mathbf{v}^T \cdot VO_1 \\ \mathbf{v}^T \cdot VO_2 \\ \vdots \\ \mathbf{v}^T \cdot VO_o \end{bmatrix}}_{L} \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_o \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_o \end{bmatrix}.$$

结合前面的旋转关系，我们可以很容易看出 L 是一个 $o \cdot o$ 的循环矩阵：

$$\begin{bmatrix} V^T \cdot \mathbf{vo}_1 & V^T \cdot \mathbf{vo}_2 & \cdots & V^T \cdot \mathbf{vo}_o \\ V^T \cdot \mathbf{vo}_o & V^T \cdot \mathbf{vo}_1 & \cdots & V^T \cdot \mathbf{vo}_{o-1} \\ \vdots & \vdots & \ddots & \vdots \\ V^T \cdot \mathbf{vo}_2 & V^T \cdot \mathbf{vo}_3 & \cdots & V^T \cdot \mathbf{vo}_1 \end{bmatrix}.$$

由于 L 是一个 $o \cdot o$ 的循环矩阵，所以我们只需要计算出 L 矩阵的第一行的 o 个元素就可以完全计算出 L。剩下的元素可以通过位移操作获得。这在计算上可以节省大量的时间。然后我们可以用扩展欧几里得算法对该线性系统进行求解，找出油变量的值。

这里值得注意的是，因为我们令所有的油变量线性系数为 0，所以 L 矩阵的每一

行都是 $\mathbf{v}^T \cdot VO_i$ 的值。那么如果 $\mathbf{v} = \mathbf{v}^2 = -\mathbf{v}^1$ 的时候，解得的线性方程组的 L 矩阵与 $\mathbf{v} = \mathbf{v}^1$ 时候是相同的。此时我们令

$$w_i = r_i - (\mathbf{v}^T \cdot VV_i \cdot \mathbf{v} + \mathbf{v}^T \cdot V_i + C_i),$$

可以看出，$\mathbf{w}$ 与 $\mathbf{v}$ 只有 $\mathbf{v}^T \cdot V_i$ 部分的值不一样。在计算上我们重复使用一些中间值，这可以极大降低复杂度。当 $\mathbf{v} = \mathbf{v}^2$ 时，我们能获得关于 o 个油变量的线性方程:

$$\underbrace{\begin{bmatrix} \mathbf{v}^T \cdot VO_1 \\ \mathbf{v}^T \cdot VO_2 \\ \vdots \\ \mathbf{v}^T \cdot VO_o \end{bmatrix}}_{L} \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_o \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_o \end{bmatrix}.$$

使用扩展欧几里得法求解线性系统 $L\mathbf{o} = \mathbf{u}$ 和 $L\mathbf{o} = \mathbf{w}$，然后我们利用 R 仿射的线性约束确定正确的解。由于两个线性系统 L 矩阵都相等，所以 L 矩阵的求逆运算我们可以只计算一次，这可以提高 UOV 解密速度。但这里有一种特殊情况，即当矩阵 L 不可逆时，我们就无法直接对两个线性系统进行求解了。这个时候我们可以计算出线性方程组

$$\begin{bmatrix} \mathbf{v}^1 \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{v}^2 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}$$

直接求解出向量 $\mathbf{x}$。

以下是映射 $G \circ R$ 的具体求逆步骤。

步骤 1：　作用 $\phi^{-1}(s_1, \cdots, s_d) \to B$。然后计算 $Z_{1,2} = \pm B^{\frac{q^d+1}{4}}$。令 $\mathbf{v^i} = \phi(Z_i)\ (i = 1, 2)$。

步骤 2：　将 $\mathbf{v}^1$ 和 $\mathbf{v}^2$ 带入到油醋多项式中获得两个线性系统 $L\mathbf{o} = \mathbf{w}$ 和 $L\mathbf{o} = \mathbf{u}$ 两个线性系统。

步骤 3：　写出循环矩阵 L 和向量 $\mathbf{u}, \mathbf{w}$ 的的相关多项式 $l(x)$、$u(x)$ 和 $w(x)$。使用扩展欧几里得法，求解 $l(x)$ 在环 $K[x]/(x^o - 1)$ 上的逆元 $l^{-1}(x)$。如果该逆元不存在，则直接求解方程组

$$\begin{bmatrix} \mathbf{v}^1 \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{v}^2 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}$$

得到逆元 $\mathbf{x}$。

步骤 4：　计算 $u(x) * l^{-1}(x)$、$w(x) * l^{-1}(x)$ 的值，得到两个油变量的备选解 $\mathbf{o}^1$ 和 $\mathbf{o}^2$。

步骤 5：　计算 $(\mathbf{v}^1, \mathbf{o}^1)$ 和 $(\mathbf{v}^2, \mathbf{o}^2)$ 对于仿射 R 的前像（此时可排除一个错误解），得到逆

元 $\mathbf{x}$。

5.2.3 解密成功率

在 SRP 方案的解密过程中，由于任何合法密文都是由明文直接计算出来的，所以针对一个合法的密文，中心映射一定存在至少一个逆元。所以在 SRP 中，对解密成功率的关注，主要在于如何解决多解问题，即如何确定唯一正确解。

回顾整个 SRP 方案解密过程。S 仿射是可逆单对单映射，其求逆过程不会出现多解问题。R 仿射是扩张单射，求解过程也不会存在多解问题。中心映射部分，Plus 部分在解密中会被直接丢掉。Square 部分中，我们要计算 $Z_{1,2} = \pm B^{\frac{q^d+1}{4}}$，这里会产生 $\mathbf{v}^1$ 和 $\mathbf{v}^2$ 两个醋变量候选值。在 UOV 部分中，我们需要求解线性方程组 $L\mathbf{o} = \mathbf{w}$ 和 $L\mathbf{o} = \mathbf{w}$。而这里矩阵 L 存在不可逆的情况，当 L 不可逆时，线性方程组也存在多解的情况。所以 SRP 中解密成功率主要需要关注如何消除 Square 部分和 UOV 部分求逆的多解问题。即两个线性方程组

$$\begin{bmatrix} \mathbf{v}^1 \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{v}^2 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \end{bmatrix}$$

的多解问题。

令

$$A = \begin{bmatrix} R_1 \\ L \cdot R_2 \end{bmatrix},$$

为这两个线性方程组的伴随矩阵。A 为 $(n+l) \cdot n$ 维度矩阵，R_1 为 $v \cdot n$ 维矩阵，$L \cdot R_2$ 为 $o \cdot n$ 维矩阵。这两个方程中，一定存在一个线性方程有解。

因此，对于正确的 $(\mathbf{v}, \mathbf{y})$，我们以极大概率可以使用简单线性代数计算出对应的正确解 $\mathbf{x}$。此时我们仅需要 A 矩阵的秩大于 n。由于 R_1 的秩为满秩 v（R_1 为我们任选的），所以仅需要 $L \cdot R_2$ 的行空间能够覆盖到剩下的 $n - v$ 维的行空间，该概率大于 $1 - q^{-l}$，几乎确定发生。

对于非正确的 $(\mathbf{v}, \mathbf{y})$ 的情况，由于矩阵 A 可认为是随机的矩阵。我们可认为错误的解向量有解 $\mathbf{x}$ 需要满足额外 l 条线性约束，其满足的概率大约为 q^{-l}。故通过该线性约束，我们可以以大约 $1 - q^{-l}$ 的概率确定出唯一正确解。

5.2.4 密钥大小

相对于比普通 SRP 方案，我们的循环 SRP 方案的 UOV 只需要存储下面的内容：$VV_i (i = 1, \cdots, o), V_i (i = 1, \cdots, o), C_i (i = 1, \cdots, o)$。

因此，循环 SRP 的私钥一共包含

$$m \cdot (m+1) + (n+l) \cdot (n+1) + (o \cdot (\frac{v \cdot (v+1)}{2} + 2v + 1)$$

个基域元素。其公钥则一共包含 $m \cdot \frac{(n+1) \cdot (n+2)}{2}$ 个基域元素。

5.3　循环 SRP 的一般性描述

这里我们给出循环 SRP 加密方案的一般性描述。令 K 为一个奇特征域且满足 $q \equiv 3 \bmod 4$。令 d 为一个奇数，扩域 E=F_{q^d}。令 $\phi : E \to K^d$ 是一个 E 与向量空间 K^d 的同构映射，o，s，l 为非负整数。SRP 方案的一般性描述如下。

密钥生成： 令 $n = d + o - l$，$n' = d + o$ 且 $m = d + o + s$，中心映射 G 是三个影射 G_S，G_R 和 G_P 的组合。公钥则为 $P = S \circ G \circ R$，其中 $S : K^m \to K^m$ 和 $R : K^n \to K^{n'}$ 是随机满秩线性仿射，G_S，G_R 和 G_P 构造如下：

1) Square 部分 $G_S : K^{n'} \to K^d$ 的构造：

$$K^{d+o} \xrightarrow{\pi_d} K^d \xrightarrow{\phi^{-1}} E \xrightarrow{X \to X^2} E \xrightarrow{\phi} K^D.$$

其中 $\pi_d : K^{d+o} \to K^d$ 为取前 d 个坐标值的投影映射。

2) UOV（Rainbow）部分: 生成一个参数为 $(K, v = d, o)$ 的循环 UOV 中心映射 $G_R = (g_1, ..., g_o) : K^{n'} \to K^o$。

3) 加映射部分 $G_P = (p_1, ..., p_s) : K^{n'} \to K^s$ 是由 s 个随机多项式 $p_1, ..., p_s$ 组成的。

4) 随机生成满秩仿射 $R : K^n \to K^{n'}$ 和可逆仿射 $S : K^{n'} \to K^{n'}$，计算公钥映射 $P = S \circ (G_S || G_R || G_P) \circ R$。

加密： 给定消息 $\mathbf{m} \in K^n$，计算密文 $\mathbf{c} = P(\mathbf{m})$。

解密： 给定密文 $\mathbf{c} = (c_1, ..., c_m)$，解密步骤如下：

1) 求逆 S 仿射: $(\mathbf{s}, \mathbf{r}, \mathbf{p}) = \mathbf{b} = (b_1, ..., b_m) = S^{-1}(\mathbf{c})$。

2) 计算 $B = \phi^{-1}(s_1, ..., s_d)$。

3) 计算 $Z_{1,2} = \pm B^{\frac{q^d+1}{4}}$，令 $\mathbf{v}^i = (v_1^i, ..., v_d^i) = \phi(Z_i)(i = 1, 2)$。

4) 将 $\mathbf{v}^1$ 和 $\mathbf{v}^2$ 带入到油醋多项式中获得两个线性系统 $L\mathbf{o} = w$ 和 $L\mathbf{o} = \mathbf{u}$ 两个线性系统。

5) 写出循环矩阵 L 和向量 $\mathbf{u}, \mathbf{w}$ 的的相关多项式 $l(x)$、$u(x)$ 和 $w(x)$。使用扩展欧几里得法，求解 $l(x)$ 在环 $K[x]/(x^o - 1)$ 上的逆元 $l^{-1}(x)$。如果该逆元不存在，则

直接求解方程组

$$\begin{bmatrix}\mathbf{v}^1\\ \mathbf{u}\end{bmatrix}=\begin{bmatrix}R_1\\ L\cdot R_2\end{bmatrix}\cdot\begin{bmatrix}\mathbf{x}\end{bmatrix},\qquad \begin{bmatrix}\mathbf{v}^2\\ \mathbf{w}\end{bmatrix}=\begin{bmatrix}R_1\\ L\cdot R_2\end{bmatrix}\cdot\begin{bmatrix}\mathbf{x}\end{bmatrix}$$

得到明文消息 $\mathbf{x}$

步骤 4： 计算 $u(x)*l^{-1}(x)$、$w(x)*l^{-1}(x)$ 的值，得到两个油变量的备选解 $\mathbf{o}^1$ 和 $\mathbf{o}^2$。

步骤 5： 计算 $(\mathbf{v}^1,\mathbf{o}^1)$ 和 $(\mathbf{v}^2,\mathbf{o}^2)$ 对于仿射 R 的前像（此时可排除一个错误解），得到明文消息 $\mathbf{m}$。

5.4 安全性分析

在具体介分析循环 SRP 安全性，我们先简单介绍 Square 系统的安全性。Square 系统最先在文章 [32] 中被提出，随后 Billet 等人在文章 [61] 提出了针对 Square 加密系统和 Square-Vinegar 签名系统的差分攻击。此后 Ding 等人在文章 [91] 中提出了 Square+ 系统和双层 Square 系统（Double-layer Square System），以抵抗差分攻击。Square+ 系统是通过向 Square 中心映射中添加随机二次多变量多项式以扰乱 Square 公钥系统的差分性质，抵抗差分攻击。双层 Square 则是在 Square 中心映射中引入双层结构，让中心映射更为复杂，从而抵抗差分攻击。不幸的是，Thomae 等人在随后在文章 [62] 中提出了对双层 Square 和 Square+ 方案的最小秩攻击法。其中针对 Square+ 系统的攻击是变种的 KS 攻击 [21]，而针对双层 Square 系统的攻击是针对特定结构的最小秩攻击。针对 Square+ 的 KS 攻击复杂度大约为扩域 E 上 $\binom{n+l+s}{2}^3$ 个操作，其中 p 为在 Square 方案中心映射中加入的随机二次多变量多项式的条数。由此可见，简单的将加方法作用到 Square 系统上并不能获得具有很好安全性的方案。针对双层 Square 系统的最小秩攻击的复杂度大约为小域 K 上 $(n+l)q^{l+1}(2n+l)^3$ 个操作，其中 $n+l$ 为第一层 Square 中心映射的输入变量数，$2n+l$ 为第二层复杂 Square 的输入变量数。从复杂度公式可以看出，想要保证双层 Square 方案的小秩攻击安全性，我们必须将 l 增大。但由于 Square 加密系统公钥本身是一个超定系统，增大 l 大小可能会在一定程度上降低其公钥系统抵抗代数攻击的能力。

SRP 加密方案，实际上可以看成是双层 Square 方案的一个变种方案。目前并没有有效的攻击方法能够对其进行攻击。在 Duong 等人在文章 [92] 中将 Cyclic 方法运用到 SRP 方案中，从而降低了其加密速度和公钥大小。本文我们通过除去冗余多项式的，降低 SRP 方案公私钥大小、密文长度，并提高其加解密性能。同时我们将前面两章中使

用到的循环矩阵方法运用到 SRP 中构造循环 SRP 方案，以获得更小的密钥和更快的解密速度。总的来讲 SRP 和循环 SRP 有很多相似之处。但在 SRP 原文的文章中，对其安全性分析并不详细。本节我们将主流的攻击方法运用到循环 SRP 方案上，并从理论和实验两个角度进行安全性分析。

5.4.1　直接攻击

攻击循环 SRP 最直接的方案就是直接求解关于密文和公钥的二次方程，得到明文消息。常见的直接攻击方法包括 F4/F5 算法、XL 算法 [93] 和庄子算法 [94] 等。因为循环 SRP 方案公钥系统本身是一个超定二次多变量方程组，所以针对循环 SRP 的公钥参数最有效的直接攻击方法是直接使用 F4/F5 算法来计算 Gröbner 基。

在进行具体实验分析之前，我们可以先大致描述循环 SRP 和 SRP 在面对直接攻击中的区别。由于 SRP 方案和循环 SRP 均为超定系统。如果我将 SRP 中心映射的 r 条冗余多项式去掉，从直接攻击的角度来看，攻击者能获得的明文密文之间的关系降低了。所以 r 条冗余多项式的去除实际上增强了 SRP 方案的直接攻击安全性。此外，使用循环 UOV 代替 Rainbow 或 UOV 并不会影响 SRP 方案的直接攻击安全性。因为我们第3.5.1中已经分析得出循环 UOV 具有和普通 UOV 一样的抗直接攻击性质。为了验证我们的猜想，我们对具有相同参数的 SRP 方案和循环 SRP 方案进行直接攻击（区别在于循环 SRP 中 $r = 0$）。我们在 Magma 上进行了一系列 F4 攻击实验。并将其实验结果记录在表5-2中。

表 5-2　Magma 上 F4 直接攻击实验结果

q, d, o, r, s, l	SRP	循环 SRP($r = 0$)
31, 11, 10, 6, 5, 6	0.839 s	1.13 s
31, 11, 10, 6, 5, 4	37.765 s	42.127 s
31, 11, 10, 6, 5, 3	105.864 s	130.826 s
31, 11, 11, 6, 5, 6	2.333 s	3.391 s
31, 13, 10, 6, 5, 6	27.238 s	41.312 s

从表5-2中我们可以看出，在参数相同的情况下，循环 SRP 比 SRP 方案具有更高的直接攻击复杂度。

随后我们将循环 SRP 公钥系统与随机二次多变量多项式系统进行直接攻击对比。

实验结果记录在表5-3中。

表 5-3 Magma 上 F4 直接攻击实验结果

q, d, o, r, s, l	m, n	循环 SRP 公钥系统	随机系统
31, 11, 10, 0, 5, 6	26, 15	0.839 s	0.843 s
31, 11, 10, 0, 5, 4	26, 17	37.765 s	37.127 s
31, 11, 10, 0, 5, 3	26, 18	105.864 s	104.826 s
31, 11, 11, 0, 5, 6	27, 16	2.333 s	2.391 s
31, 13, 10, 0, 5, 6	28, 17	27.238 s	27.312 s

由表5-3可以看出，循环 SRP 在 F4 算法下，与随机的系统表现相近。所以我们可以认为循环 SRP 和普通随机二次多变量多项式具有几乎相同的正则度。循环 SRP 的正则度 d_{reg} 等于满足多项式 $\frac{(1-z^2)^m}{(1-z)^n}$ 的 z^D 项的系数小于或等于 0 最小的次数 D 的值。故循环 SRP 的直接攻击复杂度公式为：

$$O(m \cdot \binom{m-k+d_{reg}+1}{d_{reg}}^{\omega}).$$

5.4.2 线性方程攻击

线性方程攻击在多变量公钥密码中最初被用来攻击 C* 方案。它的基本思路是通过分析中心映射的特殊结构，来得到一个关于公钥系统的输入和输出的线性关系:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}x_i y_j + \sum_{i=1}^{n} b_i x_i + \sum_{j=1}^{m} c_j y_j + d$$

其中 $\{\mathrm{y}_1, ..., \mathrm{y}_m\}$ 是 m 条公钥多项式在变量 $(\mathrm{x}_1, ..., \mathrm{x}_n)$ 上的值。当攻击者能够产生足够多的明密文对时，它就能直接求解出这个线性关系中的未知系数，从而更有效的求解公钥系统。

在循环 SRP 方案中，中心映射的 UOV 部分和 Plus 部分是完全免疫线性方程攻击的。Square 部分中，我们有 Square 映射 $X \to X^2$，这实际上是 C* 方案（$X \to X^{q^\theta+1}$）的一个特殊实例 ($\theta = 0$)。但实际上 C* 方案中的线性方程攻击对 Square 是完全无效的。因为 C* 中的线性关系 $XY^{q^\theta} - X^{q^{2\theta}}Y = 0$ 在 Square 中变成了平凡关系 $XY = XY$。所以 Square 部分的线性方程的空间的维度为 0。因为 Square 部分、UOV 部分、Plus 部分均不存在线性关系，所以理论上循环 SRP 明密文之间也不存在双线性关系。实验是检验线性关系是否存在的最佳方法。我们随机产生足够多的循环 SRP 明密文对并带入到

线性方程攻击的模型中进行求解，实验结果显示明文与密文之间并不存在明显的线性关系。

对于高阶线性攻击 [95]，攻击者是想找到一个让明密文对满足的方程组

$$\sum_{i_0=1}^{n}\sum_{i_1,\cdots,i_s=1}^{m} u_{i_0,\cdots,i_s}x_{i_0}\cdot y_{i_1}\cdots y_{i_s}+\sum_{i_0=1}^{n}\sum_{i_1,\cdots,i_{s-1}=1}^{m} v_{i_0,\cdots,i_{s-1}}x_{i_0}\cdot y_{i_1}\cdots y_{i_{s-1}}+ \cdots+\sum_{i_0=1}^{n} b_{i_0}x_{i_0}+\sum_{i_1=1}^{m} c_{i_1}y_{i_1}+d=0$$

假设一个多变量公钥系统的明密文之间存在 s 次高阶线性关系。那么这个高阶线性关系中待求解的系数数量为

$$n\sum_{j=0}^{s}\binom{m}{j}+m+1=n\binom{m+s}{s}+m+1.$$

攻击者使用公钥系统产生足够多的明密文对。通过将这些明密文对带入到方程中，我们能得到一个关于 $n\binom{m+s}{s}+m+1$ 个变量的 $n\binom{m+s}{s}+m+1$ 个线性方程。求解该方程的计算复杂度为

$$n(\binom{m+s}{s}+m+1)^w,$$

其中 w 为线性代数因子常数。在密码分析中，常令 $w=2$。循环 SRP 方案中的 Square 部分、UOV 部分和 Plus 部分均不存在明显的低次高阶线性关系，所以高阶线性攻击很难对循环 SRP 方案构成威胁。为了保证循环 SRP 方案抵抗高阶线性攻击，我们仅需证明该攻击复杂度足够大便可。所以我们可以通过实验的方式对小规模的循环 SRP 方案内进行低次高阶线性攻击实验，确定其是否存在高阶线性攻击。由于 Rainbow 和随机二次多项式均不存在低次高阶线性方程攻击，所以我们只需要对 Square 系统进行分析。本文中，我们对参数为 (GF(31),$d=19$) 的 Square 系统进行高阶线性关系搜索。实验结果显示其并不存在低于 7 次的高阶线性关系。

Rainbow 部分和 Plus 部分的扰动，也会让高阶线性攻击更加难以对 SRP 进行攻击。因此，合适参数的循环 SRP 公钥系统能够抵抗线性方程攻击和高阶线性方程攻击。

5.4.3　差分攻击

在分析循环 SRP 差分性质的时候，我们首先介绍针对 Square 方案的差分攻击。

给定公钥系统 P，差分方程的定义如下：

$$DP(\mathbf{a},\mathbf{x})=P(\mathbf{x}+\mathbf{a})-P(\mathbf{x})-P(\mathbf{a})+P(\mathbf{0}).$$

针对 Square 方案，其公钥系统 P 的差分表示为

$$DP(\mathbf{x},\mathbf{y}) = S(2\cdot R(\mathbf{x})\cdot R(\mathbf{y})).$$

在 Square 中，$R: K^n \to K^{n+l}$ 是一个扩张映射。为了简单起见，我们假设 R 是一个可逆的 $K^{n+l} \to K^{n+l}$ 仿射。随机选取 $\mathbf{y_1}$ 和 $\mathbf{y_2}$，令

$$D_1 = DP(\mathbf{x},\mathbf{y_1}) = S(2\cdot R(\mathbf{x})\cdot R(\mathbf{y_1})), D_2 = DP(\mathbf{x},\mathbf{y_2}) = S(2\cdot R(\mathbf{x})\cdot R(\mathbf{y_2})),$$

实际上我们有 $D_1 = S\circ M_{\lambda_1}\circ R$ 和 $D_2 = S\circ M_{\lambda_2}\circ R$。其中 M_{λ_i} 表示乘以扩域上元素 λ_i。我们直接计算：

$$L_0 = D_2\circ D_1^{-1}$$

很显然，L_0 具有 $L_0 = S\circ M_{\lambda_2\lambda_1^{-1}}\circ S^{-1}$ 这样的特殊格式。当得到一个特定格式的 $S\circ M_{\lambda_2\lambda_1^{-1}}\circ S^{-1}$ 后，我们可以通过对 L_0 特征多项式的分析而得到扩域上 $\lambda_2\lambda_1^{-1}$ 的值。从而计算出一个 S_0 映射的等价表示，并恢复出所有密钥。

实际在 Square 方案中，差分攻击并没有这么简单。主要原因是 $R: K^n \to K^{n+l}$ 映射是一个扩张映射。攻击者无法像普通差分攻击一样自由的调整实际中心映射的输入。这个时候，我们想要找到一个满足：

$$L\circ D_1 = D_2$$

的映射 L。$S\circ M_{\lambda_2\lambda_1^{-1}}\circ S^{-1}$ 就是我们想要求解的目标。但由于 R 是非可逆映射的原因，直接求解得到的 L 并不一定都是 $S\circ M_{\lambda}\circ S^{-1}$ 形式的。Billet 等人提出通过引入更多约束从而解出一个具备 $S\circ M_{\lambda}\circ S^{-1}$ 形式的映射 L，然后利用 L 恢复出全部公钥。对于 Square 加密方案的第一组推荐参数，该攻击方法攻击复杂度仅为 2^{39}。

为了抵抗差分攻击，Ding 等人提出了 Square+ 方案。其本质就是将 Plus 方法引入到 Square 中心映射中。通过中心映射中引入的随机二次多变量多项式扰乱公钥的差分性质。实际上，在 PMI 加密方案被差分攻击攻破后，加方法就被用来扰乱 PMI 公钥的差分性质，该改进方案被称为 PMI+ 加密方案。PMI+ 加密方案提出十多年来，一直没有被攻破。加方法的引入，可以很有效的帮助 Square 方案抵抗差分攻击。但其同时也会引入额外的安全隐患和性能损失。加方法的引入，意味着公钥系统更大，加密时间更长。同时加方法的引入，意味着攻击者能够获得更多的关于明密文之间的关系。这可能会降低直接攻击的复杂度。但就本文循环 SRP 参数而言，这并不会造成明显安全问题。

现在我们回到循环 SRP 方案上。在循环 SRP 方案中，UOV 部分和 Plus 部分的引入实际上都相当于对 Square 中心映射进行了加方法操作。由于 UOV 部分在增大多项式

数量的同时也会增大方程的变量数，所以 UOV 部分的引入对性能和安全性实际上不会造成影响。Plus 部分我们的最终取值也比较小，并不会对性能和安全性造成太明显的影响。攻击者想要利用 Square 中心映射的差分性质攻击 SRP 方案，就必须提前分离出 SRP 的 Square 部分和其他部分。如果攻击者随机的进行分离，实现分离的概率大约为 $q^{-s(o+l)}$。这显然并不是一个高效的方法。另外一个分离的方法是使用小秩攻击进行分离，这实际上需要攻击者完成最小秩攻击。所以循环 SRP 差分攻击的复杂度会大于最小秩攻击。关于循环 SRP 小秩攻击的安全性我们会在下一小节进行介绍。

5.4.4　最小秩攻击

小秩攻击是要将公钥多项式的分析问题转换为最小秩问题求解。我们要找到公钥多项式相关矩阵的一组线性组合的秩小于等于 r。在多变量方案中，我们通常会遇到两种最小秩攻击。一种是针对小域方案的最小秩攻击，如第3.5.5节中针对 UOV 的最小秩攻击。另外一种针对大域方案的最小秩攻击，通常被称为 KS 攻击。KS 攻击首先被 Kipnis 和 Shamir 提出并利用攻击 HFE 加密方案 [21]。实际上奇特征的 HFE 加密方案与 Square 方案具有很强的相似性，所以在文章 [62] 中，Thomae 等人针对 Square+ 方案提出了变种 KS 攻击。其复杂度大约为扩域上 $\binom{n+l+s}{2}^3$ 个基础运算。

幸运的是，我们的 SRP 方案虽然看上去和 Square+ 方案很像。但是实际上中心映射结构上还是有很大差别的。SRP 方案的中心映射结构从最小秩攻击的角度来上，与双层 Square 方案很相似。双层 Square 方案的最小秩攻击复杂度为小域上 $(n+l)q^{l+1}(2n+l)^3$ 个基础运算。在 SRP 的最初文章中，作者直接使用了该结论作为 SRP 方案最小秩攻击的复杂度。实际上这样直接套用是不对的，这里我们使用类似方法分析循环 SRP 小秩攻击复杂度。在循环 SRP 方案中，中心映射相关矩阵具有图5-2中几种形式：

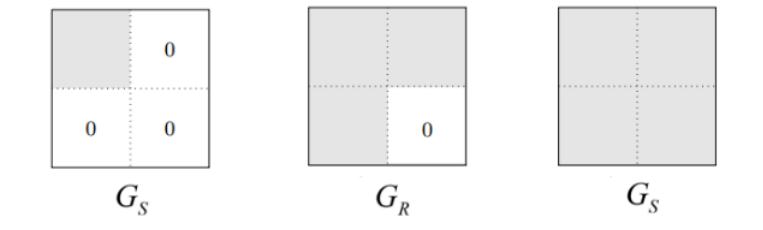

图 5-2　循环 SRP 中心映射多项式的相关矩阵

解决该最小秩问题的思路是：随机选择一个向量 $\mathbf{w} \in K^n$，并且希望它落到低秩矩阵的线性组合的核空间中。这样的话，我们就通过求解线性方程线性系统：

$$\sum_{i=1}^{n+l+s} \lambda_i \hat{P}_i \mathbf{w} = 0$$

来得到映射 S 的部分秘密值。但是该攻击方法的复杂度主要取决于一个随机抽样的 $\mathbf{w}$ 能够落到低秩矩阵的线性组合的核空间中的概率。假设 R 是一个可逆映射 $K^{n+l} \to K^{n+l}$，$\mathbf{w} \in K^{n+l}$。这个概率实际上相当于

$$R\mathbf{w} \in Ker(\sum_{i=1}^{d} \lambda_i \hat{G}_i).$$

对于随机固定的 $\mathbf{w} \in K^{n+l}$，线性系统

$$(\sum_{i=1}^{d} \lambda_i \hat{G}_{Si} + \sum_{i=1}^{o} \lambda_i \hat{G}_{Ri} + \sum_{i=1}^{s} \lambda_i \hat{G}_{Pi}) R\mathbf{w} = 0$$

存在线性组合系数 $\lambda_1, \cdots, \lambda_d \in K$ 使 $R\mathbf{w} \in Ker(\sum_{i=1}^{d} \lambda_i \hat{G}_{Si})$ 成立的概率等于图5-3中 A 矩阵不满秩的概率 $(1/q)$。图5-3实际为

$$R\mathbf{w} \in Ker(\sum_{i=1}^{d} \lambda_i \hat{G}_i)$$

的系数矩阵。

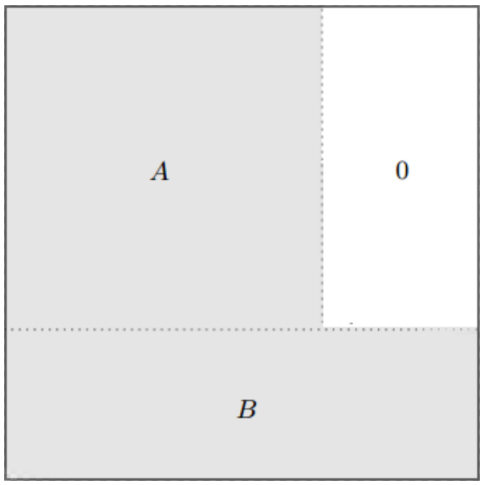

图 5-3 系数矩阵

其中 A 矩阵为 $d \cdot d$ 方阵，A 矩阵右边的空白为 $d \cdot o$ 零矩阵。对于 K 上一个随机 $d \cdot d$ 矩阵，其不满秩的大约为 $1/q$。

为了能够求出一个线性组合，我们同时还要求图5-3中矩阵的秩等于 $Rank(A) + o$。在我们的参数选择情况下，发生的概率十分接近于 1。我们可认为绝大多数时候都满足该条件。综上所其最小秩攻击的复杂度大约为:

$$qd(n + l + s)^3,$$

因为我们大约需要重复抽样 q 次才能使 A 变成奇异矩阵，且我们需要恢复 d 个线性无关的小秩组合。其中 $(n + l + s)^3$ 为求解线性方程复杂度。

现在我们才开始考虑 R 映射的线性部分是非方阵的情况。这种情况下，我们有

$$\sum_{i=1}^{d} \lambda_i \hat{G_{Si}} R\mathbf{w} + \sum_{i=1}^{o} \lambda_i \hat{G_{Ri}} R\mathbf{w} = 0 + \sum_{i=1}^{s} \lambda_i \hat{G_{Pi}} R\mathbf{w} = 0,$$

相当于

$$\sum_{i=1}^{d} \lambda_i R^T \hat{G_{Si}} R\mathbf{w} + \sum_{i=1}^{o} \lambda_i R^T \hat{G_{Ri}} R\mathbf{w} + \sum_{i=1}^{s} \lambda_i R^T \hat{G_{Pi}} R\mathbf{w} = 0.$$

这个时候，随机选择一个 $\mathbf{w}$ 落在低秩矩阵的核空间的概率依然是 $1/q$。但是 $R\mathbf{w}$ 已经不再是 K^{n+l} 上的随机元素了，所以图5-3中的行不再是随机的选择的了。因为 $n+l+s$ 个长度为 $n+s$ 的向量总是线性相关的，所以我们会得到 q^l 个解。所以针对循环 SRP 方案的小秩攻击的复杂度大约为

$$q^{l+1} d (n+l+s)^3.$$

5.5　循环 SRP 的性能

本节中我们通过实验对比来证明循环 SRP 的性能优势。我们首先将循环 SRP 与其他 SRP 进行对比，然后我们再将循环 SRP 与目前主流的多款公钥加密方案进行对比。本节中所有的实验都是在 Intel Core i7-4790@3.6Ghz CPU 上运行，并使用 AVX2 指令集对多项式求值和矩阵向量乘法进行加速。

5.5.1　与 SRP 加密方案对比

我们将循环 SRP 与 SRP 方案在密钥大小、解密性能上进行了实验对比。我们选择文章 [32] 的参数来实现 2^{80} 和 2^{112} 安全级别，并使用 C++ 代码实现进行性能测试。对每组参数进行 1000 次实验并记录平均数据到表 5-4中。可以从该表中看出，我们的循环 SRP 方案相比较于 SRP 方案在解密性能上具有很大的提高。在 2^{112} 安全级别下，公钥大小可以降低 18%，私钥大小可降低 70%。加密速度大约为原方案的 1.27 倍，解密速度大约为原方案的 3.5 倍。

5.5.2　与其他加密方案对比

为了进一步证明循环 SRP 的高效性，我们将循环 SRP 与一些目前主流的公钥加密方案如 RSA，Ring-LWE 等进行对比。同时我们也把目前已知最快的多变量加密方案实现 [96] 引入到对比中。对于 Ring-LWE，我们选择目前最快的格密码加密方案实现 [97] 的数据。对于 RSA 加密方案，我们直接使用 OpenSSL 中的实现在本机进行测速。表5-5给出了具体的实验对比数据。

由表5-5可以看出，循环 SRP 在加密速度上相对于其他方案具有明显优势。但是解

表 5-4 循环 SRP 与 SRP 性能对比

方案	参数 q,d,o,r,s,l	m,n	安全性	公钥 (kB)	私钥 (kB)	加密周期 (10^3 cycles)	解密周期 (10^3 cycles)
SRP	31,33,32, 16,5,16	86, 49	80	69.9	55.9	35	1873
SRP	31,47,47, 22,5,22	121, 72	112	200.0	157.7	95	5483
循环 SRP	31,33,32, 0,5,16	70, 49	80	54.5	17.3	28	466
循环 SRP	31,47,47, 0,5,22	99, 72	112	163.2	45.3	75	1557

表 5-5 循环 SRP 与其他公钥加密方案对比

方案	参数	安全性	公钥 (kB)	私钥 (kB)	加密周期 (10^3 cycles)	解密周期 (10^3 cycles)	密文扩张
SMES	SMES49	80	472.8	64.2	74.7	85.8	2
Ring-LWE	RING-LWE256	80	0.8	0.8	121.2	43.3	26
RSA	1024	80	0.13	0.13	54	475	1
循环 SRP	31,33,32, 0,5,16	80	54.5	37.9	28	466	1.76

密速度和公私钥大小方面仍然存在不足。[97] 中 Ring-LWE 的实现虽然看上去在公私钥、加解密速度上都有很好的效果。但实际上为了保证该高效的解密速度，存储开销其实是相当大，只不过并没有体现在私钥大小中。同时其密文扩张也很大。总的来说，各个方案实际上均存在优缺点，并没有某个方案具有压倒性优势。循环 SRP 方案在特定场景下，具有很强竞争力。

5.6 小结

本章中，我们提出了一种约减 SRP 加密方案中的冗余油醋多项式的方法，并将循环矩阵的方法引入到 SRP 中心映射中得到了循环 SRP 加密方案。我们对循环 SRP 的解密正确性、安全性进行了详细的分析。我们可能威胁到 SRP 加密方案安全性的攻击方法运用到循环 SRP 上分析其安全性，并提出其与 SRP 加密方案所相对应的安全参数。

与 SRP 加密方案相比，循环 SRP 加密方案具有更快的加解密速度、更短的公私钥和密文长度。我们的实验结果显示，循环 SRP 能降低大约 70% 的私钥长度和 18% 的公钥长度。其加密速度大约为 SRP 的 1.27 倍，解密速度大约为 SRP 的 3.5 倍。这样的优势令循环 SRP 方案更加适用于在资源受限的环境中使用。同时我们将循环 SRP 方案与常见的公钥加密方案进行对比，实验结果显示循环 SRP 具有相当强的竞争力。

第六章　适用于无线传感网络的在线离线循环 UOV 签名方案

无线传感器网络 (Wireless Sensor Networks, WSN) 是一种分布式传感网络 [98]。它的末端是可以感知和检查外部世界的传感器。由于其低廉的成本和广泛的适用性，它被广泛使用于商业和工业应用中。在某些 WSN 应用中，传送数据的正确性至关重要。比如，在病患状态监测系统中，病人关键生理信息如果被篡改将会造成无法挽回的损失。

然而，由于无线通讯的特殊性，WSN 相比于传统有线网络更容易受到攻击 [99]。不幸的是，无线传感器网络设备通常只有十分有限的能量、计算力和存储空间。传统的数字签名方案如 ECDSA、RSA 等方案因为能耗太高、签名延迟太长等原因都不太适用于无线传感网络 [100]。设计一个轻量级的、节能的适用于无线传适用感网络的数字签名方案目前是一个十分有价值的研究方向。

6.1　能量收集技术

能量收集技术（Energy Harvesting Technologies）的发展 [101]，为降低 WSN 设备的能耗提供了新的机遇。拥有能量收集功能的无线传感器设备能够从环境中收集能量（风能、太阳能等），并将其转换成可用的电能存储在电容器中。其收集能量的大小取决于周围的具体环境，十分不稳定。以太阳能为例，一般在中午时太阳辐射比较强能量收集设备所收集到的电能也比较多。当收集能量超出电容器的最大容量时，我们称发生了一次能量峰。如果不马上使用，能量峰发生时溢出的能量就会被浪费了。在夜晚的时候，设备无法接收到太阳的辐射能量，因此无法产生电能。

预计算技术可以用来应对这样的能量收集不均的情况。在一个可进行预计算的签名方案中，我们可以将一次签名计算分成在线阶段和离线阶段 [102]。离线阶段独立于要签名的消息，可以提前进行计算。在线阶段取决于要进行签名的消息，只能在知道消息之后才能计算。当能量峰发生时，我们可以进行大量预计算（离线阶段）并存储其结果到存储器中。当收到具体要签名的消息的时候，再从存储器中取出预计算的结果并完成剩下的签名计算（在线阶段）。预计算的方法能够降低数字签名方案运行时的延迟和能耗。

在文章 [103] 中，作者们提出了 AGREE 方案。AGREE 方案结合能量收集技术和预计算技术在无线传感网络节点上实现基于属性的加密方案，从而降低了基于属性的加

密方案在无线传感器网络节点上的运行时延迟和能耗。经过该工作的启发，结合能量收集技术和预计算技术的方法在无线传感网络上实现数字签名方案的工作开始出现 (如 ECDSA [104][105]，基于哈希的签名方案 [106] 和基于格的签名方案 [107])。

多变量签名方案（UOV 和 Rainbow）被认为是无线传感网络的一个很好候选方案，因为它对计算性能要求较低。目前已经有很多 UOV 签名方案在低功耗设备上的实现 [108][109][110]，但目前没有人将能量收集技术和预计算技术相结合应用到 UOV 上。在文章 [107] 中，作者认为多变量公钥密码方案很难进行构造在线离线签名方案。因为在“求逆”公钥系统 $P = S \circ G \circ R$ 的过程中，需要逐步求逆。而进行求逆的第一步，对仿射 S 进行求逆便取决于消息 **m**，在不知道消息 **m** 的情况下，很难完成多变量密码方案的预计算。在本章中，我们将提出一个适用于具备能量收集功能的无线传感网络的 UOV 在线离线签名方案。

6.2　预计算技术

WSN 设备一般都拥有较长的休眠期。当一个任务来到的时候才中断休眠进行计算，计算完成之后继续进入休眠状态。所以其工作模式很适合将一次签名生成过程分成在线阶段和离线阶段。离线阶段的工作可以在任务请求来到之前的休眠时间内完成，它包括了所有与具体待签名消息无关的计算。在线阶段的计算只包含与消息相关的操作，这样能极大的提高无线传感节点的反应速度。不过实际上预计算的方法其实很少在无线传感网络中使用，因为预计算相比较于普通计算方法往往需要消耗更多的能量和存储空间。

能量收集技术的发展，为预计算在无线传感网络中应用提供了新的机会。在不同的环境中，能量收集器所收集到的能量具有很大的方差。图6-1给出了使用 IXOLAR XOB17-04x3 太阳能电池板收集能量到 1F Maxwell HC Powerseries 电容器的能量图。

图中红色的区域表示能量峰。能量峰发生时溢出的能量如果不及时使用掉，就会被浪费了。一个比较直观的想法是，在能量峰发生时进行签名生成的预计算，同时保存计算结果到存储器中。当获得具体待消息签名时，再利用预计算结果来进行签名。这样预计算技术可以充分利用能量峰的溢出能量，降低签名生成的总的能量开销和运行时延迟。

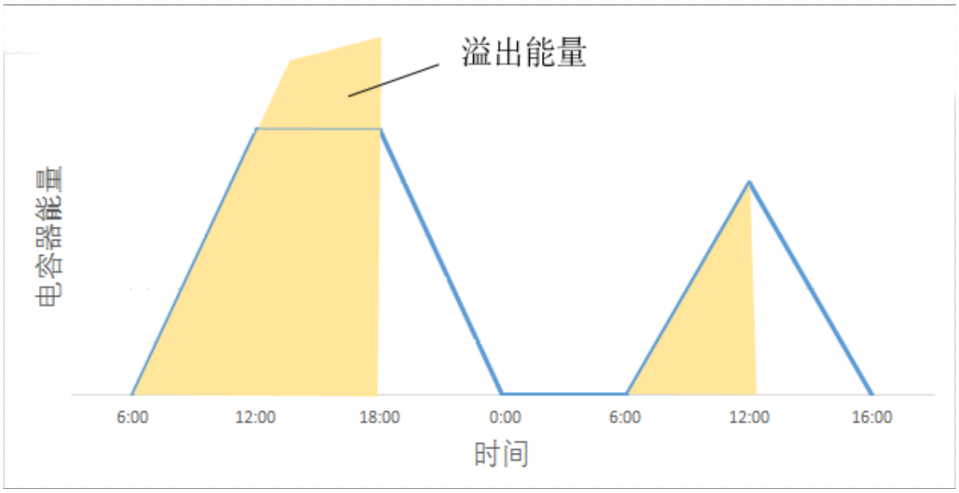

图 6-1 电容器能量图

6.3 在线离线 UOV

在本节中，我们将提出一个为具有能量收集功能的无线传感网络节点（Energy-Harvesting Wireless Sensor Network，EH-WSN）[111] 所设计的在线离线 UOV 签名方案。

6.3.1 基本方案

在 UOV 签名过程中，开销最大的是中心映射的求逆过程。我们需要选取一个随机的醋向量并将其带入到中心映射系统中以获得一个线性系统 $L\mathbf{o} = \mathbf{u}$，然后使用高斯消元求解线性方程。这里我们给出将 UOV 中心映射的求逆过程分为在线和离线阶段的方法。

首先我们回顾一下 UOV 中心映射多项式相关矩阵表示和其签名生成具体过程。图6-2是 UOV 中心映射的多项式矩阵表示，我们这里我们保留常数项和线性项。

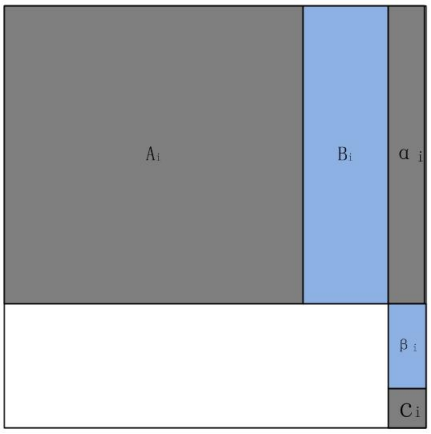

图 6-2 UOV 中心映射多项式的矩阵表示

假设待求逆的值为 **m**。我们随机选择醋变量 **v** 并带入到$(x_1, \cdots, x_v)$ 中。对于每条

中心映射多项式g_k 我们能得到方程

$$\underbrace{\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k}_{\text{常数项}} + \underbrace{\mathbf{v}^T * B_k * \mathbf{o} + \beta_{\mathbf{k}} \cdot \mathbf{o}}_{\text{线性项}} = m_k.$$

向量**o**=$(o_1, \cdots, o_o)$ 代表油变量$(x_{v+1}, \cdots, x_{v+o})$。令 $y_k = (\mathbf{v}^T * A_k * \mathbf{v} + \mathbf{v}^T \cdot \alpha_k + c_k)$ $(k \in [1, \cdots, o])$，我们能得到线性系统 $L\mathbf{o} = \mathbf{u}$：

$$\underbrace{\begin{pmatrix} \mathbf{v}^T * B_1 + \beta_1 \\ \mathbf{v}^T * B_2 + \beta_2 \\ \vdots \\ \mathbf{v}^T * B_{o-1} + \beta_{o-1} \\ \mathbf{v}^T * B_o + \beta_o \end{pmatrix}}_{\text{L}} \begin{pmatrix} o_1 \\ o_2 \\ \vdots \\ o_{o-1} \\ o_o \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{o-1} \\ m_o \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{o-1} \\ y_o \end{pmatrix}.$$

在这个线性系统中，矩阵 L 和向量 **y** 实际上都与消息 **m** 无关。所以我们可以在知道消息之前先计算好它们。这样我们就可以将 UOV 签名生成过程分成离线和在线阶段，并对在线阶段进行预计算。下面我们给出 UOV 签名生成过程的在线离线阶段划分。

离线阶段：

随机选取醋变量$v_1, .., v_v \in (K^v)$。用 $(v_1, \cdots, v_v)$ 代替$(x'_1, \cdots, x'_v)$ 带入中心映射多项式中进行计算，我们将获得一个 $o \cdot o$ 维矩阵 L 和向量 o 维向量 **y**。计算 L 的逆矩阵 L^{-1}。如果 L 不可逆, 重新选择一个醋变量 **v**，直到 L 矩阵可逆。将得到的预计算结果 $(\mathbf{v}, \mathbf{y}, L^{-1})$ 存储到存储器中。

在线阶段：

当收到待签名的消息 **m** 后，计算 $\mathbf{u} = \mathbf{m} - \mathbf{y}$，计算出线性方程的解 $\mathbf{o} = L^{-1}\mathbf{u}$ 的解。然后计算 $(\mathbf{v}, \mathbf{o})$ 对仿射 R 的逆元 **s** 作为消息。

6.3.2　性能分析

相比于直接计算签名，使用预计算的方法进行 UOV 签名生成实际上负载会更大一点。在预计算完成之后我们需要存储预计算结果 $(\mathbf{v}, \mathbf{y}, L^{-1})$，这需要系统拥有更多的存储空间。表 6-1给出了在线离线 UOV 签名方案的计算复杂度。从表6-1 中我们可以看出，尽管使用预计算在线离线 UOV 签名的总的复杂度没有降低，但是运行时的能耗和延迟却大大降低了。当消息到来的时候，我们只需要进行两个简单的向量矩阵乘法操作。当 $v = 2o$ 时，这能降低签名延迟接近 o 倍。

表 6-1 在线离线 UOV 签名方案运算复杂度

	离线阶段	在线阶段
复杂度	$\Omega(o^3+v^2o)$	$\Omega(o^2+n^2)$

尽管在线离线 UOV 签名生成能够降低运行时能耗和延迟，但在实际实验中效果并不是很好。主要原因是预计算结果 $(\mathbf{v},\mathbf{y},L^{-1})$ 的大小太大，而无线传感网络节点的存储通常非常有限，往往没有足够多的空间存储足够多的预计算结果。假设我们有 10kB 的 RAM 和 1024kB 的 Flash，取 (GF(31),o=33,v=66) 为 UOV 签名方案的参数。每个预计算结果 $(\mathbf{v},\mathbf{y},L^{-1})$ 大约需要 10kB 的存储空间。RAM 完全无法满足要求，这时候我们只能将预计算结果其写入到 Flash 中。然而 Flash 的读写会带来额外的时间和存储损耗。另一个问题是，尽管我们使用了 Flash 进行预计算结果存储，由于 Flash 大小同样有限，实际上在能量峰出现的时候我们只能进行大约 100 次预计算，这极大地限制了预计算方法的可用性。要解决这个问题，我们必须要降低预计算结果的大小。

6.4 在线离线循环 UOV

相比较于 UOV 签名算法，循环 UOV 本身具有更小的私钥和更快的签名速度。它更加适用于无线传感网络节点。在本节中，我们将提出为能量收集节点设计的循环 UOV 在线离线签名方案。

6.4.1 基本方案

类似于普通 UOV 签名方案，我们主要关注于中心映射的求逆。下面我们给出循环 UOV 签名生成过程的在线离线阶段划分：

离线阶段：

随机选择醋变量$v_1,..,v_v\in(K^v)$，计算 $\mathbf{v}*B_1+\beta_1$ 得到矩阵 L 的第一行。写出矩阵 L 的多项式形式 $f(x)=\sum_{i=0}^{o-1}l_ix^i$，使用扩展欧几里得算法找到 $f(x)$ 在多项式环 $K[x]/(x^o-1)$ 上的逆元。如果逆元 $g(x)$ 不存在，说明矩阵 L 不可逆。则重新选择醋向量 $\mathbf{v}$。令 $y_k=(\mathbf{v}^T*A_k*\mathbf{v}+\mathbf{v}^T\cdot\alpha_k+c_k)\,(k\in[1,\cdots,o])$，得到向量 $\mathbf{y}$。存储预计算结果 $(\mathbf{v},\mathbf{y},g(x))$ 到内存中。

在线阶段：

首先计算消息 $\mathbf{m}$ 对可逆仿射 S 的逆 $\mathbf{w}=S^{-1}(\mathbf{m})$。然后计算 $\mathbf{u}=\mathbf{m}'-\mathbf{y}$ 并得到它的相关多项式 $u(x)$。通过计算 $u(x)*g(x)$ 得到中心映射油变量的解 $(\hat{x_1},\cdots,\hat{x_o})$。作用 R 的逆仿射到$(x_1,\cdots,x_n)$，得到签名。

6.4.2　性能分析

表 6-2给出了在线离线循环 UOV 签名生成的计算复杂度。综合表 6-2和表 6-1，我们可以发现在线离线循环 UOV 签名算法相比较于在线离线普通 UOV 签名算法，总的计算开销更小。实际上预计算方法非常适用于循环 UOV，存储其预计算结果 $(\mathbf{v}, \mathbf{y}, g(x))$ 的空间需求要比在线离线普通 UOV 小很多。对于参数为 (GF(31),o=33,v=66) 的在线离线循环 UOV 签名方案，存储一个预计算结果只需要 0.2kB 的存储空间，这比在线离线普通 UOV 的预计算结果小了 50 倍。它将极大的增大预计算的数量。同时，更小的预计算结果也降低了读写 Flash 内存的额外时间和能耗。

表 6-2 在线离线循环 UOV 签名方案运算复杂度

	离线阶段	在线阶段
复杂度	$O(v^2o)$	$O(o^2+n^2)$

在第四章中，我们对循环 UOV 和循环 Rainbow 方案进行了性能上的对比。循环 Rainbow 相比较于循环 UOV 拥有更好的性能。部分读者可能会问，为什么不使用循环 Rainbow 来构造在线离线方案。事实上，Rainbow 方案由于其多层结构，在线离线方法对其性能提高作用并不明显。

6.5　性能测试

在本节中，我们将对在线离线签名方案进行性能测试。我们使用实际生活中的太阳能数据，对以上几种在线离线签名方案进行仿真测试和实验。

6.5.1　测试平台

这里我们简单介绍我们的实验平台和能量收集装置。

在我们的实验中，我们使用 TelosB 作为无线传感节点。并用一个太阳能电池板和电容器对其进行供电，一颗不可充电电池作为备用能量源。TelosB 是一个最初由加州理工大学伯克利分校研发的无线传感模块，后来被 Crossbow 公司商业化。它具有 8MHZ 的 MSP430 微控制器 (16-bit RISC 处理器)、48kB 的 ROM、10kB 的 RAM、1024kB 的 Flash 和一个 IEEE 802.15.4 的无线收发器。我们使用 IXOLAR XOB17-04x3 太阳能电池和 1F Maxwell HC Powerseries 电容器作为能量收集平台。

如今的传感器节点一般都嵌入了较大存储量的 Flash 芯片。Flash 芯片是特定类型的 EEPROM，可以在单个操作中访问多个字节块。且 Flash 的存储是非易失性的，这意

味着我们不需要额外的能量来维持芯片中的数据。TelosB 节点使用 ST M25P80 40MHz 的 Flash 作为额外数据和代码存储器。它支持随机数据访问，但是与 CC2420 收发器共享 SPI 通讯总线。对该 Flash 进行写操作之前必须擦除要写入的块。为了能够方便对存储器进行管理，我们使用 TinyOS 系统 [112] 的接口来进行内存读写操作。

6.5.2 实验场景

本文中我们关注于无线传感节点在气候监控的运用。我们使用无线传感器节点对周围环境的气温和湿度每分钟进行两次监测和传输。我们使用 Sensirion SHT1x 传感器来进行温度和湿度的监控。测量得到的数据将会被签名方案进行签名，然后使用 IEEE 802.15.4 无线收发器将签名和数据一起发送出去。在非工作时间，微控制器处于睡眠阶段以降低能耗。

6.5.3 实现细节

参数选择：我们选择 GF(31) 作为 UOV 的基础域以获得更快的基础操作速度。根据文献 [78] 的结论和本文的论证，我们选择 $(GF(31), v = 38, o = 19)$ 来作为 UOV 和循环 UOV 安全级别为 2^{64} 的参数，对于 2^{80} 的安全参数，我们选用第三章中的结论。

混合表示：因为 MSP430 是一个 16-bit 的 RISC 处理器，我们使用 16-bit 整数来表示一个域上元素。GF(31) 上的元素实际上只需要 5-bit 就能够表示，使用 16-bit 来存储将会有很多冗余。但是这些冗余能够让我们使用延迟模规约等技术来提高域上基础算术的性能。但总的来说，在存储器中存储 16-bit 的操作数还是非常浪费存储空间。比较好的做法是使用混合表示法，我们可以将 3 个 5-bit 大小的元素存放进一个 16-bit 的内存中，在要使用到他们的时候将其变为 3 个 16-bit 的操作数。这样的混合表示，能够降低 65% 的存储负载。

延迟模规约：为了在基域上计算 $a + b$ 操作，我们首先要进行整数加法计算 $int(a) + int(b)$，然后将其结果模到 $[0, 30]$ 上。Shift-and-add 的方法可以被运用到模规约操作上以提高处理速度。因为我们在做计算时，使用 16-bit 整数来表示一个域上的元素，$int(a)$ 和 $int(b)$ 实际上远小于 2^{16}。所以在进行加法的时候，不会产生溢出。$int(a)$*$int(b)$ 的操作结果也严格小于 $(2^5 - 1)^2$，我们可以在进行了 64 次乘加操作之后才进行一次模运算，这极大的提高了计算效率。

6.5.4　预计算对系统性能的影响

我们在上述实验平台上实现了 2^{80} 安全级别的 UOV、在线离线 UOV、循环 UOV 和在线离线循环 UOV 等签名算法。我们对每个方案进行 1000 次性能测试并记录其平均性能和能耗在表 6-3中。能耗数据是通过运行时电压、电流和 MCU 的运行时间计算得来。这里因为我们仅在无线传感节点上实现签名算法，无需使用到公钥进行签名验证。故我们仅在表 6-3中记录签名性能和私钥大小。

表 6-3 TelosB 节点上各个方案的平均性能

方案	安全级别	私钥大小	离线阶段	在线阶段
UOV	2^{64}	18.8 kB	88ms/0.48mJ	
UOV	2^{80}	96.5 kB	409ms/2.21mJ	
在线离线 UOV	2^{64}	18.8 kB	112ms/0.60mJ	7ms/0.03mJ
在线离线 UOV	2^{80}	96.5 kB	508ms/2.74mJ	19ms/0.10mJ
循环 UOV	2^{64}	12.5 kB	59ms/0.31mJ	
循环 UOV	2^{80}	53.9 kB	244ms/1.32mJ	
在线离线循环 UOV	2^{64}	12.5 kB	50.4ms/0.27mJ	8ms/0.04mJ
在线离线循环 UOV	2^{80}	53.9 kB	223ms/1.21mJ	21ms/0.11mJ
ECDSA[104]	2^{80}	0.02 kB	3984ms/21.02mJ	
在线离线 ECDSA[104]	2^{80}	0.02 kB	–	485ms/2.62mJ

从表 6-3中我们可以看出循环 UOV 在 TelosB 上的运行时能耗和延迟大约是普通 UOV 的 60%，私钥大约是普通 UOV 的 55%。从另外一个角度我们可以看到，预计算的方法能够显著的降低 EH-WSN 节点实时能耗和延迟。对于普通 UOV 而言，预计算法能够降低 93% 的运行时能耗和延迟。对于循环 UOV，预计算能够降低 87% 的运行时能耗和延迟。同时表 6-3中我们将在线离线循环 UOV 和文章 [104] 中的在线离线 ECDSA 方案进行对比，结果显示在线离线循环 UOV 相对于在线离线 ECDSA 有明显的功耗优势。

6.5.5　预计算对系统可用性的影响

我们在开源仿真平台 GreenCastalia [113] 上仿真运行了 UOV、在线离线 UOV、循环 UOV 和在线离线循环 UOV 等签名算法。在仿真中，我们考虑和第6.5.2节一样的的实验场景。

在仿真能量收集功能之前，我们首先仿真仅使用电池进行供电的系统。我们使用 MSP430 的功耗模型来描述 TelosB 的能量模型。我们同时提供了一个计算读写 Flash 功耗和延迟的模型。每次使用板上 Sensirion SHT1x 进行测量的能量损耗设置为 3mW, 时间设置为 171ms。取 GreenCastalia 的默认通讯模块作为通信模块。图6-3与图6-4记录了在不使用预计算的情况下，2^{80} 安全级别的普通 UOV 和循环 UOV 的单日能量开销比例图。

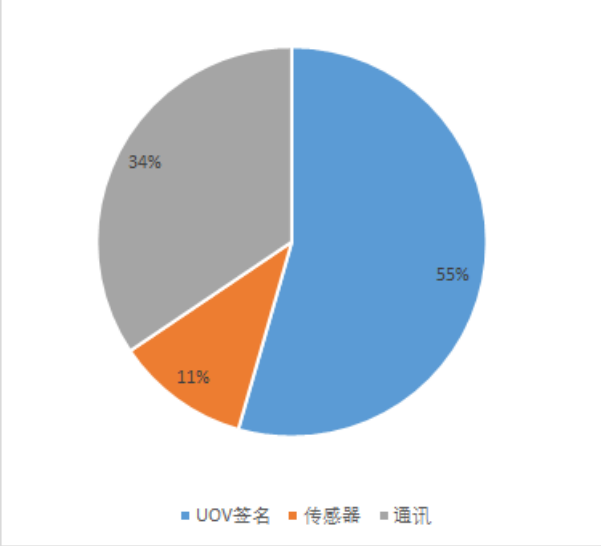

图 6-3 普通 UOV 单日能量开销比例图

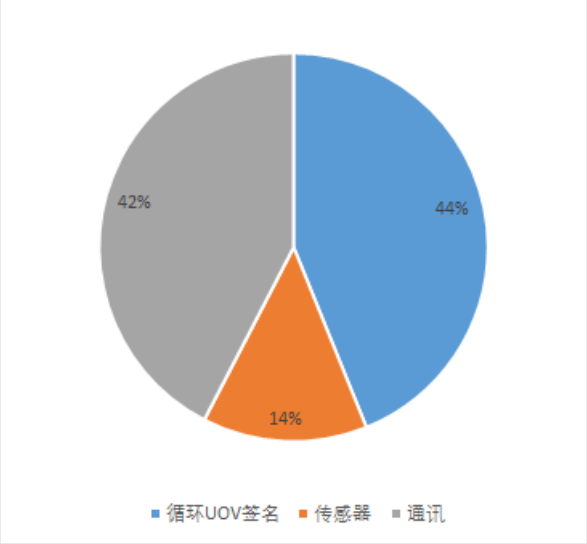

图 6-4 循环 UOV 单日能量开销比例图

从图6-3与图6-4中我们可以看出，两个系统的最大能量开销都是签名生成的计算开销。它大约占了普通 UOV 方案单日能耗的 55% 和循环 UOV 方案单日能耗的 44%。通讯的能量开销是第二大开销。它大约占普通 UOV 方案单日能耗的 34% 和循环 UOV 单

日能耗的 42%。传感器的运行能耗是占比最小的。对签名方案进行功耗优化，能够很好的降低系统总的能量开销。

为了更好的理解能量收集和预计算技术对 EH-WSN 节点的影响。我们接下来对由太阳能电池板和不可充电电池进行供电的系统进行仿真。我们的系统使用美国国家可再生能源实验室所发布的 2016 年纽约市太阳能数据进行仿真，随机选择纽约 8 月中连续 7 天的太阳能数据如图6-5。从图中我们可以看出，太阳辐射能量具有极大的方差。假设 I 是太阳照射到地面的单位能量，我们可以通过公式 $P_h = S \cdot \eta \cdot I$ 计算出面积为 S 转换率为 η 的太阳能电池板的瞬时能量收集。

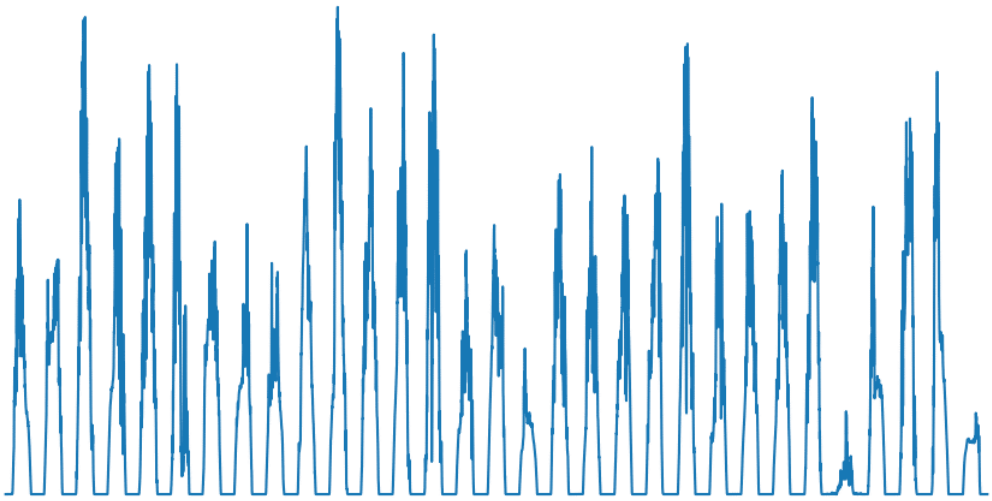

图 6-5　8 月纽约市连续 7 天太阳能辐射能量图

图 6-6记录了使用不同的签名方案时电容器的电能图。从中我们可以看到，当使用 UOV 作为签名方案时候，电容器中存储的电能会在 6 小时左右用完。预计算的方法可以让这个过程延长，但是由于存储限制，这个延长作用十分有限。循环 UOV 由于其拥有更快的签名方法，它相比于 UOV 消耗能量要慢一些。四个签名方案中，在线离线循环 UOV 签名方案能量消耗最慢。当预计算对足够多的时候，满能量的电容器能够支持系统运行接近 12 个小时。在同样是进行预计算的情况下，在线离线普通 UOV 与在线离线循环 UOV 的电容器电能图中，在放电阶段都存在一个明显的斜率变化，造成这个变化的原因是预计算对被用完了。由此可见，预计算方法对 UOV 签名方案的能量开销影响很大，它能够帮系统更加有效的利用收集到的环境能量。

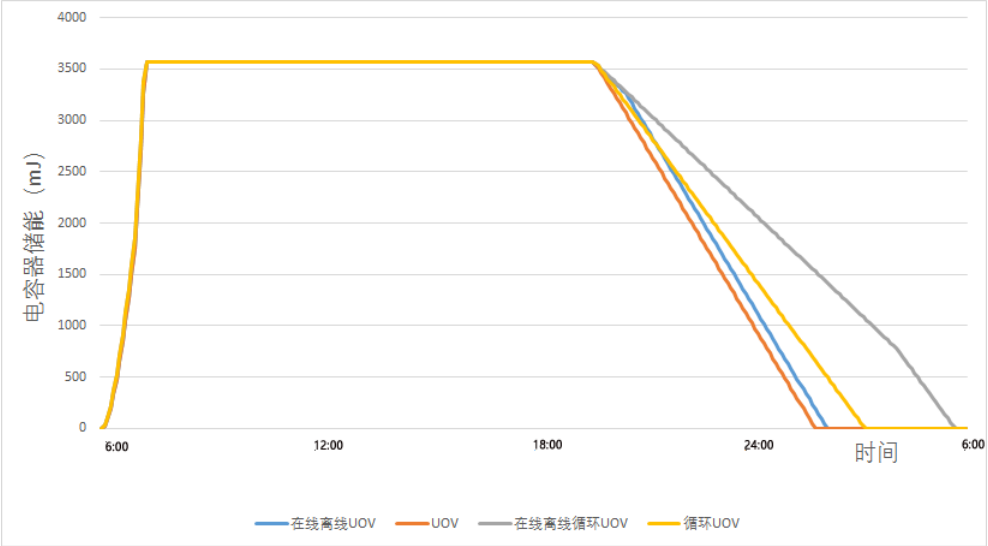

图 6-6 不同签名方案实验第一天电容器电能图

在使用太阳能电池主供电设备，电池作为备用供电设备的情况下。能量收集技术能够帮我们降低电池的能量开销。当从环境中收集到的能量充足的时候，系统使用收集到的能量进行供电。当从环境中收集到的能量不足以支持系统运行时，系统将消耗备用电池的能量。图 6-4给出了 EH-WSN 节点每天进行签名的电池能量损耗（太阳能优先供电，不可充电电池作为备用电源）。UOV 签名每日电池能耗仅为 1.92J，循环 UOV 每日电池能耗仅为 1.01J。很明显，能量收集技术的运用可以增长电池的使用寿命，降低系统的总的电能损耗。

表 6-4 太阳能优先供电，不可充电电池作为备用电源时各方案日均电池能耗

能耗类型	UOV	循环 UOV	在线离线 UOV	在线离线循环 UOV
日均通讯电池能耗	1.51	0.72	1.41	0.21
日均传感器电池能耗	0.31	0.22	0.29	0.07
日均签名电池能耗	0.10	0.07	0.09	0.02
日均总电池能耗	1.92	1.01	1.79	0.30

在线离线签名法在 EH-WSN 上对系统总可用时长的影响也十分明显。在线离线 UOV 签名方案签名的日均总电池能耗约为普通 UOV 签名的电池能耗的 93%，在线离线循环 UOV 签名方案的日均总电池能耗约为循环 UOV 的 30%。

6.5.6 实验结果

为了测量带预计算技术的循环 UOV 签名在 EH-WSN 平台上的实际影响。我们按第6.5.2节的实验场景进行实际实验测试。此处我们仅使用太阳能电池板为系统供电，选

取 2^{80} 安全级别的签名方案。

我们将该节点置于广州市华南理工大学计算机学院 B3 楼办公室的窗边，并记录 2017 年 5 月 25 日早上六点到 5 月 28 日早上六点的三天内各个签名方案产生的总的签名数到表6-5中。

表 6-5 仅使用太阳能供电，三天内各个签名方案产生的总的签名数

方案	UOV	循环 UOV	在线离线 UOV	在线离线循环 UOV
签名数	6287	6785	7647	8340

从表6-5中，我们可以预计算技术和能量收集技术的结合能够帮助我们获得更加绿色、可用性更高的方案。在线离线循环 UOV 签名方案结合能量收集技术能够很好的提高 EH-WSN 设备的能量使用效率。

6.6　小结

本章中，我们主要关注于多变量公钥密码的在无线传感网络中的运用。我们提出了适用于无线传感网络的在线离线循环 UOV 签名方案。并利用能量收集技术和预计算技术降低系统的整体能耗。我们从理论上讨论了方案的计算开销，并用仿真和实验的方法证实了在线离线循环 UOV 签名方案在具有能量收集功能的无线传感网络节点上的适用性。

结　束　语

随着互联网和物联网的发展，网络中的传输的重要数据越来越多，公钥密码方案在当前社会发挥着越来越重要的作用。量子计算机的出现，将对目前广泛应用的公钥密码体制造成严重威胁。毋庸置疑，设计抗量子计算机攻击的公钥密码方案是密码学当前非常重要和紧迫的任务。基于格的公钥密码体制、基于哈希的公钥密码体制、多变量公钥密码体制和基于编码的公钥密码体制是目前最受关注的四大后量子密码体制。尽管多变量公钥密码体制拥有近三十年的发展历史，但多变量公钥密码相关设计仍然处于一个不太能令人满意的阶段。许多多变量公钥密码方案被攻破，一些安全的多变量公钥密码方案在性能上又存在一定缺陷。

对多变量公钥密码体制的相关算法进行研究和改进是本文的主要内容。本文首先是对目前主流的多变量公钥密码方案和相关攻击算法进行简单介绍。然后针对目前方案的一些缺点，提出相应的变种改进方案。本文的主要核心思想是在多变量密码小域方案的中心映射中引入了部分旋转关系，并利用该旋转关系提高签名或解密速度，降低私钥大小。

在第三章中，我们将旋转关系引入到 UOV 中心映射中，使其具有更快的签名速度和更短的私钥。我们对这种旋转关系对性能和安全性的影响进行了理论分析和实验验证。我们提出了有限域上随机循环矩阵的可逆概率公式，从理论上证明了旋转关系引入对性能的提升。在安全性上，我们将目前所有针对 UOV 的攻击方案运用到循环 UOV 上，提出了适用于不同安全级别的循环 UOV 参数。

在第四章中，我们首先分析了近年来一些稀疏密钥 Rainbow 方案的安全性，并对其安全参数提出了对应修正。随后我将循环 UOV 方案扩展为循环 Rainbow 方案。相比于循环 UOV，由于循环 Rainbow 每层油变量数 o_i 相对更小。这对循环 Rainbow 在折中安全性和性能上折提出了更严格的要求。与循环 UOV 类似，我们对这种旋转关系对循环 Rainbow 在性能和安全性上的影响进行了理论分析和实验验证。同时我们将循环 Rainbow 与其他 Rainbow 变种方案和一些主流数字签名方案进行对比，实验结果显示循环 Rainbow 在签名和验证速度上具有明显优势。

在第五章中，我们提出了一种新型的 SRP 变种加密方案，我们称之为循环 SRP 加密方案。我们首先分析了 SRP 方案中心映射中冗余油醋多项式存在的原因，并提出新的方法除去这些冗余油醋多项式。这能够降低 SRP 方案的公私钥大小并提高其加解密

速度。随后我们将旋转关系引入到 SRP 中心映射中，进一步以提高其解密速度。在安全性上，我们从理论和实验两个方向细致的分析了循环 SRP 方案的安全性。在性能上，我们将循环 SRP 方案与多个公钥加密方案进行对比。实验结果证明循环 SRP 具有很强的竞争力。

在第六章中，我们主要关注于多变量公钥密码的在无线传感网络中的运用。我们提出了适用于无线传感网络的在线离线循环 UOV 签名方案。该方案将循环 UOV 签名生成过程分成在线离线两个阶段，并利用能量收集技术和预计算技术降低系统的整体能耗。我们从理论上讨论了方案的计算开销，并用仿真和实验的方法证实了在线离线循环 UOV 签名方案在 EH-WSN 节点上的适用性。

在整个方案设计和安全性分析过程中，我们总结了以下经验：

1) 尽管许多多变量公钥密码方案目前并没有可证明安全，但是随着针对多变量密码分析技术的发展，人们对多变量公钥密码的安全性已经有了更深入的理解。我们认为对多变量公钥密码的详尽安全性分析，能够帮助其满足安全标准。

2) 新的多变量公钥密码方案的提出，除了需要考虑传统的攻击方法之外，我们还需要专门针对新方案的特殊结构进行安全性分析。稀疏密钥结构的多变量公钥密码方案，往往会引入新的安全性弱点。但这些安全弱点并不意味着稀疏密钥的方法在多变量公钥密码中不可用。实际上一旦我们分析清楚这些安全性问题之后，我们可以通过修正方法和参数选择的方式，使多变量公钥密码达到需求的安全标准。

3) 多变量公钥密码在计算需求方面非常适用于在低功耗设备中使用。能量收集功能和预计算技术的结合使用能够进一步推进多变量公钥密码方案在无线传感网络中的应用。

我们的工作只是在改进了多变量公钥密码的道路上的一小步，但目前多变量公钥密码方案离实用普及还有一段距离。多变量公钥密码在普及之前，还有相当多问题需要解决。为了进一步改进多变量公钥密码，我们计划在未来的工作中关注以下方向。

1) 进一步降低多变量公钥密码密钥大小。密钥过大的问题，不仅影响多变量公钥密码在存储受限环境中的应用。密钥过大也会降降低多变量公钥密码的实际运行速度。只有将多变量公钥密码方案的密钥降低到资源受限设备可接受的大小，多变量公钥密码才能获得更多运用。

2) 进一步研究和改进多变量公钥密码的密码分析方法。我们关注于不断改进多变

量公钥密码的攻击方法和防御方法来提高人们对多变量公钥密码安全性的理解，从而进一步推动多变量公钥密码的应用。

3) 最后，也是我们觉得最有价值的一点。将多变量公钥密码更多应用于物联网、车联网等具体的应用环境中。只有更多的实践和应用，才能进一步推动多变量公钥密码的发展。

参考文献

[1] Diffie W, Hellman M. New directions in cryptography[J]. IEEE transactions on Information Theory, 1976, 22(6): 644–654.

[2] Rivest R L, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM, 1978, 21(2): 120–126.

[3] Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA)[J]. International Journal of Information Security, 2001, 1(1): 36–63.

[4] Shor P W. Algorithms for quantum computation: Discrete logarithms and factoring[A]. In: Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on[C], 1994: 124–134.

[5] Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. SIAM review, 1999, 41(2): 303–332.

[6] Bernstein D J, Buchmann J, Dahmen E. Post-quantum cryptography[M].[S.l.]: Springer Science & Business Media, 2009.

[7] Buchmann J A, Butin D, Göpfert F, et al. Post-Quantum Cryptography: State of the Art[M]. In: The New Codebreakers[C].[S.l.]: Springer, 2016: 88–108.

[8] Micciancio D, Regev O. Lattice-based cryptography[M]. In: Post-quantum cryptography[C].[S.l.]: Springer, 2009: 147–191.

[9] Dods C, Smart N P, Stam M. Hash based digital signature schemes[J]. Lecture notes in computer science, 2005, 3796: 96.

[10] Overbeck R, Sendrier N. Code-based cryptography[M]. In: Post-quantum cryptography[C].[S.l.]: Springer, 2009: 95–145.

[11] Ding J, Yang B Y. Multivariate public key cryptography[M]. In: Post-quantum cryptography[C].[S.l.]: Springer, 2009: 193–241.

[12] Matsumoto T, Imai H. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption.[A]. In: Eurocrypt[C], 1988. 88:419–453.

[13] Patarin J. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88[A]. In: Crypto[C], 1995. 95:248–261.

[14] Patarin J, Courtois N, Goubin L. Flash, a fast multivariate signature algorithm[J]. Topics in Cryptology

CT-RSA 2001, 2001: 298–307.

[15] Steinwandt R, Geiselmann W, Beth T. A theoretical DPA-based cryptanalysis of the NESSIE candidates FLASH and SFLASH[J]. Information Security, 2001: 280–293.

[16] Dubois V, Fouque P A, Shamir A, et al. Practical cryptanalysis of SFLASH[J]. Advances in Cryptology-CRYPTO 2007, 2007: 1–12.

[17] Ding J. A new variant of the Matsumoto-Imai cryptosystem through perturbation[A]. In: International Workshop on Public Key Cryptography[C], 2004: 305–318.

[18] Fouque P A, Granboulan L, Stern J. Differential Cryptanalysis for Multivariate Schemes.[A]. In: Eurocrypt[C], 2005. 3494:341–353.

[19] Ding J, Gower J E, et al. Inoculating multivariate schemes against differential attacks[A]. In: Public Key Cryptography[C], 2006. 3958:290–301.

[20] Patarin J. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms[A]. In: Advances in Cryptology-EUROCRYPT' 96[C], 1996: 33–48.

[21] Kipnis A, Shamir A. Cryptanalysis of the HFE public key cryptosystem by relinearization[A]. In: Crypto[C], 1999. 99:19–30.

[22] Courtois N. The security of hidden field equations (HFE)[J]. Topics in Cryptology CT-RSA 2001, 2001: 266–281.

[23] Granboulan L, Joux A, Stern J. Inverting HFE is quasipolynomial[A]. In: Crypto[C], 2006. 4117:345–356.

[24] Petzoldt A, Chen M S, Yang B Y, et al. Design principles for HFEv-based multivariate signature schemes[A]. In: International Conference on the Theory and Application of Cryptology and Information Security[C], 2015: 311–334.

[25] Porras J, Baena J, Ding J. ZHFE, a new multivariate public key encryption scheme[M]. In: Post-Quantum Cryptography[C].[S.l.]: Springer, 2014: 229–245.

[26] Ikematsu Y, Duong D H, Petzoldt A, et al. Revisiting the Efficient Key Generation of ZHFE[A]. In: International Conference on Codes, Cryptology, and Information Security[C], 2017: 195–212.

[27] Cabarcas D, Smith-Tone D, Verbel J A. Key Recovery Attack for ZHFE[A]. In: International Workshop on Post-Quantum Cryptography[C], 2017: 289–308.

[28] Vates J, Smith-Tone D. Key recovery attack for all parameters of HFE[A]. In: International Workshop

on Post-Quantum Cryptography[C], 2017: 272–288.

[29] Patarin J. The oil and vinegar algorithm for signatures[A]. In: Dagstuhl Workshop on Cryptography[C], 1997.

[30] Ding J, Schmidt D. Rainbow, a new multivariable polynomial signature scheme[A]. In: Applied Cryptography and Network Security[C], 2005: 317–366.

[31] Tao C, Diene A, Tang S, et al. Simple Matrix Scheme for Encryption.[J]. PQCrypto, 2013, 13: 231–242.

[32] Yasuda T, Sakurai K. A multivariate encryption scheme with rainbow[A]. In: International Conference on Information and Communications Security[C], 2015: 236–251.

[33] Sakumoto K, Shirai T, Hiwatari H. Public-key identification schemes based on multivariate quadratic polynomials[A]. In: Annual Cryptology Conference[C], 2011: 706–723.

[34] Hülsing A, Rijneveld J, Samardjiska S, et al. From 5-pass MQ-based identification to MQ-based signatures.[J]. IACR Cryptology ePrint Archive, 2016, 2016: 708.

[35] Sakumoto K. Public-Key Identification Schemes Based on Multivariate Cubic Polynomials.[A]. In: Public Key Cryptography[C], 2012. 7293:172–189.

[36] Ding J, Petzoldt A. Current State of Multivariate Cryptography[J]. IEEE Security & Privacy, 2017, 15(4): 28–36.

[37] Kipnis A, Patarin J, Goubin L. Unbalanced Oil and Vinegar signature schemes[A]. In: Advances in Cryptology-EUROCRYPT'99[C], 1999: 206–222.

[38] Gary M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-completeness[M].[S.l.]: WH Freeman and Company, New York, 1979.

[39] Clough C, Baena J, Ding J, et al. Square, a New Multivariate Encryption Scheme.[A]. In: CT-RSA[C], 2009. 5473:252–264.

[40] Ding J, Gower J E, Schmidt D S. Oil-Vinegar Signature Schemes[J]. Multivariate Public Key Cryptosystems, 2006: 63–97.

[41] Kipnis A, Shamir A. Cryptanalysis of the Oil and Vinegar signature scheme[A]. In: Annual International Cryptology Conference[C], 1998: 257–266.

[42] Menezes A J, van Oorschot P C, Vanstone S A. Some computational aspects of root finding in GF (qm)[A]. In: International Symposium on Symbolic and Algebraic Computation[C], 1988: 259–270.

[43] Wang L C, Yang B Y, Hu Y H, et al. A medium-field multivariate public-key encryption scheme[A].

In: CT-RSA[C], 2006. 3860:132–149.

[44] Patarin J. Asymmetric cryptography with a hidden monomial[A]. In: Advances in Cryptology CRYPTO 96[C], 1996: 45–60.

[45] Shen W, Tang S. RGB, a Mixed Multivariate Signature Scheme[J]. The Computer Journal, 2015, 59(4): 439–451.

[46] Braeken A, Wolf C, Preneel B. A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes.[A]. In: CT-RSA[C], 2005. 3376:29–43.

[47] Ding J, Yang B Y, Chen C H O, et al. New differential-algebraic attacks and reparametrization of Rainbow[A]. In: Applied Cryptography and Network Security[C], 2008: 242–257.

[48] Petzoldt A, Bulygin S. Linear recurring sequences for the UOV key generation revisited[M]. In: Information Security and Cryptology–ICISC 2012[C].[S.l.]: Springer, 2013: 441–455.

[49] Petzoldt A, Bulygin S, Buchmann J. Fast Verification for Improved Versions of the UOV and Rainbow Signature Schemes[M]. In: Post-Quantum Cryptography[C].[S.l.]: Springer, 2013: 188–202.

[50] Petzoldt A, Bulygin S, Buchmann J. A multivariate signature scheme with a partially cyclic public key[A]. In: In Proceedings of SCC 2010[C], 2010.

[51] Moh T. A public key system with signature and master key functions[J]. Communications in Algebra, 1999, 27(5): 2207–2222.

[52] Chen J M, Yang B Y. A more secure and efficacious TTS signature scheme[A]. In: International Conference on Information Security and Cryptology[C], 2003: 320–338.

[53] Yang B Y, Chen J M. Building secure tame-like multivariate public-key cryptosystems: The new TTS[A]. In: Australasian Conference on Information Security and Privacy[C], 2005: 518–531.

[54] Yasuda T, Ding J, Takagi T, et al. A variant of Rainbow with shorter secret key and faster signature generation[A]. In: Proceedings of the first ACM workshop on Asia public-key cryptography[C], 2013: 57–62.

[55] Yasuda T, Takagi T, Sakurai K. Efficient variant of Rainbow using sparse secret keys.[J]. JoWUA, 2014, 5(3): 3–13.

[56] Tan Y, Tang S. Two Approaches to Build UOV Variants with Shorter Private Key and Faster Signature Generation[A]. In: International Conference on Information Security and Cryptology[C], 2015: 57–74.

[57] Thomae E. A Generalization of the Rainbow Band Separation Attack and its Applications to Multi-

variate Schemes.[J]. IACR Cryptology ePrint Archive, 2012, 2012: 223.

[58] Bardet M, Faugere J C, Salvy B, et al. Asymptotic expansion of the degree of regularity for semi-regular systems of equations[A]. In: Mega[C], 2005: 1–14.

[59] Tao C, Xiang H, Petzoldt A, et al. Simple Matrix–A Multivariate Public Key Cryptosystem (MPKC) for Encryption[J]. Finite Fields and Their Applications, 2015, 35: 352–368.

[60] Faugre J C. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)[A]. In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation. ISSAC[C], 2002. 2:75–83.

[61] Billet O, Macario-Rat G. Cryptanalysis of the square cryptosystems[A]. In: International Conference on the Theory and Application of Cryptology and Information Security[C], 2009: 451–468.

[62] Thomae E, Wolf C. Roots of Square: Cryptanalysis of Double-Layer Square and Square+.[A]. In: PQCrypto[C], 2011: 83–97.

[63] Winkler F. Polynomial algorithms in computer algebra[M].[S.l.]: Springer Science & Business Media, 2012.

[64] Hillar G. Intel AVX2 Will Bring Integer Instructions with 256-bit SIMD Numeric Processing Capabilities[J]. Dr Dobb's Bloggers, Jun, 2011, 24.

[65] Becker T, Weispfenning V. Gröbner bases[M]. In: Gröbner Bases[C].[S.l.]: Springer, 1993: 187–242.

[66] Bosma W, Cannon J, Playoust C. The Magma algebra system I: The user language[J]. Journal of Symbolic Computation, 1997, 24(3): 235–265.

[67] Faugere J C. A new efficient algorithm for computing Gröbner bases (F4)[J]. Journal of pure and applied algebra, 1999, 139(1): 61–88.

[68] Bialostocki A, Lefmann H, Meerdink T. On the degree of regularity of some equations[J]. Discrete Mathematics, 1996, 150(1-3): 49–60.

[69] Lyubashevsky V, Peikert C, Regev O. On ideal lattices and learning with errors over rings[A]. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques[C], 2010: 1–23.

[70] Baldi M. QC-LDPC code-based cryptography[M].[S.l.]: Springer Science & Business, 2014.

[71] Petzoldt A, Bulygin S, Buchmann J. CyclicRainbow–A multivariate signature scheme with a partially cyclic public key[M]. In: Progress in Cryptology-INDOCRYPT 2010[C].[S.l.]: Springer, 2010:

33–48.

[72] Koblitz N, Menezes A J. Another look at provable security[J]. Journal of Cryptology, 2007, 20(1): 3–37.

[73] Hoffstein J, Pipher J, Silverman J H. NTRU: A ring-based public key cryptosystem[A]. In: International Algorithmic Number Theory Symposium[C], 1998: 267–288.

[74] Stehlé D, Steinfeld R. Making NTRU as Secure as Worst-Case Problems over Ideal Lattices.[A]. In: Eurocrypt[C], 2011. 6632:27–47.

[75] Regev O. The learning with errors problem[J]. Invited survey in CCC, 2010: 15.

[76] Kra I, Simanca S R. On circulant matrices[J]. Notices of the AMS, 2012, 59(3): 368–377.

[77] Lidl R, Niederreiter H. Finite fields[M]. Vol. 20.[S.l.]: Cambridge university press, 1997.

[78] Petzoldt A. Selecting and Reducing Key Sizes for Multivariate Cryptography[D]:[PhD Thesis]. [S.l.]: [s.n.] , 2013.

[79] Shamir A. Efficient signature schemes based on birational permutations[A]. In: Annual International Cryptology Conference[C], 1993: 1–12.

[80] Patarin J, Goubin L, Courtois N. C-+* and HM: Variations around two schemes of T. Matsumoto and H. Imai[A]. In: International Conference on the Theory and Application of Cryptology and Information Security[C], 1998: 35–50.

[81] Chen A I T, Chen M S, Chen T R, et al. SSE implementation of multivariate PKCs on modern x86 CPUs[M]. In: Cryptographic Hardware and Embedded Systems-CHES 2009[C].[S.l.]: Springer, 2009: 33–48.

[82] Yasuda T, Takagi T, Sakurai K. Efficient variant of Rainbow without triangular matrix representation[A]. In: Information and Communication Technology-EurAsia Conference[C], 2014: 532–541.

[83] Thomae E, Wolf C. Cryptanalysis of enhanced TTS, STS and all its variants, or: Why cross-terms are important[J]. Progress in Cryptology-AFRICACRYPT 2012, 2012: 188–202.

[84] Peng Z, Tang S. Circulant Rainbow: A New Rainbow Variant With Shorter Private Key and Faster Signature Generation[J]. IEEE Access, 2017, 5: 11877–11886.

[85] Billet O, Gilbert H. Cryptanalysis of Rainbow[A]. In: International Conference on Security and Cryptography for Networks[C], 2006: 336–347.

[86] Viega J, Messier M, Chandra P. Network security with openSSL: cryptography for secure communi-

cations[M].[S.l.]: O'Reilly Media, Inc, 2002.

[87] Barker E, Barker W, Burr W, et al. NIST Special Publication 800-57 Recommendation for Key Management–Part 1: General[G][S.l.]: Citeseer, 2012.

[88] Tsujii S. Public Key Cryptosystem using Nonlinear Equar tions[A]. In: The 8th Symposium on Information Theory and Its Applications1985[C], 1985: 156–157.

[89] Strang G. A proposal for Toeplitz matrix calculations[J]. Studies in Applied Mathematics, 1986, 74(2): 171–176.

[90] Bojanczyk A W, Brent R P, De Hoog F R, et al. On the stability of the Bareiss and related Toeplitz factorization algorithms[J]. SIAM Journal on Matrix Analysis and Applications, 1995, 16(1): 40–57.

[91] Clough C L, Ding J. Secure Variants of the Square Encryption Scheme.[J]. PQCrypto, 2010, 6061: 153–164.

[92] Duong D H, Petzoldt A, Takagi T. Reducing the Key Size of the SRP Encryption Scheme[A]. In: Australasian Conference on Information Security and Privacy[C], 2016: 427–434.

[93] Ars G, Faugere J C, Imai H, et al. Comparison between XL and Gröbner basis algorithms[J]. Advances in Cryptology-ASIACRYPT 2004, 2004: 157–167.

[94] Ding J, Gower J E, Schmidt D. Zhuang-Zi: A New Algorithm for Solving Multivariate Polynomial Equations over a Finite Field.[J]. IACR Cryptology ePrint Archive, 2006, 2006: 38.

[95] Ding J, Hu L, Nie X, et al. High order linearization equation (hole) attack on multivariate public key cryptosystems[A]. In: Public Key Cryptography[C], 2007. 7:233–248.

[96] Peng Z, Tang S, Chen J, et al. Fast Implementation of Simple Matrix Encryption Scheme on Modern x64 CPU[A]. In: International Conference on Information Security Practice and Experience[C], 2016: 151–166.

[97] De Clercq R, Roy S S, Vercauteren F, et al. Efficient software implementation of ring-LWE encryption[A]. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015[C], 2015: 339–344.

[98] Akyildiz I F, Su W, Sankarasubramaniam Y, et al. Wireless sensor networks: a survey[J]. Computer networks, 2002, 38(4): 393–422.

[99] Perrig A, Stankovic J, Wagner D. Security in wireless sensor networks[J]. Communications of the ACM, 2004, 47(6): 53–57.

[100] Wander A S, Gura N, Eberle H, et al. Energy analysis of public-key cryptography for wireless sensor networks[A]. In: Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on[C], 2005: 324–328.

[101] Priya S, Inman D J. Energy harvesting technologies[M]. Vol. 21.[S.l.]: Springer, 2009.

[102] Even S, Goldreich O, Micali S. On-line/off-line digital signatures[A]. In: Conference on the Theory and Application of Cryptology[C], 1989: 263–275.

[103] Bianchi G, Capossele A T, Petrioli C, et al. AGREE: exploiting energy harvesting to support data-centric access control in WSNs[J]. Ad hoc networks, 2013, 11(8): 2625–2636.

[104] Ateniese G, Bianchi G, Capossele A T, et al. Low-Cost Standard Signatures for Energy-Harvesting Wireless Sensor Networks[J]. ACM Transactions on Embedded Computing Systems (TECS), 2017, 16(3): 64.

[105] Ateniese G, Bianchi G, Capossele A, et al. Low-cost standard signatures in wireless sensor networks: a case for reviving pre-computation techniques[A]. In: Proceedings of NDSS 2013[C], 2013.

[106] Aysu A, Schaumont P. Precomputation methods for hash-based signatures on energy-harvesting platforms[J]. IEEE Transactions on Computers, 2016, 65(9): 2925–2931.

[107] Aysu A, Schaumont P. Precomputation Methods for Faster and Greener Post-Quantum Cryptography on Emerging Embedded Platforms.[J]. IACR Cryptology ePrint Archive, 2015, 2015: 288.

[108] Czypek P, Heyse S, Thomae E. Efficient implementations of MQPKS on constrained devices[J]. Cryptographic Hardware and Embedded Systems–CHES 2012, 2012: 374–389.

[109] Seo H, Kim J, Choi J, et al. Small private key MQPKS on an embedded microprocessor[J]. Sensors, 2014, 14(3): 5441–5458.

[110] Berbain C, Billet O, Gilbert H. Efficient implementations of multivariate quadratic systems[A]. In: Selected Areas in Cryptography[C], 2006. 4356:174–187.

[111] Kansal A, Hsu J, Zahedi S, et al. Power management in energy harvesting sensor networks[J]. ACM Transactions on Embedded Computing Systems (TECS), 2007, 6(4): 32.

[112] Levis P, Madden S, Polastre J, et al. TinyOS: An operating system for sensor networks[J]. Ambient intelligence, 2005, 35: 115–148.

[113] Benedetti D, Petrioli C, Spenza D. GreenCastalia: an energy-harvesting-enabled framework for the castalia simulator[A]. In: Proceedings of the 1st International Workshop on Energy Neutral Sensing

Systems[C], 2013: 7.

攻读博士学位期间取得的研究成果

一、已发表（包括已接受待发表）的论文，以及已投稿、或已成文打算投稿、或拟成文投稿的论文情况：

序号	作者（按顺序排序）	题　目	发表或投稿刊物名称、级别	发表的卷期年月、页码	相当于学位论文的哪一部分（章、节）	被索引收录情况
1	Zhiniang Peng, Shaohua Tang	Circulant Rainbow: A New Rainbow Variant With Shorter Private Key and Faster Signature Generation	IEEE Access	2017 年、卷号 5、页码 11877-11886	第四章	SCI，JCR Q1
2	Zhiniang Peng, Shaohua Tang	Circulant UOV: A New UOV Variant With Shorter Private Key and Faster Signature Generation	KSII Transactions on Internet and Information Systems	已录用	第三章	SCI，JCR Q4
3	Zhiniang Peng, Shaohua Tang, Ju Chen, Chen Wu	Fast Implementation of Simple Matrix Encryption Scheme on Modern x64 CPU	The 12th International Conference on Information Security Practice and Experience	2016 年、卷号 10060、页码 151-166	第五章	EI

续上表

序号	作者（按顺序排序）	题　目	发表或投稿刊物名称、级别	发表的卷期年月、页码	相当于学位论文的哪一部分（章、节）	被索引收录情况
4	Zhiniang Peng, Shaohua Tang	A Symmetric Authenticated Proxy Re-Encryption Scheme with Provable Security	The third International Conference on Cloud Compting and Security	2017 年、卷号 10603、页码 86-99	第五章	EI
5	Bo Lv, Zhiniang Peng, Shaohua Tang	A Secure Variant of the SRP Encryption Scheme with Shorter Private Key	The 13th International Conference on Information Security Practice and Experience	已录用	第五章	EI
6	Shaohua Tang, Bo Lv, Guomin Chen, Zhiniang Peng, Adama Diene, Xiaofeng Chen	Efficient hardware implementation of PMI+ for low-resource devices in mobile cloud computing	Future Generation Computer Systems	2015 年、卷号 52、页码 116-124	第二章	SCI，JCR Q1

续上表

序号	作者（按顺序排序）	题　目	发表或投稿刊物名称、级别	发表的卷期年月、页码	相当于学位论文的哪一部分（章、节）	被索引收录情况
7	Shaohua Tang, Bo Lv, Guomin Chen, Zhiniang Peng	Efficient Hardware Implementation of MQ Asymmetric Cipher PMI+ on FPGAs	The 10th International Conference on Information Security Practice and Experience	2014 年、卷号 8434、页码 187-201	第二章	EI
8	Zhiniang Peng, Shaohua Tang	Practical Hybrid Encryption from Simple Matrix Encryption Scheme		拟投稿	第五章	
9	Zhiniang Peng, Shaohua Tang	UOV Signature Scheme with Precomputation for Energy-Harvesting Wireless Sensor Network Platforms		拟投稿	第六章	

二、与学位内容相关的其他成果（包括专利、著作、获奖项目等）

专利类型	发明人	发明名称	申请号或公开号	申请日期	相当于学位论文的哪一部分（章、节）	受理情况
中国专利	陈驹，彭峙酿，唐韶华	一种多变量公钥的签名系统和方法	CN106330463A	2016.09.09	第四章	受理
中国专利	吴宸，彭峙酿，唐韶华	一种基于对称密码的可认证的代理重加密系统及方法	CN106534077A	2016.10.18	第五章	受理

获奖项目：2014“湖湘杯”全国网络信息安全公开赛，第一名，队长

致　谢

四年多的博士研究生学习生涯过得非常快。从刚开始读博士时候的雄心勃勃，到初遇困难时的迷茫，再到科研信心的重建。一路过来收获和感悟良多。回顾这一路的艰辛，感激之情油然而生。

首先我要特别感谢我的导师唐韶华教授热情关怀和悉心指导。在我整个博士研究生学习的过程中，唐教授倾注了大量的心血和汗水教育和培养我。在科研上，唐教授悉心细致传授我做研究的思路与方法，帮助我提升自身的研究能力。每次在研究方面遇到困难，唐教授也会耐心为我梳理讲解。可谓“授我以鱼又授我以渔”。在论文撰写方面，唐教授在论文的撰写方法以及成文定稿上，都给予了我悉心细致的教诲和无私的帮助。在生活上，唐教授经常主动关心我的生活，无私为我提供帮助。让我在生活上无后顾之忧，可专心于学术。在师从唐教授的这段时间内，他广博的学识、深厚的学术素养、严谨的治学精神和一丝不苟的工作作风也使我终生受益，在此表示真诚地感谢。

其次我要感谢我的母校“华南理工大学”。从本科到博士，我在华南理工大学待了八年多的时光。“博学慎思，明辨笃行”的校训深深的刻在了我的脑海里。漫漫人生中，八年生涯终身难忘。母校的教育之恩，终生感激。

然后，我要感谢和我一路走过来的师兄师弟、师姐师妹们。感谢“刘纽、沈武强、谈杨、陈家辉、李晓瑜、胡沐创、王婷、吕耀磊、王兴平、何伟明、吴健豪、李泽桦、黎凤霞、吕波、蒋科、赵实丰、凡冲、李伟、邓云、陆鸣、匡光彩、钟彩金、陈驹、李桂英、吴宸、魏新岸、林杰文、何志强、农彩勤、杨淑美、张远驰、何东霖、方东翔、任维涛、彭帆、杨志昊”等人。感谢一路上你们的陪伴与支持。

再者，我要感谢在学术研究上给予我支持和帮助的人。感谢胡红钢教授、龚征教授、吴昊天老师、蒋林智老师、陶成东博士、李宋宋同学、王润娴同学等人。感谢你们的无私帮助。

还有，我要感谢我的家人。家人是我精神上的支柱和动力，是你们的支持，让我一步一步走到现在。

最后，感谢此次参加论文评阅和答辩的老师专家们，谢谢你们。

彭峙酿

2017 年 12 月 5 日