

一种基于 Feistel 结构和 WTS 的分组密码

时阳阳, 黄玉划, 陈帮春

(南京航空航天大学 计算机科学与技术学院, 江苏 南京 210016)

摘要: AES 为新的数据加密标准, 通过研究分组密码算法加密的整体结构和 AES 加密算法, 文中设计了一种基于 Feistel 结构和 WTS 策略的分组密码算法 FWTS。FWTS 采用 Feistel 结构, 轮函数借鉴 AES 的 WTS 策略, 分组长度为 256 bits, 密钥长度为 128 bits, 192 bits, 256 bits。通过依赖性测试表明, FWTS 算法 4 轮充分满足雪崩效应、严格雪崩准则和完备性。通过不可能差分分析, FWTS 算法的 6 轮不可能差分所需的时间复杂度要大于 AES 算法的 6 轮不可能差分的时间复杂度。FWTS 算法的安全性不低于 AES 算法。通过效率测试表明 FWTS 的加密效率要高于 AES。

关键词: AES; Feistel 结构; WTS 策略; 依赖性测试; 不可能差分分析

中图分类号: TN918.1

文献标识码: A

文章编号: 1673-629X(2014)08-0126-05

doi: 10.3969/j.issn.1673-629X.2014.08.029

A Block Cipher Based on Feistel Structure and WTS

SHI Yang-yang, HUANG Yu-hua, CHEN Bang-chun

(College of Computer Science & Technology, Nanjing University of Aeronautics & Astronautics,
Nanjing 210016, China)

Abstract: AES is the advanced data encryption standard. A block cipher based on Feistel structure and WTS strategies named FWTS is designed by studying the overall structure of the block cipher and AES algorithm. FWTS algorithm adopts Feistel structure and its round function refers to WTS strategies of AES encryption. The block size of FWTS is 256 bits, the key size is 128 bits, 192 bits, 256 bits. Dependence test shows that FWTS algorithm fully satisfies the avalanche effect with 4-round. Impossible differential analysis with 6-round presents that FWTS algorithm has a higher time complexity than AES. The security of FWTS algorithm is not lower than AES. The efficiency test shows that FWTS algorithm has a higher efficiency compared with AES.

Key words: AES; Feistel structure; WTS strategies; dependence test; impossible differential analysis

0 引言

现代密码学的研究, 是开始于 20 世纪 70 年代“密码学新方向”^[1]的发表和美国数据加密标准 DES^[2]的实施。从这个时候起, 密码学开始运用到民用研究, 这种转变也导致了密码学的空前发展。自此密码学研究的两个基本方向: 一是秘密密钥分组密码其以 DES 为代表; 二是公开密钥密码其以 RSA^[3]为代表。由于具有很好的扩展性、实用性、易于软硬件实现和标准化等优点, 分组密码通常是信息和网络安全中实现数据加密、消息鉴别、认证及密钥管理的核心算法。

1997-2000 年高级加密标准 AES 在世界范围内进行公开透明的征集和选评。这次活动是分组密码发展过程中的标志性事件。2001 年美国国家标准技术

研究所(NIST)正式公布采用 Rijndael^[4]作为新的加密标准 AES。Rijndael 算法是从五个 AES 候选算法(MARS^[5]、Rijndael、Serpent^[6]、Twofish^[7]、RC6^[8])中经过分析和测试选出来的。从此许多密码产品都不再使用 DES 或其变形算法, 而逐渐转向使用 AES。我国在国家“863”计划中也指定 AES 的分析及其应用研究为分组密码的重点研究方向之一。

分组密码的整体结构影响分组密码的轮数和算法在软硬件上实现的性能。比较常见的有 Feistel 结构、SP 结构和 Lai-Massey 结构。AES 的设计思想是宽轨迹策略(WTS), 其分组长度为 128 比特, 密钥长度为 128、192、256 比特三种可选, 迭代次数分别为 10、12、14 轮。文中通过研究分组密码的整体结构, 在 AES 的

收稿日期: 2013-10-07

修回日期: 2014-01-13

网络出版时间: 2014-04-24

基金项目: 航空科学基金(20081952014, 20085552021); 区域光纤通信网与新型光通信系统国家重点实验室开放课题(2008SH06); 南京航空航天大学基本科研业务费专项科研项目(NS2010097)

作者简介: 时阳阳(1986-), 女, 河南舞钢人, 硕士研究生, 研究方向为信息安全; 黄玉划, 博士, 副教授, 硕士生导师, 研究方向为信息安全。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140424.0839.085.html>

基础上设计一种基于 Feistel 结构和 WTS 策略的分组密码算法 FWTS。FWTS 采用 Feistel 结构,轮函数借鉴 AES 的 WTS 策略,在保证安全性的基础上使 FWTS 算法的效率高于 AES。因为 FWTS 的轮函数借鉴 AES 的 WTS 策略,所以对于所有 AES 能抵抗的攻击方法,FWTS 也能抵抗。

1 Feistel 结构

分组密码的一个重要特征是算法的整体结构,算法选用的整体结构对于分组密码迭代轮数的选择、硬件实现性能都有非常大的影响。整体结构的研究多采用可证明安全理论的方法,研究它们在一定假设下的伪随机性和超伪随机性,研究它们对差分、线性等分析方法的抵抗能力。文中新设计的分组密码是面向 32 位机,整体结构采用 Feistel 结构,下面简单介绍 Feistel 结构的基本知识。

Feistel 密码是由 H. Feistel 设计的一种迭代密码,并因为 DES 的使用而流行。许多分组密码都采用了这种结构,例如 RC5、GOST、CAST、Lucifer、Camellia、LOKI 和 E2 等。

对于分组长度是 $2n$ 比特的输入 $L_0 || R_0$, L_0 和 R_0 都是 n 比特,一轮 Feistel 结构通过如下计算输出 $L_1 || R_1$:

$$\begin{aligned} L_1 &= R_0 \\ R_1 &= L_0 \oplus F(R_0, K_0) \end{aligned}$$

这里的 \oplus 表示两个比特串的异或, $F: \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ 是轮函数。

由于 Feistel 结构的优点是“加解密相似”,所以设计轮函数时可以不考虑解密;而且关于 Feistel 结构的安全性的研究成果比较多,所以采用 Feistel 结构来设计新算法安全性会更加有保证一些。

在 Feistel 结构的安全性分析方面,文献[9-11]等给出了 Feistel 结构抵抗差分分析、线性分析的量化结果。1988 年 Luby 和 Rackoff 证明了如果轮函数是伪随机的,则 3 轮 Feistel 结构是伪随机的,4 轮 Feistel 结构是超伪随机的。近几年相关的工作还在不断推进。

2 FWTS 算法

2.1 密钥扩展算法

FWTS 算法采用 AES 的密钥扩展算法,密钥长度以 4 字节的字为单位来表示,密钥长度可记为 N_k , $N_k = 4, 6$ 或 8 。然后由 N_k 个原始密钥字扩展为 $N_b * N_r$ 个字 (N_r 为加密轮数, N_b 为分组长度),扩展算法的伪代码如下:

```
KeyExpansion(byte key[4 *  $N_k$ ]), word w[ $N_b * N_r$ ,  $N_k$ ]  
begin
```

```
word temp  
 $i = 0$   
while(  $i < N_k$  )  
   $w[i] = \text{word}(\text{key}[4 * i], \text{key}[4 * i + 1], \text{key}[4 * i + 2], \text{key}[4 * i + 3])$   
   $i = i + 1$   
end while  
 $i = N_k$   
while(  $i < N_b * N_r$  )  
   $\text{temp} = w[i - 1]$   
  if(  $i \bmod N_k = 0$  )  
     $\text{temp} = \text{SubWord}(\text{RotWord}(\text{temp})) \text{ xor } \text{Rcon}[i / N_k]$   
  else if(  $N_k > 6$  and  $i \bmod N_k = 4$  )  
     $\text{temp} = \text{SubWord}(\text{temp})$   
  end if  
   $w[i] = w[i - N_k] \text{ xor } \text{temp}$   
   $i = i + 1$   
end while  
end
```

其中 SubWord() 输入为一个 4 字节的字,对子节中的每个字节做 S 盒变换,变换后的 4 个字节所组成的字为 SubWord() 的输出。

RotWord() 的返回值为一个 4 字节的字,它将输入的 4 个字节以字节为单位循环左移一个字节。Rcon[] 是一个常数, $\text{Rcon}[j] = (c_j, 00, 00, 00)$, $j \geq 1$:

$$\begin{aligned} c_1 &= 01 \\ c_j &= \{02\} \odot c_{j-1}, j \geq 1 \end{aligned}$$

2.2 算法加密过程

FWTS 算法的分组长度为 256 比特,密钥长度有三种分别为 128、192、256 比特,迭代轮数为 15 轮。该算法采用 Feistel 结构,以 32 比特的字为处理单位,256 比特的分组分为左右两个分支 L_i, R_i , 然后进行 15 轮完全相同的运算。加密过程为:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus F(R_{i-1}, K_i) \end{aligned}$$

其中轮函数 F 采用 AES 的加密流程,轮函数 F 为:

```
SubBytes(  $R_{i-1}$  )  
ShiftRows(  $R_{i-1}$  )  
MixColumns(  $R_{i-1}$  )  
AddRoundKey(  $R_{i-1}, K_i$  )
```

SubBytes(): 是一个关于字节的非线性变换,它将加密过程中的每个字节非线性地变换为另外一个字节。FWTS 算法的字节变换中的 S 盒采用 AES 加密中的 S 盒。

ShiftRows(): 行位移变换是对一个状态的每一行循环左移不同的位移量。第 0 行不移位,第一行循环左移一个字节,第二行循环左移两个字节,第三行循环

左移三个字节。即：

$$\begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{bmatrix} \xrightarrow{\text{ShiftRows}()} \begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{11} & s_{12} & s_{13} & s_{10} \\ s_{22} & s_{23} & s_{20} & s_{21} \\ s_{33} & s_{30} & s_{31} & s_{32} \end{bmatrix}$$

MixColumns() :列混合变换是对一个状态按列依次进行变换,即它将一个状态的每一列视为有限域 GF(2⁸)上的一个多项式。即：

$$\begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{bmatrix} \xrightarrow{\text{MixColumns}()} \begin{bmatrix} s'_{00} & s'_{01} & s'_{02} & s'_{03} \\ s'_{10} & s'_{11} & s'_{12} & s'_{13} \\ s'_{20} & s'_{21} & s'_{22} & s'_{23} \\ s'_{30} & s'_{31} & s'_{32} & s'_{33} \end{bmatrix}$$
$$\text{其中,} \begin{bmatrix} s'_{0j} \\ s'_{1j} \\ s'_{2j} \\ s'_{3j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} s_{0j} \\ s_{1j} \\ s_{2j} \\ s_{3j} \end{bmatrix}, 0 \leq j \leq 3$$

AddRoundKey() :子密钥加变换是简单的将一个轮子密钥按位异或到一个状态上。轮子密钥的长度为 4 个字(128 比特),一个字为 4 个字节(32 比特)。轮子密钥按顺序取自扩展密钥。扩展密钥的长度是 N_b * N_r 个字。

$$\begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{bmatrix} \xrightarrow{\text{AddRoundKey}()} \begin{bmatrix} s'_{00} & s'_{01} & s'_{02} & s'_{03} \\ s'_{10} & s'_{11} & s'_{12} & s'_{13} \\ s'_{20} & s'_{21} & s'_{22} & s'_{23} \\ s'_{30} & s'_{31} & s'_{32} & s'_{33} \end{bmatrix}$$

(s'_{0j}, s'_{1j}, s'_{2j}, s'_{3j}) = (s_{0j}, s_{1j}, s_{2j}, s_{3j}) ⊕ (k_{0j}, k_{1j}, k_{2j}, k_{3j}), 0 ≤ j ≤ 3。其中 (k_{0j}, k_{1j}, k_{2j}, k_{3j}) 表示扩展密钥中的第 r * N_b + j 个字, 0 ≤ j ≤ N_r。

经过 15 轮加密后由 (L₁₅, R₁₅) 级联得到密文比特串。

3 效率测试和安全性分析

3.1 效率测试

作者对 FWTS 和 AES 算法用 C 语言编程测试,且在 Intel(R), CPU: Core(TM), i3, 主频 2.93 GHz, 内存

4 GB 的测试平台下进行测试,两种算法的效率见表 1 和表 2。

表 1 AES 和 FWTS 算法不含密钥扩展的效率

密钥长度/bit	AES/(Mb/s)	FWTS(Mb/s)
128	583.4	722.35
192	537.36	722.35
256	474.43	722.35

表 2 AES 和 FWTS 算法含密钥扩展的效率

密钥长度/bit	AES/(Mb/s)	FWTS(Mb/s)
128	301.74	457.80
192	280.71	447.40
256	240.06	414.37

通过表 1 可知,由于 FWTS 的加密轮数建议是 15 轮,所以在不含密钥加密时,加密效率和密钥长度无关且都大于 AES 算法的加密效率。通过表 2 可以得出,在包含密钥扩展时,FWTS 的加密效率也是高于 AES。

3.2 依赖性测试

密码算法的依赖性测试^[12]包括算法的完备性、雪崩效应、严格雪崩准则等方面。完备性是指加密函数输出的任一比特都与输入的全部比特有关;雪崩效应是指输入任一比特的改变都应造成输出平均半数比特的改变;严格雪崩准则是指输入任一比特的改变都应造成输出每一比特以 1/2 的概率发生改变。

(1)GF(2)上的{0,1}ⁿ表示所有 n 比特向量 x = (x₁, x₂, ..., x_n)的集合, w(x) 表示向量 x 的汉明距离即元素 1 的个数, x⁽ⁱ⁾ 表示向量 x 的第 i 位取反。

(2)若算法 f 是完备的,则对于密码算法 f: {0,1}ⁿ → {0,1}^m, 每一比特的输出都与每一比特输入有关, 即 ∀ i = 1, 2, ..., n 和 ∀ j = 1, 2, ..., m, ∃ x ∈ {0,1}ⁿ 使得 (f(x⁽ⁱ⁾))_j ≠ (f(x))_j。

(3)若密码算法 f 满足严格雪崩准则,则密码算法 f 输入序列的任一比特发生改变,使得算法输出序列中每一比特发生改变的概率为 0.5, 即 ∀ i = 1, 2, ..., n 和 ∀ j = 1, 2, ..., m, Pr (f(x⁽ⁱ⁾))_j ≠ (f(x))_j = 0.5。

(4)若密码算法 f 满足雪崩效应,则改变密码算法 f 的任意 1-bit 输入,使得输出序列平均有一半发生变化, 即: 2⁻ⁿ ∑_{x ∈ {0,1}ⁿ} w(f(x⁽ⁱ⁾) - f(x)) = m/2, ∀ i = 1, 2, ..., n。

(5) 距离阵 A [n] [m+1] 中的元素 a_{ij} 表示第 i 位输入改变时导致 j 位输出改变的向量个数, 即 a_{ij} = # | x ∈ {0,1}ⁿ | w(f(x⁽ⁱ⁾) - f(x)) = j。其中 # 表示选取的样本总数, 下同。

(6) 依赖阵 B [n] [m] 中的元素 b_{ij} 表示第 i 位输入改变时导致第 j 位输出改变的向量个数, 即 b_{ij} = # | x

- $\in \{0,1\}^n \mid (f(x^{(i)}))_j \neq (f(x))_j\}$ 。
- (7)数值 $d_c = 1 - \#\{(i,j) \mid b_{ij} = 0\} / (nm)$ 称为完备度。
- (8)数值 $d_a = 1 - \sum_{i=1}^n \left| \sum_{j=1}^m 2ja_{ij} / \#X - m \right| / (nm)$ 称为算法的雪崩效应度。
- (9)数值 $d_{sa} = 1 - \sum_{i=1}^n \sum_{j=1}^m |2b_{ij} / \#X - 1| / (nm)$ 称

表 3 FWTS 算法的依赖性测试结果

轮数	雪崩效应度 d_a	严格雪崩效应度 d_{sa}	完备度 d_c	输出改变 变位数	输出改变的 平均位数	输出改变 变概率	输出改变的 平均概率
3	0.813 267	0.802 534	0.812 5	36 ~ 169	104.365 7	0 ~ 0.570 8	0.500 461
4	0.999 239	0.991 720	1.000 000	81 ~ 170	128.016 921	0.479 5 ~ 0.522 2	0.500 461
5	0.999 522	0.991 997	1.000 000	89 ~ 168	127.999 231	0.478 4 ~ 0.520 0	0.499 997
15	0.999 517	0.992 017	1.000 000	87 ~ 168	128.001 877	0.478 9 ~ 0.523 5	0.500 007

由表 3 可以得出,FWTS 算法从第 4 轮开始充分满足完备性、雪崩效应、严格雪崩准则的基本要求。AES 算法从第 3 轮开始充分满足完备性、雪崩效应、严格雪崩准则的基本要求。虽然 FWTS 算法与 AES 相比晚了一轮开始满足上述各项指标,但是 FWTS 的建议轮数是 15 轮,所以综合考虑 FWTS 的各项指标还是都与 AES 相当。

3.3 不可能差分分析

文献[13]中提出了不可能差分密码分析方法,它利用的是概率为 0 的特征(或差分),其基本思想是排除那些导致概率为 0 的特征或差分的候选密钥。对于一条概率为 0 的差分路径,当用正确密钥解密密文对时,不会得到符合该路径的差分;如果用猜测密钥解密密文对,得到符合该差分的路径,那么该密钥猜测值是错误的;那么筛去所有的错误猜测值,剩下的就是需要恢复的正确密钥。

Knudsen 在研究 DEAL 算法的安全性时发现,如果 Feistel 结构密码的轮函数是双射,则算法存在天然 5 轮不可能差分,从而对 6 轮密码的安全性构成威胁。不可能差分密码分析是当前对简化轮数的 Rijndael 算法和 Camellia 算法最有效的攻击手段。

不可能差分密码分析的第一步通常是用中间相错的方法构造不可能差分。

- 不可能差分密码分析的基本过程^[14]为:
- 步骤 1:寻找 $r-1$ 轮不可能差分 $a_0 \xrightarrow{r-1} a_{r-1}$;
- 步骤 2:选择满足差分为 a_0 的明文对 $(P, P \oplus a_0)$, 并进行 r 轮加密,所得密文记为 C 和 C^* ;
- 步骤 3:猜测第 r 轮的轮密钥 K_r 的所有可能值,对每一个猜测的密钥分别对 C 和 C^* 解密一轮,所得中间值不妨记为 D 和 D^* ;判断 $D \oplus D^* = a_{r-1}$ 是否成立,若成立则对应的猜测值一定是错误密钥;
- 步骤 4:重复上述步骤,直到密钥唯一确定为止。

为算法的严格雪崩准则度。

如果加密算法满足完备度 $d_c = 1$,雪崩效应度 $d_a \approx 1$,严格雪崩效应度 $d_{sa} \approx 1$,则可以认为该算法具有良好的依赖性。

测试 FWTS 算法的依赖性,输入样本个数取 10 000 个,密钥长度 128 bit,FWTS 明文长度 256 bit,得到的测试结果见表 3。

通过分析 FWTS 的加密过程,假设输入差分为 $(\alpha, 0)$,则根据 Feistel 结构的特点可以知道,第 1 轮的输出差分为 $(0, \alpha)$ 。由于轮函数采用 AES,则当输入存在差分 α 时,输出一定存在非零差分 β ,从而第 2 轮的输出差分为 (α, β) ,其中 $\beta \neq 0$ 。同样,当 AES 算法输入存在非零差分 β 时,输出一定存在非零差分 γ ,从而第 3 轮的输出差分为 $(\beta, \alpha \oplus \gamma)$ 。从解密方向看差分的传播,假设第 5 轮的输出差分为 $(0, \alpha)$,则第 4 轮的输出差分为 $(\alpha, 0)$,从而第 3 轮的输出差分为 (φ, α) , $\varphi \neq 0$ 。若 $(\alpha, 0) \rightarrow (0, \alpha)$ 是一条可能的差分,则在特定条件下, $(\beta, \alpha \oplus \gamma) = (\varphi, \alpha)$,从而 $\alpha \oplus \gamma = \alpha$ 即 $\gamma = 0$,这与 γ 非零矛盾。所以可以找到一个 5 轮的不可能差分即: $(\alpha, 0) \xrightarrow{5} (0, d)$ (见图 1)。其中 $\alpha \neq 0$,最后一轮包括数据的左右交换。

利用上述 5 轮不可能差分路径可以计算求出第 6 轮密钥的时间复杂度和数据复杂度。

选取 2^{64} 个明文 $P_i = (L_i, R)$, $i = 1, 2, \dots, 64$, 它们的右半部分相同, $C_i = (Z_i, W_i)$ 是对应的密文,计算 $L_i \oplus W_i$, 对于 $i \neq j$, 寻找 $L_i \oplus W_i = L_j \oplus W_j$ 。可以得到 2^{63} 个匹配。设 $\alpha = L_i \oplus L_j = W_i \oplus W_j$ 。

第 6 轮的子密钥可能值有 2^{128} 个,任取一个 K_6 ,用 K_6 对找到的所有匹配的密文解密,如果得到的第 5 轮的差分存在 0α ,则此密钥是错误的。对正确的子密钥 K_6 ,第 5 轮的差不存在 0α ,但对于错误的子密钥值,每个匹配对产生错误密钥值的概率为 2^{-64} ,因此,在对 2^{63} 个匹配试验完后,大约有一半子密钥被放弃。反复攻击 128 次,所剩的值中必有正确的子密钥。攻击需要 $128 \times 2^{64} = 2^{71}$ 个选择明文, $(2^{128} + 2^{127} + \dots + 2 + 1) \times 2^{64} \approx 2^{193}$ 次加密。

表 4 列出了 AES 和 FWTS 算法 6 轮不可能差分分析所需的时间复杂度和数据复杂度(AES 算法的 6 轮

不可能差分分析参考文献[15])。

表 4 AES 和 FWTS 算法 6 轮不可能差分分析结果

算法	时间复杂度	数据复杂度
AES	2^{104}	$2^{5.5}$
FWTS	2^{193}	2^{71}

通过表 4 可以看出,AES 和 FWTS 所需的数据复杂度相差不大,但是二者所需的时间复杂度相差比较大,也就是说相同情况下攻击 FWTS 所需的时间要比 AES 长。

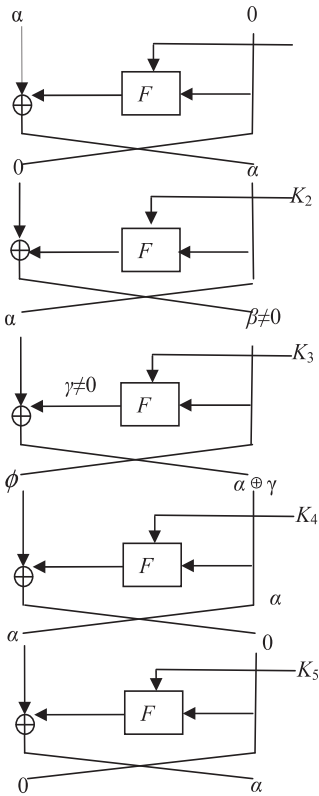


图 1 FWTS 密码的 5 轮不可能差分

4 结束语

文中通过研究分组密码的设计原理和整体结构,设计了一种新的分组密码算法 FWTS。FWTS 采用 Feistel 结构,轮函数借鉴 AES 的 WTS 策略。因为 FWTS 的轮函数借鉴 AES 的 WTS 策略,所以 FWTS 算法的安全性有所保障。通过依赖性测试,FWTS 算法从 4 轮开始充分满足雪崩效应、严格雪崩准则。通过 FWTS 和 AES 的 6 轮不可能差分分析表明,FWTS 的 6 轮不可能差分所需的数据复杂度基本和 AES 一样,但是 FWTS 算法的 6 轮不可能差分所需的时间复杂度要大于 AES。安全性测试表明 FWTS 算法的安全性不低于 AES 的安全性。通过效率测试表明,FWTS 算法的加密效率要高于 AES。然而 AES 的 S 盒采用的生成

多项式是不可约多项式,但不是本原多项式。用本原多项式取代算法中的生成多项式,设计新算法是下一步的工作。

参考文献:

[1] Diffie W, Hellman M E. New directions in cryptography [J]. IEEE Transactions on Information Theory, 1977, IT-22 (6): 74-84.

[2] NBS. Data Encryption Standard [S]. Washington D C: National Bureau of Standards, 1977.

[3] Pointcheval D. RSA public-key encryption [M]//Encyclopedia of cryptography and security. US: Springer, 2011.

[4] Wang Q, Gu D, Rijmen V, et al. Improved impossible differential attacks on large-block rijndael [C]//Proc of ICISC 2012. Berlin: Springer, 2013: 126-140.

[5] Mohan H S, Reddy A R. Performance analysis of AES and MARS encryption algorithms [J]. International Journal of Computer Science Issues, 2011, 8(4): 363-368.

[6] Chen Wenlue, Li Boli, Hu Zhihua. Rectangle algebraic attack of serpent encryption algorithm [C]//Proc of 2010 international symposium on intelligence information processing and trusted computing. Huanggang: IEEE, 2010: 573-576.

[7] Landge I, Bharmal T, Narwankar P. Encryption and decryption of data using twofish algorithm [J]. World Journal of Science and Technology, 2012, 2(3): 157-161.

[8] Kim G H, Kim J N, Cho G Y. An improved RC6 algorithm with the same structure of encryption and decryption [C]//Proc of 11th international conference on advanced communication technology. Phoenix Park: IEEE, 2009: 1211-1215.

[9] Nyberg K, Kundsén L. Provable security against differential cryptanalysis [J]. Journal of Cryptology, 1995, 1 (8): 156-168.

[10] Knudsen L R. Practically secure Feistel ciphers [M]//Fast software encryption. Berlin: Springer, 1994.

[11] Kanda M. Practical security evaluation against differential and linear cryptanalyses for Feistel ciphers with SPN round function [J]. LNCS, 2001, 2012: 324-338.

[12] Serf P. The degrees of completeness, of avalanche effect, and of strict avalanche criterion for mars, rc6, rijndael, serpent, and twofish with reduced number of rounds [EB/OL]. 2000-02-03. <http://www.cosic.esat.kuleuven.ac.be/nessie/reports/phase1/sagwp3-003.pdf>.

[13] Biham E, Biryukov A, Shamir A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials [C]//Proc of Eurocrypt '99. [s. l.]: [s. n.], 1999: 12-23.

[14] 刘亚. 若干分组密码不可能差分分析与代数分析方法的研究 [D]. 上海: 上海交通大学, 2013.

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

一种基于Feistel结构和WTS的分组密码


作者:

[时阳阳](#), [黄玉划](#), [陈帮春](#), [SHI Yang-yang](#), [HUANG Yu-hua](#), [CHEN Bang-chun](#)

作者单位:

[南京航空航天大学 计算机科学与技术学院, 江苏 南京, 210016](#)

刊名:

[计算机技术与发展](#) 

英文刊名:

[Computer Technology and Development](#)

年, 卷(期):

2014(8)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjtz201408029.aspx