QLW:A block cipher with high diffusion

based on Generalized Feistel Structure

**Abstract.** QLW是一个支持64/128bit明文，128bit密钥的轻量级分组密码，它基于generalized feistel structure(GFS)结构进行设计，因此也保持了此结构加解密一致的特性。QLW的设计为三型广义feistel的一个变种设计，不同于寻常的三型广义feistel结构利用两个或多个轮函数(Function)进行设计，此结构在三型广义feistel结构的基础经过一个公共的轮函数F,并且在每轮加密中，使用双重设计和移位等结构的配合使用，减少了非线性组件的构造和保持良好扩散性的优势。在轮函数F中，使用SP结构来设计轮函数，进一步弥补了feistel结构扩散性差的缺点，且在Sbox引入bit-slice技术设计，在保证安全的前提下增加加解密的速率。在密钥扩展中，通过轮密钥控制每轮常数C的生成，实现了每生成一轮密钥，就会产生不同的常数C，使得每轮常数动态变化，进一步增加了安全性。

QLW密码的安全界清晰，可以有效抵抗差分攻击、线性攻击已知攻击,有较大的安全冗余。算法的基本运算为循环移位、异或等逻辑运算，易于软件和硬件实现，算法实现方式灵活，软硬件速度快，可适用于多种平台。

**Key:**轻量级；广义feistel；SPN；bit-slice

**1 Introduction**

随着信息技术的蓬勃发展，对数据安全的关切日益加深。在众多应用场景中，轻量级分组密码作为一种专为资源受限环境设计的加密算法，备受瞩目。其核心设计理念是在计算和存储资源受限的情况下，仍能提供足够的安全性，同时降低硬件和软件实现的成本。

传统的分组密码算法，如AES和DES，在通用计算环境中表现卓越，然而在资源受限的场景，如物联网设备、嵌入式系统和移动设备等，它们显得过于复杂，不适合轻量级应用。轻量级分组密码的兴起填补了这一空白，成为解决资源受限环境下数据安全性问题的有效手段。

当前，轻量级分组密码领域涌现了众多创新性设计和研究方向，学术界和工业界都在不懈努力探索新的设计策略、分析方法以及实际应用中的性能表现。轻量级密码学的研究旨在在有限的资源条件下设计高效、紧凑的加密算法，以适应不断增长的物联网和边缘计算等领域的需求。

因为5G通信技术的全球推广，我们正处于信息传输领域的关键时刻。5G的快速传输速度不仅确保了实时数据的迅捷传输，而且推动了第四次工业革命的不断发展，该革命主要以物联网（IoT）技术为核心。在这个新时代，信息的规模和重要性变得前所未有，同时也带来了巨大的安全挑战。特别是在5G协议下，随着大量数据流中包含着私人和敏感信息，我们迫切需要创新性的物联网保密通信方案，以确保信息的机密性和完整性。在这一背景下，加密算法成为维护通信安全的关键组成部分，而轻量级分组密码则是应对资源受限环境的一项重要技术。

近年来，国际学术界在轻量级分组密码领域取得了显著的进展，涌现出一系列令人瞩目的算法，如SCENERY、WARP、SIMON、SPECK、Shadow、VH、BORON、CHAM、LVPDA、GIFT、QTL、FPL、SFN和RECTANGLE等。这些密码算法以其在硬件实现上的高效性和优越的性能指标而备受关注。以PRESENT为例，它在CHES 2007中首次亮相，以1570个GE的硬件实现成本在资源受限环境中取得了良好的适应性，特别适用于传感器网络和RFID标签等场景。与此同时，SIMON密码由美国国家安全局发布，采用了非传统的AND-RX/ARX操作，其在SMIC 0.13μm上的硬件实现仅需1751个GEs，显示出其在资源利用上的高效性。

这些轻量级分组密码的涌现为在硬件资源受限的设备上提供高效加密方案打开了新的可能性。在本文中，我们将深入研究轻量级分组密码的最新进展，探讨其在实际应用中的性能和安全性，以期为未来的安全通信技术提供更加创新和可行的解决方案。

类似于SIMON密码的算法，由于采用Feistel结构，存在相对缓慢的扩散特性。这使得类SIMON的密码通常需要更多的轮次来实现足够的扩散，进而在硬件实现中导致更多的面积消耗。尽管Feistel结构在加密和解密过程中保持一致，能够节约一半的资源，但由于每一轮操作只改变了一半的数据，其扩散速度相对较慢。相反，SPN结构具备良好的扩散性，一轮运算就改变了所有数据，但其加密和解密过程通常不一致，需要更多的资源投入。此外，设计带有密钥控制的高安全性可逆非线性函数S也相对困难。QLW算法独特之处在于结合了广义Feistel和SPN的优点，实现了加解密一致性，同时在一轮加密运算中影响所有明文的bit位，具有高安全性和高扩散性。在本文中，QLW采用了一种新的4分支广义Feistel的变体结构，以平衡安全性、成本和效率。在结构的中间部分，引入了2个分支的XOR操作，使得所有分支相互关联，进一步提高了算法的复杂性和安全性。

QLW主要包括两个方面。

（1）SPN与广义Feistel相结合

（2）设计密钥与常数的相关性，增加轮函数安全性。

（3）具有优良的扩散性。

**2 Specifiction of QLW**

QLW基于4分支的广义Feistel（GFS）结构，具有64bit明文和128bit密钥的分组长度。轮数(RN)为16轮，算法表示为QLW-64。

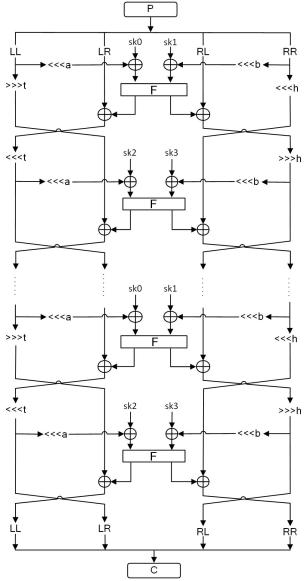
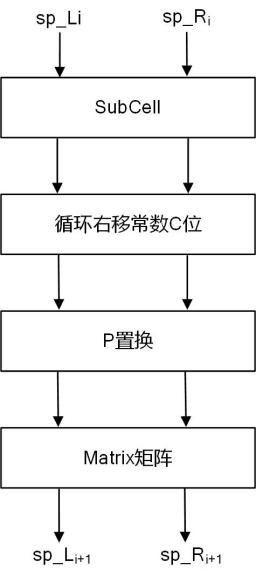
2.1 Notations

在QLW的规范中，我们使用了表1中的以下符号。

|  |  |  |  |
| --- | --- | --- | --- |
| **table 1** Notation of QLW | | |  |
| Notations |  |  | Descriptions |
| P |  |  | 64-bit plaintext |
| C |  |  | 64-bit ciphertext |
| K |  |  | 128-bit master key or round key |
| SK**i r** |  |  | 16-bit round subkey SK**i** in the r round 64-bit plaintext |
| F |  |  | Function F |
| **||** |  |  | Concatenation of two binary strings |
| **<<<** |  |  | Left circular shift operation |
| **>>>** |  |  | Right circular shift operation |
| ⊕ |  |  | Bitwise exclusive-OR operation |

2.2 Encryption process

|  |
| --- |
| **Algorithm 1** QLW Encyption Routine |
| **Input:** Plaintext,key; |
| **Output:** Ciphertext; |
| 1. (LL,LR,RL,RR)←Plaintext; |
| 2. for r = 1 to (RN-1) do |
| 3. M = (LL<<<a⊕sk0); N = (RR<<<b⊕sk1); |
| 4. state0 = M||N; |
| 5. state1 = Function(state0); |
| 6. state1 = spL0||spR0; (将state1分为2个16-bit的分组长度) |
| 7. LR = (LR⊕spL0); RL = (RL⊕spR0); |
| 8. LL = (LR<<<t); LR = (LL>>>t); RL = (RR<<<h); RR = (RL>>>h); (前半轮交换) |
| 9. U = (LL<<<a⊕sk2); V = (RR<<<b⊕sk3) |
| 10. state2 = U||V; |
| 11. state3 = Function(state2); |
| 12. state3 = spL1||spR1; (将state3分为2个16-bit的分组长度) |
| 13. LL = (LR⊕spL1); LR = LL; RL = RR; RR = (RL⊕spR1); (后半轮交换) |
| 1. end for |
| 1. LL = LL; LR = LR; RL = RL ;RR = RR; (最后一轮不交换) |
| 1. Ciphertxet←(LL,LR,RL,RR); |
| 1. Return Ciphertext; |

**Fig. 1** The encryption of QLW **Fig. 2** The function of QLW

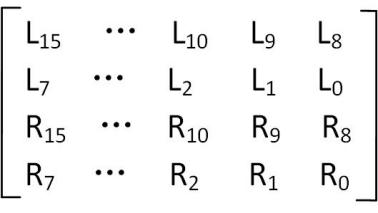
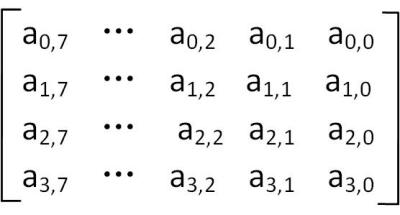
由于本文设计的QLW的特殊结构，解密过程与加密过程一致，轮密钥的使用顺序相反。QLW主要包含这几个组件：1）Function函数；2）Rotation；3）XOR。QLW算法的轮函数如Fig 1所示。在QLW加密的过程中，明文分组长度被分成几个相同大小的分组长度。通过初始密钥产生的多个子密钥，在多次迭代后生成密文。如Algorithm 1中所述。

从Fig 1可以看出，QLW的轮函数可以简化为上下两层的3-型广义Feistel结构。在这种情况下，每轮加密分为两层（前半层与后半层），都是经过一个Function（F）函数，结构如Fig 2所示。F函数由SPN结构构成，包括SubCells、Rotation、PermBits与Matrix矩阵这五个部分。

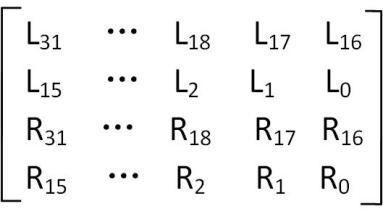
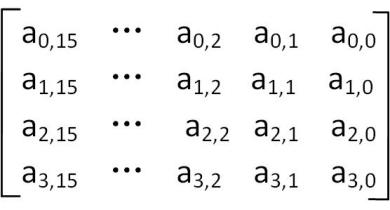
1）SubCells：在QLW中使用的S-box是一个4bit到4bit的S-box，F2 4→F2 4。此S-box的十六进制转换如Table 2所示。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 2** | | | | | | | | | | | | | | | | |
| X | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| S[X] | C | 5 | 6 | B | 9 | 0 | A | D | 3 | E | F | 8 | 4 | 7 | 1 | 2 |

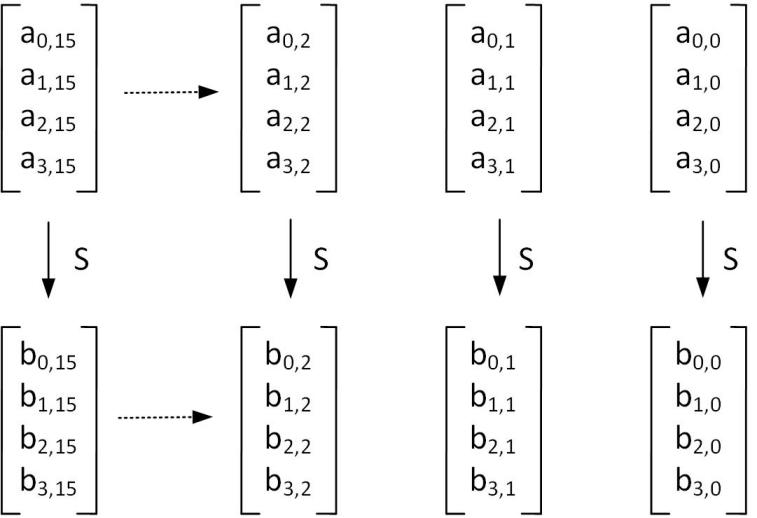
并且在QLW算法中，通过使用位片技术来保证其轻量级和快速的实现，并行应用s盒到同一列中的4位。子列的操作情况Fig 9所示。Fig 5与Fig 6表示为QLW-64的实现方式，Fig 7与Fig 8表示为QLW-128的实现方式。

**Fig. 5** 32-bit cipher state **Fig. 6**  Two dimensional way

**Fig. 3** 64-bit cipher state **Fig. 4**  Two dimensional way



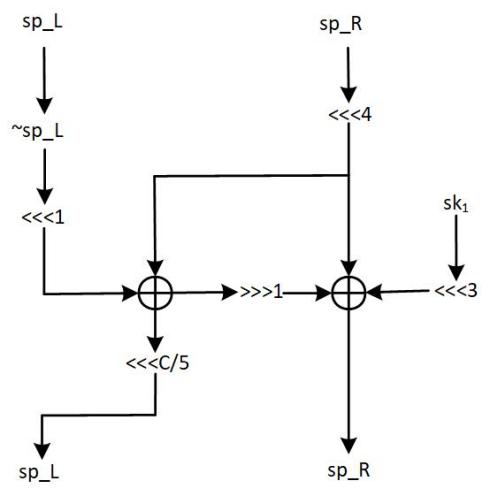
**Fig. 5** SubColumn Operates on the Columns of the State

2）Rotation：经过并行S盒进行字节代换后，将中间值spLL，spRR组合成64-bit state，轮常数C为通过每轮密钥扩展进行生成，将state循环左移轮常数C-bit，增加算法的混淆性。

3）PermBits：在QLW的P置换如Table 3所示。State中的位置i被移动到位置P(i)。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 3** | | |  |  |  |  |  |  |
| **i** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **i** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| P[i] | 0 | 14 | 22 | 25 | 4 | 17 | 30 | 9 |
| **i** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** |
| P[i] | 1 | 7 | 23 | 13 | 31 | 26 | 2 | 8 |
| **i** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| P[i] | 18 | 12 | 29 | 5 | 21 | 10 | 3 | 24 |

4）Matrix矩阵：在QLW的矩阵的设计中，使用矩阵线性实现结构来实现。如Fig 6所示。



**Fig. 6** Matrix矩阵

2.3 Decryption process

由于本文设计的QLW加密算法的特殊结构，解密过程与加密过程一致，轮密钥被反向使用。解密算法中的子密钥的顺序与加密过程相反。例如，第一轮解密中的子密钥是最后一轮加密过程中的子密钥，值得注意的是为了方便可以直接解密，因此在最后一轮（RN轮）加密设计中，该轮所产生的密文不进行交叉互换。

2.4 Key schedule

在QLW的密钥扩展中。初始密钥Key=k0||k1||k2···||k126||k127。轮常数用C表示。密钥扩展过程包括此四个部分：轮常量C生成、Matrix矩阵、Subcell代换、循环移位。下面为详细介绍。

1）轮常量C生成。

C = k0k1k2k3k4 ⊕ k3k4k5k6k7 ⊕ r（1≦r≦12）

2）Subcell代换。取出每行中间4bit进行S-box字节代换，更新原位置的值，即：

k8n+2k8n+3k8n+4k8n+5 ← S[k8n+2k8n+3k8n+4k8n+5] (0≦n≦14)

3）128-bit的P置换

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| p[i] | 18 | 79 | 8 | 13 | 22 | 83 | 12 | 17 | 26 | 87 | 16 | 21 | 30 | 91 | 20 | 25 |
| i | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| p[i] | 2 | 63 | 120 | 125 | 6 | 67 | 124 | 1 | 10 | 71 | 0 | 5 | 14 | 75 | 4 | 9 |
| i | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| p[i] | 50 | 111 | 40 | 45 | 54 | 115 | 44 | 49 | 58 | 119 | 48 | 53 | 62 | 123 | 52 | 57 |
| i | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| p[i] | 34 | 95 | 24 | 29 | 38 | 99 | 28 | 33 | 42 | 103 | 32 | 37 | 46 | 107 | 36 | 41 |
| i | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| p[i] | 82 | 15 | 72 | 77 | 86 | 19 | 76 | 81 | 90 | 23 | 80 | 85 | 94 | 27 | 84 | 89 |
| i | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| p[i] | 66 | 127 | 56 | 61 | 70 | 3 | 60 | 65 | 74 | 7 | 64 | 69 | 78 | 11 | 68 | 73 |
| i | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| p[i] | 114 | 47 | 104 | 109 | 118 | 51 | 108 | 113 | 122 | 55 | 112 | 117 | 126 | 59 | 116 | 121 |
| i | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| p[i] | 98 | 31 | 88 | 93 | 102 | 35 | 92 | 97 | 106 | 39 | 96 | 101 | 110 | 43 | 100 | 105 |

4）循环移位。

将字节代换后的128-bit key分为U0，U1，U2，U3四个分组长度,每个分组长度为16bit。

即：key=U0||U1||U2||U3

将U0、U1、U2、U3分别循环左移1、2、7、11位。

即：keyʹ=U0<<<1||U1<<<2||U2<<<7||U3<<<11

然后，对Keyʹ=k0ʹ||k1ʹ||k2ʹ···||k126ʹ||k127ʹ进行循环左移1bit位，即可得到每轮所需的128-bit的轮密钥SK。

即：SK=Kʹ<<<1

最后即可从轮密钥得到4个16bit的子密钥sk0,sk1,sk2,sk3，即SK=sk0||sk1||sk2||sk3。

**3 Motivation for Design Choices of QLW**

3.1 High-level structure

QLW支持64位分组长度以适应标准应用程序，以及64位和128位密钥以实现适度的安全级别。为了解决轻量级分组长度密码在传统的Feistel结构中的缓慢扩散的问题，我们设计了一种新的具有快速扩散的新型广义Feistel结构。传统的Feistel类型的结构在一个迭代回合中只改变了一半的分组长度消息，因此其扩散速度较差。QLW采用了一种具有4个高对称分支的新型广义Feistel变体结构，中间2分支在轮函数的作用下进行加密，外围2分支均进行循环移位操作，在经过一轮加密操作后，使得明文分组长度能在一轮加密中就能改变了所有的分组长度消息，因此克服了传统Feistel结构只加密一半明文的缺点。该结构的另一个优点是在轮函数（Function）中进行SPN结构的设计，使QLW能在有效兼顾各自结构的优势前提下，设计出了一种具有加解密一致且扩散速度快的轻量级分组密码，并且能有效的降低资源面积消耗，满足轻量级的设计标准。

同样在S盒字节代换设计中，子列基于bit-slice技术实现。由于结构特点，不需要考虑子列的逆过程，整个解密算法也可以循环加密过程。它支持一个最小的微分特征概率和最佳的线性特征近似，其中用于微分分析和线性分析的密码是安全的。因此，QLW的结构提高了Feistel型结构中轻量级分组长度密码的安全性。

3.2 Functions structure

3.2.1 Design criteria of the S-box

对于n×m的S盒，n位二进制输入将产生m位二进制输出。从安全性能方面来看，n和m值越大，非线性程度和混淆程度越高。从实现效率来看，分组密码中使用的S盒越大消耗的资源越多，加密算法实现效率越低。DBST在对S盒尺寸的选择经过深思熟虑后，决定使用4×4的S盒来实现高效的硬件性能和满足轻量级分组密码标准。

定义1：代换。让S:→。对于任意的x、x′∈和x≠x′，都有S(x)≠S(x′)，那么S就是一个代换。

定义2：差分均匀性。设f是一个定义在→上的函数。对于(a,b)∈×对应的任意差值，定义集合Df(a→b)={x∈|f(x⊕a)⊕f(x)=b}，集合Df(a→b)中的元素个数为δf(a,b)，则函数f的差分均匀度为δ(f)=maxa≠0,b，δf(a,b)，函数f称为δ(f)-均匀。

定义3：线性。设f是定义在→上的函数。对于任何线性组合，定义Walsh变换为:

×

那么函数f的线性为,其中<b,f(x)>和<a,x>表示内积运算。

定义4：S盒的差分分布表。设m,n∈n，将到的非线性映射(也称为S盒)记为:S:→。给定α∈，β∈，以α为S盒的输入微分，β为S盒的输出微分，构造个表。行和列的交集取值Ns(α,β)。





定义5：S盒的线性近似表。设m,n∈n，将到的非线性映射(也称为S盒)记为:S:→。给定α∈，β∈以α为S盒的输入掩码，β为s盒的输出掩码，构造个表。行和列的交集取值Ns(α，β)-2n−1。





QLW的S盒设计标准如下：

1)S盒满足定义1的代换准则。

2)S盒为差分均匀性δ(f)=maxa≠0,b，δf(a,b)≦4。

3)S盒是线性,≤8的4×4的S盒。

3.2.2 The Permutation and Shift Transformation

为了QLW分组内比特之间的混淆性和扩散性，简化了算法的设计和实现，保证硬件和软件实现更为灵活，因此我们在轮函数中采用了两种高效的线性层结构，分别是P置换和循环移位。。

QLW-64与QLW-128分别为一个32bit的P置换与一个128bit的P置换。P置换可以确保输入分组的每个比特与输出分组的每个比特都有复杂的关联。因此，对输入的微小改变会导致输出的广泛变化，提高了密码算法的扩散性。这对于防止局部攻击和差分攻击具有重要意义，在硬件中非常容易实现。此外，它还可以使密码算法在处理不同输入时产生相似的输出模式，这对于减轻侧信道攻击（如时序分析、功耗分析等）的影响至关重要。通过混淆和扩散，P置换可以降低对输入的侧信道分析的效果。

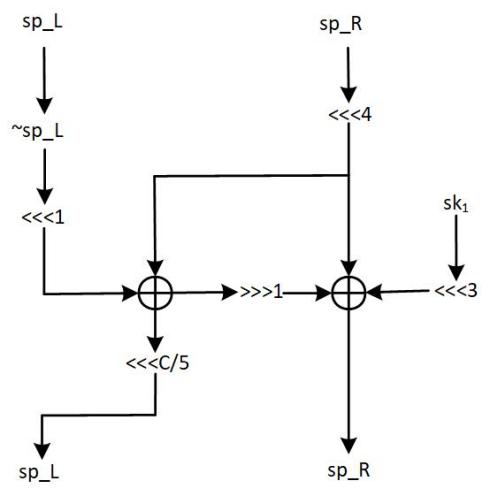
在循环移位的设计中，与传统循环移位去设计一个固定的循环移位常数有所不同，QLW循环移位是进行一个轮常数C动态的循环移位，因为轮常数由密钥扩展生成，因此受密钥扩展的影响，能够提高加密过程的安全性。其次，循环移位的结构是特殊和高效的，在软硬件受限的环境中可以进行很好的实现。并且还可以提高对密码分析的安全性。对与循环移位，我们充分考虑其作为密码结构的对合性质，它不破坏加密过程，与解密过程相同。具体来说，在安全条件下，这两个线性层是没有任何硬件成本的，并且它可以增加分组内比特之间的混淆性和扩散性。通过在每一轮中对分组进行不同方向和不同位数的循环移位，可以确保输入的微小改变在输出中产生广泛的变化，从而增强密码算法的扩散性。

P置换和循环移位为轻量级分组密码的设计旨在提高安全性、增强非线性性、适应资源受限环境等方面的起关键作用。这些设计选择是在实现对抗各种密码分析攻击的同时，确保算法在轻量级环境中仍然能够高效运行的重要考虑因素

3.2.3 The linear matrix

在QLW的矩阵设计中，采取线性的方式去进行矩阵的设计，通过其结合非线性元素和灵活性，达到增加混淆、提高扩散性、适应性和硬件实现简单的目的，以满足轻量级密码算法在资源受限环境下的需求。

线性矩阵的设计可以通过合理选择和设计线性矩阵，可以使加密过程中的数据更加均匀地分布在密码算法的不同部分，提高密码算法的扩散性，使得输入的微小变化会导致输出的巨大变化，增强密码算法的安全性。尽管线性矩阵是线性运算的组合，但在整体密码算法中，可以与其他非线性运算（如S盒）结合使用，从而引入更多的非线性元素来增加密码算法的混淆度，提高其抵抗不同类型的密码分析攻击的能力。



**Fig. 6** 线性矩阵

QLW的轮函数的线性矩通过取反、循环移位、异或线性操作来构造而成。与传统的线性矩阵的设计有所不同，QLW的线性矩阵引入了一个轮常数C/5动态的循环移位，并且同样引入每轮所生成的轮密钥sk1，由于轮常数C与轮密钥sk1由密钥扩展生成，使得线性矩阵过程能受到密钥扩展生成的影响，有助于增加密码算法的混淆度，提高其抵抗不同类型的密码分析攻击的能力，增加加密过程的安全性。QLW的线性矩阵结构简单，在硬件实现中，不需要大量的门和逻辑电路，使得其在硬件实现中更容易设计和验证，为硬件实现可以降低成本并提高性能。与MDS矩阵相比，在混淆与扩散性相差不大的前提下能够具有更加轻量的特征。

3.3 The key schedule

我们的目标是设计一种硬件高效且快速的扩散加密算法。因此密钥扩展应该满足以下三个基本要求。

(1)硬件面积消耗应尽可能小;

(2)每个子密钥应尽可能受到主密钥的影响;

(3)密钥调度中的延迟应尽可能低。

为了在这三个方面取得平衡，我们选择采用线性密钥扩展的主要思路。在密钥扩展的过程中，我们在QLW中使用了基于轮的Feistel结构，这种设计虽然在一定程度上牺牲了一些延迟，但确保了整个密钥的良好扩散。我们将这一设计原则应用于QLW，并注重考虑面积消耗和扩散特性。通过实验结果，我们最终选择了P置换和循环移位，主要考虑到微分分析和线性分析的影响。这些选择显著提高了密钥扩展的扩散率。此外，我们还在QLW中引入了十进制整数作为轮常数，以避免可能的循环移位对称和滑动攻击。通过这些努力，我们成功得出了最终的密钥扩展方案。

**4 Security evaluation**

4.1Differential cryptanalysis

差分密码分析是一种通过分析输入差分和输出差分之间关系的密码分析方法。对于QLW，我们首先定义了适用于其结构的差分特性。通过广泛的实验和理论分析，我们评估了各种差分路径对密钥恢复攻击的有效性。在实验中，我们确保了密钥空间的充分探索，并通过实际攻击尝试来衡量差分路径的成功率。结果表明，QLW能够抵御高概率差分攻击，其设计原则在提供充分的安全保障。

4.2 Linear cryptanalysis

线性密码分析是另一种常见的密码分析技术，它利用线性逼近来寻找密码算法的弱点。对QLW进行线性密码分析，我们首先确定了最佳的线性逼近式，并计算了相关的偏差值。通过比较不同线性逼近式的偏差值，我们评估了QLW在各轮次的安全性。实验结果表明，QLW具有高度的非线性特性和复杂的密钥依赖结构，使其能有效阻断线性攻击。

4.3 Impossible differential cryptanalysis

不可能差分密码分析是一种先进的密码分析技术，它利用密码算法中不可能发生的差分路径来攻击。对于QLW，我们详细研究了其结构特点和可能的不可能差分路径。通过对潜在的不可能差分路径进行分类和评估，我们确定了最有潜力的攻击矢量。接着，我们设计了针对这些路径的攻击，并评估了其对QLW安全性的影响。研究结果显示，QLW对常见的不可能差分路径具有内在的抵抗力，无法通过这些路径有效地攻破其安全性。

**5 Hardware performance**

**6 Conclusion**

**References**

1. Bogdanov A,Knudsen L R,Leander G,Paar C,Poschmann A,Robshaw M J B,Seurin Y,Vikkelsoe C. PRESENT: an ultra lightweight block cipher. In: Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems. 2007,450–466
2. Feng J, Li L. SCENERY: a lightweight block cipher based on Feistel structure. Frontiers of Computer Science, 2022, 16(3): 163813
3. Banik S, Bao Z, Isobe T, Kubo H, Liu F, Minematsu K, Sakamoto K, Shibata N, Shigeri M. WARP:revisiting GFN for lightweight 128-bit block cipher.In:Proceedings of the 27th International Conference on Selected Areas in Cryptography.2020,535–564
4. Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L. The SIMON and SPECK lightweight block ciphers. In: Proceedings of the 52nd Annual Design Automation Conference. 2015, 175
5. Guo Y, Li L, Liu B. Shadow: a lightweight block cipher for IoT nodes. IEEE Internet of Things Journal, 2021, 8(16): 13014–13023
6. Dai X, Huang Y, Chen L, Lu T, Su F. VH: a lightweight block cipher based on dual pseudo-random transformation. In: Proceedings of the 1st International Conference on Cloud Computing and Security. 2015, 3–13
7. Bansod G, Pisharoty N, Patil A. BORON: an ultra-lightweight and low power encryption design for pervasive computing. Frontiers of Information Technology & Electronic Engineering, 2017, 18(3): 317–331
8. Koo B, Roh D, Kim H, Jung Y, Lee D G, Kwon D. CHAM: a family of lightweight block ciphers for resource-constrained devices. In: Proceedings of the 20th International Conference on Information Security and Cryptology. 2017, 3–25
9. Zhang J, Zhao Y, Wu J, Chen B. LVPDA: a lightweight and verifiable privacy-preserving data aggregation scheme for edge-enabled IoT. IEEE Internet of Things Journal, 2020, 7(5): 4016–4027
10. Banik S, Pandey S K, Peyrin T, Sasaki Y, Sim S M, Todo Y. GIFT: a small present: towards reaching the limit of lightweight encryption. In: Proceedings of the 19th International Conference on Cryptographic Hardware and Embedded Systems. 2017, 321–345
11. Li L, Liu B, Wang H. QTL: a new ultra-lightweight block cipher. Microprocessors and Microsystems, 2016, 45: 45–55
12. Kwon J, Lee B, Lee J, Moon D. FPL: white-box secure block cipher using parallel table look-ups. In: Proceedings of Cryptographers’ Track at the RSA Conference. 2020, 106–128
13. Li L, Liu B, Zhou Y, Zou Y. SFN: a new lightweight block cipher. Microprocessors and Microsystems, 2018, 60: 138–150
14. Zhang W, Bao Z, Lin D, Rijmen V, Yang B, Verbauwhede I. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences, 2015, 58(12): 1–15