

HIGHT 算法的积分攻击

郭建胜^{1,2}, 崔竞一¹, 潘志舒³, 刘翼鹏¹

(1. 解放军信息工程大学三院, 河南 郑州 450001; 2. 信息保障技术重点实验室, 北京 100072; 3. 西安卫星测控中心, 陕西 西安 710043)

摘 要: 对轻量级分组密码算法 HIGHT 在积分攻击方法下的安全性进行了研究。首先纠正了现有研究成果在构造区分器时的不当之处, 重新构造了 HIGHT 算法的 11 轮积分区分器, 并构造了相应高阶积分扩展下的 17 轮区分器; 其次利用所构造的 17 轮区分器, 结合“时空折中”原理对 25 轮 HIGHT 算法进行了积分攻击; 最后对攻击算法的复杂度进行了分析, 攻击算法需要的数据复杂度为 $2^{62.92}$, 时间复杂度为 $2^{66.20}$, 空间复杂度为 2^{119} 。分析结果表明, 所给出的攻击算法的攻击轮数和时间复杂度要优于现有研究结果。

关键词: 密码分析; 分组密码; 积分攻击; HIGHT 算法

中图分类号: TN918.1

文献标识码: A

Integral attack on HIGHT block cipher

GUO Jian-sheng^{1,2}, CUI Jing-yi¹, PAN Zhi-shu³, LIU Yi-peng¹

(1. The Third Department, The PLA Information Engineering University, Zhengzhou 450001, China;

2. Science and Technology on Information Assurance Laboratory, Beijing 100072, China;

3. Xi'an Satellite Control Center, Xi'an 710043, China)

Abstract: The security of HIGHT block cipher under integral attack was studied. Firstly, the flaw in the existing results on building the distinguisher was corrected. And a new 11-round integral distinguisher of HIGHT was built. Based on this new distinguisher, a 17-round multiple-integral distinguisher was built. By using the 17-round distinguisher, 25-round integral attack on HIGHT was proposed based on the principle of time memory trade-off, with the data, time and memory complexity of $2^{62.92}$, $2^{66.20}$ and 2^{119} respectively. The results show that the attack was better than results before on the number of round and time complexity.

Key words: cryptanalysis, block cipher, integral attack, HIGHT block cipher

1 引言

HIGHT 算法是由 Hong 等^[1]在 CHES 2006 上提出的轻量级分组密码, 其采用一种广义 Feistel 结构, 轮函数输入和输出均为 8 bit, 规模很小, 且没有用 S 盒, 只使用循环移位、异或和模加操作, 在 8 位处理器的环境中表现出很好的效率。子密钥是在加密过程中通过密钥扩展算法得到的, 密钥寄存器只需要存储 128 bit 的主密钥。

针对 HIGHT 的分析方面, 最初由设计者给出了 HIGHT 算法的差分分析、线性分析、截段差分分析、不可能差分分析、积分分析等结果^[1]。其积

分分析构造了 12 轮的积分区分器, 并对 16 轮 HIGHT 算法进行了攻击。2009 年, 文献[2]指出了算法设计者在构造积分区分器时的错误, 利用高阶积分扩展构造了 17 轮的积分区分器, 并利用密钥扩展算法攻击了 22 轮 HIGHT 算法。在 ICISC 2010 上, Koo 等^[3]在相关密钥条件下给出了全轮 HIGHT 算法的攻击结果。在单密钥条件下, Hong 等^[4]在 ICISC 2011 上利用 Biclique 攻击给出了全轮 HIGHT 算法的攻击结果, 但其时间复杂度较高。Chen 等^[5]在 2012 年非洲密码年会上提出了 HIGHT 算法不可能差分分析的相关结论。在 2015 年, 由 Igarashi 等^[6]提出了对 19 轮 HIGHT 算法的中间相遇攻击结

收稿日期: 2015-01-27; 修回日期: 2016-06-04

基金项目: 中国博士后科学基金资助项目 (No.2014M562582)

Foundation Item: China Postdoctoral Science Foundation (No.2014M562582)

果。同时，HIGHT 算法也有在故障分析下安全性的相关研究成果^[7,8]。

本文对 HIGHT 算法在积分攻击下的安全性进行了研究。首先，对文献[2]所构造的积分区分器进行了分析，指出了文献[2]在构造区分器时的不当之处，重新构造了 HIGHT 算法的 11 轮积分区分器，并构造了相应高阶积分扩展下的 17 轮区分器；然后，根据高阶积分扩展构造的 17 轮区分器，攻击了 25 轮 HIGHT 算法。其中，根据“时空折中”的原理，利用存储空间分担了部分计算时间，降低了整个攻击算法的计算复杂度。攻击 25 轮 HIGHT 算法的数据复杂度、时间复杂度和空间复杂度分别为 $2^{62.92}$ 、 $2^{66.20}$ 和 2^{119} 。攻击轮数和时间复杂度都要优于文献[2]与文献[11]的结果。

2 相关知识

2.1 HIGHT 算法结构简介

HIGHT 算法采用了具有 8 分支的广义 Feistel 结构。其分组长度为 64 bit，密钥长度为 128 bit，加密轮数为 32 轮。每一轮包含 2 个不同的 F 函数 ($F_0: \{0,1\}^8 \rightarrow \{0,1\}^8$, $F_1: \{0,1\}^8 \rightarrow \{0,1\}^8$)、异或运算 \oplus 、模 2^8 加运算 \boxplus 以及内部位置变换。其 F 函数为： $F_0(x) = x \lll 1 \oplus x \lll 2 \oplus x \lll 7$ ， $F_1(x) = x \lll 3 \oplus x \lll 4 \oplus x \lll 6$ 。设 64 bit 明文为 $(P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0)$ ，经过 32 轮算法后变换成 64 bit 密文 $(C_7, C_6, C_5, C_4, C_3, C_2, C_1, C_0)$ 。具体结构如图 1 所示。

HIGHT 算法具体加密流程如下。

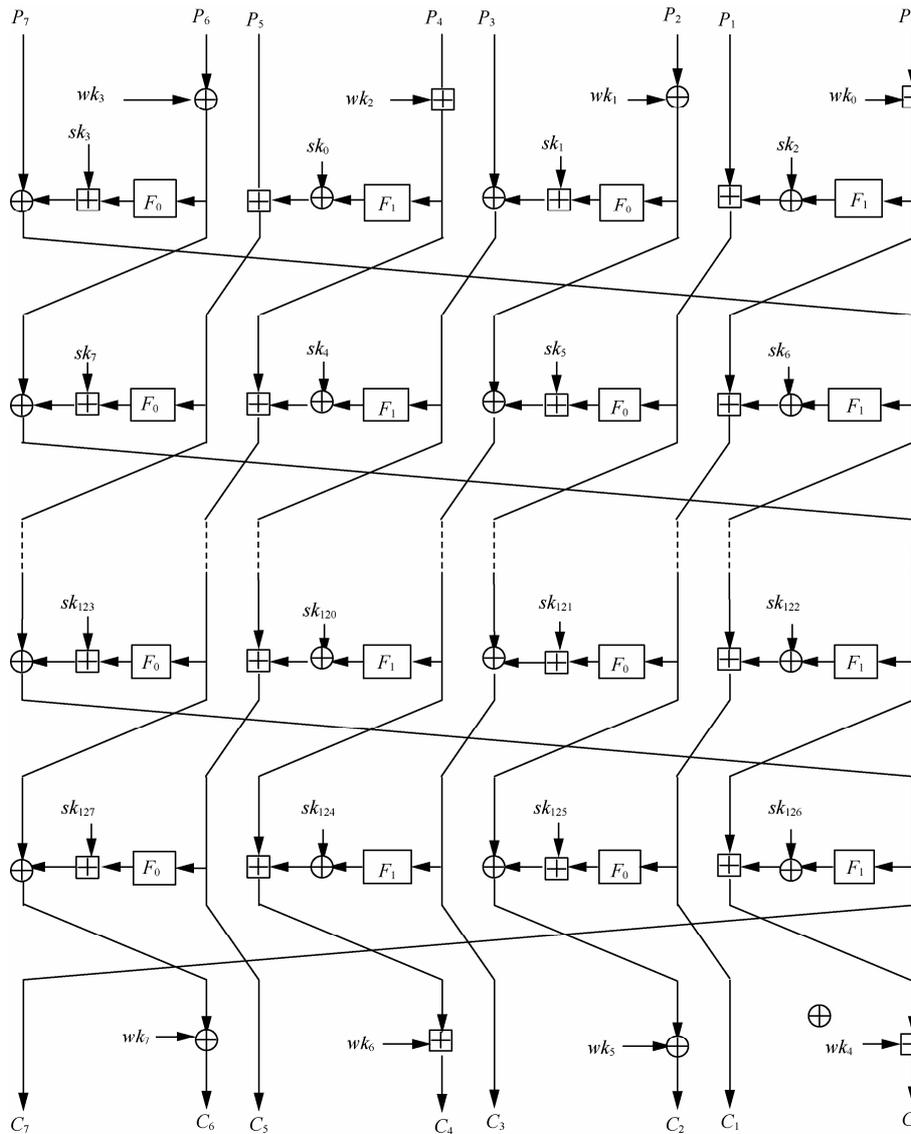


图 1 HIGHT 算法加密流程

1) 初始白化密钥加变换

$$X_{0,7} = P_7, X_{0,6} = P_6 \oplus wk_3, X_{0,5} = P_5, X_{0,4} = P_4 \boxplus wk_2$$

$$X_{0,3} = P_3, X_{0,2} = P_2 \oplus wk_1, X_{0,1} = P_1, X_{0,0} = P_0 \boxplus wk_0$$

2) 轮变换

$$(X_{i-1,7}, X_{i-1,6}, X_{i-1,5}, X_{i-1,4}, X_{i-1,3}, X_{i-1,2}, X_{i-1,1}, X_{i-1,0})$$

为第 i 轮的输入, 其中, $i=1,2,\dots,31$, 则输出为

$$X_{i,7} = X_{i-1,6}, X_{i,6} = X_{i-1,5} \boxplus (F_1(X_{i-1,4}) \oplus sk_{4i-4})$$

$$X_{i,5} = X_{i-1,4}, X_{i,4} = X_{i-1,3} \oplus (F_0(X_{i-1,2}) \boxplus sk_{4i-3})$$

$$X_{i,3} = X_{i-1,2}, X_{i,2} = X_{i-1,1} \boxplus (F_1(X_{i-1,0}) \oplus sk_{4i-2})$$

$$X_{i,1} = X_{i-1,0}, X_{i,0} = X_{i-1,7} \oplus (F_0(X_{i-1,6}) \boxplus sk_{4i-1})$$

$$(X_{31,7}, X_{31,6}, X_{31,5}, X_{31,4}, X_{31,3}, X_{31,2}, X_{31,1}, X_{31,0})$$

为第 32 轮的输入, 则对应的输出为

$$X_{32,7} = X_{31,0}, X_{32,6} = X_{31,7} \oplus F_0(X_{31,6}) \boxplus sk_{127}$$

$$X_{32,5} = X_{31,6}, X_{32,4} = X_{31,5} \boxplus (F_1(X_{31,4}) \oplus sk_{124})$$

$$X_{32,3} = X_{31,4}, X_{32,2} = X_{31,3} \oplus F_0(X_{31,2}) \boxplus sk_{125}$$

$$X_{32,1} = X_{31,2}, X_{32,0} = X_{31,1} \boxplus (F_1(X_{31,0}) \oplus sk_{126})$$

3) 结尾白化密钥加变换

$$C_7 = X_{32,7}, C_6 = X_{32,6} \oplus wk_7$$

$$C_5 = X_{32,5}, C_4 = X_{32,4} \boxplus wk_6$$

$$C_3 = X_{32,3}, C_2 = X_{32,2} \oplus wk_5$$

$$C_1 = X_{32,1}, C_0 = X_{32,0} \boxplus wk_4$$

HIGHT 算法的密钥扩展算法由两部分组成。第一部分是常数生成部分, 利用线性反馈移位寄存器生成 128 个 7 bit 常数 $\delta_0, \delta_1, \dots, \delta_{127}$; 第二部分为白化密钥和子密钥生成部分, 通过主密钥 MK 与第一部分生成的常数生成白化密钥和子密钥。首先将主密钥 MK 化分为 16 byte, 即 $MK = (MK_{15}, MK_{14}, \dots, MK_0)$ 。

白化密钥生成部分: $wk_i = MK_{i+12} (i=0,1,2,3)$,
 $wk_i = MK_{i-4} (i=4,5,6,7)$ 。

子密钥生成部分: $sk_{16i+j} = MK_{(j-i) \bmod 8} \boxplus \delta_{16i+j}$,
 $sk_{16i+j+8} = MK_{(j-i) \bmod 8+8} \boxplus \delta_{16i+j+8} (0 \leq i, j \leq 7)$ 。

2.2 积分攻击

积分攻击是 Knudsen 等^[9]总结提出的一种分组密码选择明文攻击方法, 自提出以来, 其得到了越来越广泛的关注, 该攻击方法被应用于许多算法的安全性分析中, 例如 AES^[10]、LBlock^[11]、E2^[12]和 MIBS^[13]等。

积分攻击就是选择特定形式的明文进行加密, 再对所得密文求和 (积分), 通过积分值的不随机性将密码算法与随机置换区分开。在构造积分区分器时, 需要定义一些符号。

定义 1^[9,14] 一些特殊形式的集合

1) 活跃集: 若对任意的 $0 \leq i < j \leq 2^n - 1$, 都有 $x_i \neq x_j$, 则集合 $\{x_i | x_i \in F_{2^n}, 0 \leq i \leq 2^n - 1\}$ 是活跃集, 记为 A 。

2) 稳定集: 若对任意的 $0 < i \leq 2^n - 1$, 都有 $x_i = x_0$, 则集合 $\{x_i | x_i \in F_{2^n}, 0 \leq i \leq 2^n - 1\}$ 是稳定集, 记为 C 。

3) 平衡集: 若 $\bigoplus_{i=0}^{2^n-1} x_i = 0$, 则集合 $\{x_i | x_i \in F_{2^n}, 0 \leq i \leq 2^n - 1\}$ 是平衡集, 记为 B 。

此外, 记不能确定是否平衡的集合为 U 。

这些集合之间的运算遵循一些基本原则。

性质 1^[9,14] 不同集合之间满足如下性质。

1) 集合 A 通过双射 (如密钥加) 后, 仍是集合 A ; 集合 C 通过双射后, 仍是集合 C 。

2) 2 个集合 A 的异或和不一定为集合 A , 但一定是集合 B ; 集合 A 与集合 C 的异或和仍是集合 A ; 2 个集合 B 的异或和仍为集合 B 。

3) 集合 B 通过非线性双射 (如模 2^8 加), 将无法确定其平衡性。

本文讨论的 HIGHT 算法以字节为操作单位, 即将定义 1 中的 n 取 8。

由于 HIGHT 算法涉及到模 2^8 加运算, 有必要对不同形式集合间的模 2^8 加运算原则进行归纳, 由性质 1 的 3) 进一步得到性质 2。

性质 2 不同集合类型的字节之间进行模 2^8 加运算遵循以下性质。

1) 集合 A 与集合 C 的模 2^8 加和仍是集合 A ; 集合 B 与集合 C 的模 2^8 加和仍是集合 B 。

2) 集合 B 与集合 A 的模 2^8 加和无法确定其平衡性, 但其最低比特仍保持平衡, 记为 $uuuuuuub$; 2 个 B 集合的模 2^8 加和无法确定其平衡性, 但其最低比特仍保持平衡, 记为 $uuuuuuub$ 。

证明 2 个集合间的模 2^8 加运算结果: 最低比特为 2 个集合最低比特之间进行异或加运算所得, 其余比特为 2 个集合对应比特之间进行异或加运算再加上低比特的进位所得。由于集合 B 与集合 A 最低比特均为平衡比特, 所以根据性质 1 的 2) 可得, 集合 B 与集合 A 进行模 2^8 加运算后, 最低比特仍

保持平衡，其余比特由于涉及到低比特的进位，平衡性不能确定。同理可得 2 个集合 B 的模 2^8 加运算的结果。

证毕

3 HIGHT 算法 17 轮积分区分器的构造

文献[2]给出了 HIGHT 算法的 11 轮积分区分器，在构造过程中，认为集合 A 经过 F 函数仍为集合 A ，根据性质 1 的 2) 可知，集合 A 经过 F 函数后应为集合 B ，据此，本文重新构造 HIGHT 算法的 11 轮积分区分器(定理 1)，所得结果与文献[2]仍相同。如图 2 所示。

定理 1 (11 轮积分区分器) 选择 2^8 个明文，满足条件： P_7 遍历所有 2^8 个取值，即为集合 $A, P_6, P_5, P_4, P_3, P_2, P_1, P_0$ 均为固定值，即为集合 C 。则经过 11 轮 HIGHT 算法后，则输出的 $X_{11,3}$ 最低比特仍保持平衡。

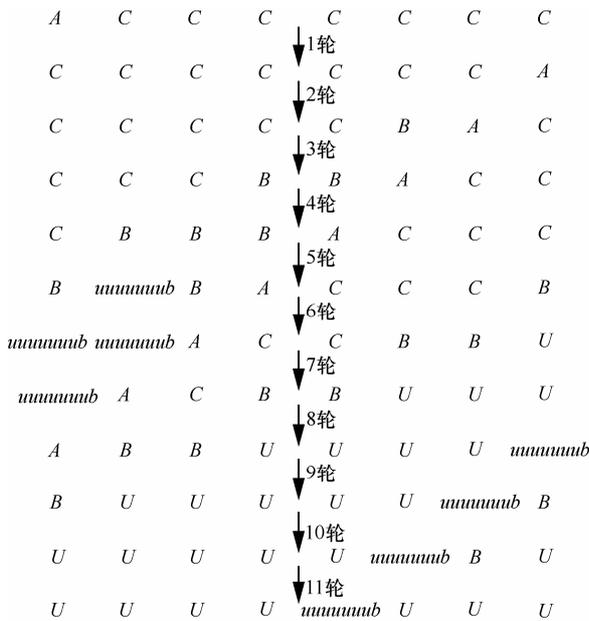


图 2 HIGHT 算法的 11 轮积分区分器

同理，可以得到另一个 11 轮积分区分器

$$(C, C, C, C, A, C, C, C) \xrightarrow{11 \text{ 轮}} (uuuuuuub, U, U, U, U, U, U, U)$$

进一步，文献[2]将 11 轮区分器向前做高阶积分扩展，得到 17 轮积分区分器(定理 2)。如图 3 所示。

定理 2 (17 轮积分区分器) 选择 2^{56} 个明文，满足条件： $P_7, P_6, P_5, P_4, P_3, P_2, P_1$ 分别遍历所有 2^8 个取值， P_0 为固定值，即为集合 C 。则经过 17 轮 HIGHT

后，输出的 $X_{17,3}$ 最低比特仍保持平衡。

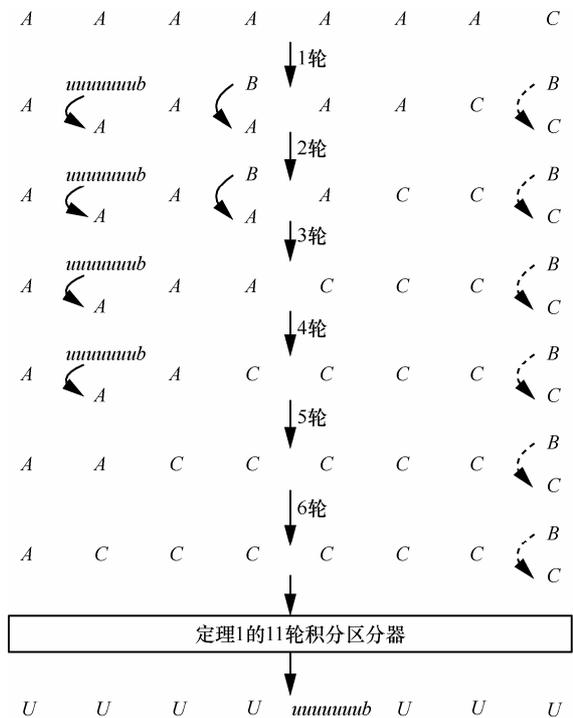


图 3 17 轮积分区分器

同理，可以得到另一个 17 轮积分区分器 I:

$$(A, A, A, C, A, A, A, A) \xrightarrow{17 \text{ 轮}} (uuuuuuub, U, U, U, U, U, U, U)$$

4 对 25 轮 HIGHT 算法的积分攻击

文献[2]在 17 轮积分区分器的基础上攻击了 22 轮 HIGHT 算法，本文利用“时空折中”的原理降低攻击的计算复杂度，实现对 25 轮 HIGHT 算法的积分攻击。选取构造 17 轮区分器时需要的明文，得到 25 轮加密的密文，通过猜测相关密钥，从 25 轮加密的结果恢复出第 17 轮的结果，验证其中 $X_{17,3}$ 的最低比特 $X_{17,3}^{(0)}$ 是否平衡。

4.1 积分攻击流程

记字节 X 的最低比特为 $X^{(0)}$ 。根据算法加密流程，攻击算法如下。

算法 1 25 轮 HIGHT 算法积分攻击

步骤 1 选择构造 17 轮积分区分器时的明文(2^{56} 个)，进行 25 轮加密得到 2^{56} 个密文。

步骤 2 猜测结尾密钥 wk_7, wk_6, wk_5, wk_4 ，对 2^{56} 个密文解密，计算第 25 轮的输出： $X_{25,7} = C_7$ ， $X_{25,6} = C_6 \oplus wk_7$ ， $X_{25,5} = C_5$ ， $X_{25,4} = C_4 \oplus wk_6$ ， $X_{25,3} = C_3$ ， $X_{25,2} = C_2 \oplus wk_5$ ， $X_{25,1}^{(0)} = C_1^{(0)}$ ， $X_{25,0} = C_0 \oplus wk_4$ 。

步骤 3 猜测第 25 轮子密钥 $sk_{99}, sk_{96}, sk_{97}, sk_{98}$, 用步骤 2 的结果解密第 25 轮, 得到 2^{56} 个第 24 轮部分输出: $X_{24,7} = X_{25,6} \oplus (F_0(X_{25,5}) \boxplus sk_{99})$, $X_{24,6} = X_{25,5}$, $X_{24,5} = X_{25,4} \boxplus (F_1(X_{25,3}) \oplus sk_{96})$, $X_{24,4} = X_{25,3}$, $X_{24,3} = X_{25,2} \oplus (F_0(X_{25,1}) \boxplus sk_{97})$, $X_{24,2} = X_{25,1}^{(0)}$, $X_{24,1} = X_{25,0} \boxplus (F_1(X_{25,7}) \oplus sk_{98})$, $X_{24,0} = X_{25,7}$ 。

步骤 4 猜测第 24 轮子密钥 $sk_{95}, sk_{92}, sk_{93}$ 和 $sk_{94}^{(0)}$, 用步骤 3 的结果解密第 24 轮, 得到 2^{56} 个第 23 轮部分输出: $X_{23,7} = X_{24,0} \oplus (F_0(X_{24,7}) \boxplus sk_{95})$, $X_{23,6} = X_{24,7}$, $X_{23,5} = X_{24,6} \boxplus (F_1(X_{24,5}) \oplus sk_{92})$, $X_{23,4} = X_{24,5}$, $X_{23,3} = X_{24,4} \oplus (F_0(X_{24,3}) \boxplus sk_{93})$, $X_{23,1}^{(0)} = X_{24,2}^{(0)} \oplus F_1(X_{24,1})^{(0)} \oplus sk_{94}^{(0)}$, $X_{23,0} = X_{24,1}$, 其中, $F_1(X_{24,1})^{(0)}$ 表示 $F_1(X_{24,1})$ 的最低比特。

步骤 5 猜测第 23 轮子密钥 $sk_{91}, sk_{88}, sk_{89}$, 用步骤 4 的结果解密第 23 轮, 得到 2^{56} 个第 22 轮部分输出: $X_{22,7} = X_{23,0} \oplus (F_0(X_{23,7}) \boxplus sk_{91})$, $X_{22,6} = X_{23,7}$, $X_{22,5} = X_{23,6} \boxplus (F_1(X_{23,5}) \oplus sk_{88})$, $X_{22,4} = X_{23,5}$, $X_{22,3} = X_{23,4} \oplus (F_0(X_{23,3}) \boxplus sk_{89})$, $X_{22,0}^{(0)} = X_{23,1}^{(0)}$ 。

步骤 6 猜测第 22 轮子密钥 sk_{84}, sk_{85} , $sk_{87}^{(0)}$, 用步骤 5 的结果解密第 22 轮, 得到 2^{56} 个第 21 轮部分输出: $X_{21,7}^{(0)} = X_{22,0}^{(0)} \oplus (F_0(X_{22,7})^{(0)} \oplus sk_{87}^{(0)})$, $X_{21,6} = X_{22,7}$, $X_{21,5} = X_{22,6} \boxplus (F_1(X_{22,5}) \oplus sk_{84})$, $X_{21,4} = X_{22,5}$, $X_{21,3} = X_{22,4} \oplus (F_0(X_{22,3}) \boxplus sk_{85})$, 其中, $F_0(X_{22,7})^{(0)}$ 表示 $F_0(X_{22,7})$ 的最低比特。

步骤 7 猜测第 21 轮子密钥 sk_{80}, sk_{81} , 用步骤 6 的结果解密第 21 轮, 得到 2^{56} 个第 20 轮部分输出: $X_{20,6}^{(0)} = X_{21,7}^{(0)}$, $X_{20,5} = X_{21,6} \boxplus (F_1(X_{21,5}) \oplus sk_{80})$, $X_{20,4} = X_{21,5}$, $X_{20,3} = X_{21,4} \oplus (F_0(X_{21,3}) \boxplus sk_{81})$ 。

步骤 8 猜测第 20 轮子密钥 sk_{77} , $sk_{76}^{(0)}$, 用步骤 7 的结果解密第 20 轮, 得到 2^{56} 个第 19 轮部分输出: $X_{19,5}^{(0)} = X_{20,6}^{(0)} \oplus (F_1(X_{20,5})^{(0)} \oplus sk_{76}^{(0)})$, $X_{19,4} = X_{20,5}$, $X_{19,3} = X_{20,4} \oplus (F_0(X_{20,3}) \boxplus sk_{77})$, 其中, $F_1(X_{20,5})^{(0)}$ 表示 $F_1(X_{20,5})$ 的最低比特。

步骤 9 猜测第 19 轮子密钥 sk_{73} , 用步骤 8 的结果解密第 19 轮, 得到 2^{56} 个第 18 轮部分输出: $X_{18,4}^{(0)} = X_{19,5}^{(0)}$, $X_{18,3} = X_{19,4} \oplus (F_0(X_{19,3}) \boxplus sk_{73})$ 。

步骤 10 猜测第 18 轮子密钥 $sk_{69}^{(0)}$, 用步骤 9 的结果解密第 18 轮, 得到 2^{56} 个第 17 轮部分输出:

$X_{17,3}^{(0)} = X_{18,4}^{(0)} \oplus (F_0(X_{18,3})^{(0)} \oplus sk_{69}^{(0)})$, $F_0(X_{18,3})^{(0)}$ 表示 $F_0(X_{18,3})$ 的最低比特, 验证 $X_{17,3}^{(0)}$ 是否为平衡比特, 若是, 则所猜测的密钥为候选密钥, 否则为错误密钥, 删除。

步骤 11 选择另一组构造 17 轮区分器时的明文, 重复步骤 1~步骤 10, 直至密钥唯一确定。

攻击流程如图 4 所示。

4.2 积分攻击算法的分析

对算法 1 进行分析, 利用存储空间分担算法的计算时间: 单独计算算法中每一步骤的时间复杂度, 存储每一步的计算结果, 在下一步计算中调用上一步存储的结果, 最后将每一步的时间复杂度相加得到整个算法的时间复杂度。得到定理 3。

定理 3 利用算法 1 对 25 轮 HIGHT 算法进行积分攻击, 攻击的数据复杂度为 $2^{62.92}$, 时间复杂度为 $2^{66.20}$, 空间复杂度为 2^{119} 。

证明 算法 1 需要猜测 8 bit 密钥 $wk_7, wk_6, wk_5, wk_4, sk_{73}, sk_{77}, sk_{80}, sk_{81}, sk_{84}, sk_{85}, sk_{88}, sk_{89}, sk_{91}, sk_{92}, sk_{93}, sk_{95}, sk_{96}, sk_{97}, sk_{98}, sk_{99}$ 和 1 bit 密钥 $sk_{69}^{(0)}, sk_{76}^{(0)}, sk_{87}^{(0)}, sk_{94}^{(0)}$ 。根据密钥扩展算法, 这些密钥由某些主密钥生成, 如表 1 所示。

根据算法 1 及表 1 可知, 攻击过程中需要先后猜测 wk_7, wk_6, wk_5, wk_4 ; sk_{99}, sk_{98} ; $sk_{95}, sk_{92}, sk_{93}, sk_{94}^{(0)}$; $sk_{91}, sk_{88}, sk_{89}$; sk_{84} ; sk_{77} 的前 7 bit, sk_{73} , 共 120 bit 密钥。对于正确密钥, 一定能保证 $X_{17,3}^{(0)}$ 为平衡比特; 对于错误密钥, 其使 $X_{17,3}^{(0)}$ 平衡的概率为 $\frac{1}{2}$, 所以经过一组明密文淘汰后, 剩余错误密钥数

目为 $(2^{120} - 1) \times \frac{1}{2} \approx 2^{119}$, 为了唯一确定正确密钥, 需要 121 组明文, 从而攻击的数据复杂度为 121 组 ($2^{56} \times 121 \approx 2^{62.92}$ 个明文)。

攻击的时间复杂度按如下方法估计: 对第一组明文, 步骤 1 需要 2^{56} 次 25 轮加密, 步骤 2 需要猜测 wk_7, wk_6, wk_5, wk_4 共 32 bit 密钥, 共 2^{32} 个可能值, 在密钥固定的情况下, 结尾密钥模 2^8 加运算只依赖于 C_4 和 C_0 , 取值最多 2^{16} 种 (相同的不重复计算), 最多需 $2^{32} \times 2^{16} \times 2 = 2^{49}$ 次模 2^8 加运算, 步骤 3 需要猜测 sk_{99}, sk_{98} 共 16 bit 密钥, 2^{16} 个可能值, 在密钥固定的情况下, 模 2^8 加运算依赖于 $F_0(X_{25,5}), X_{24,4}, F_1(X_{25,3}), F_0(X_{25,1}), X_{25,0}, F_1(X_{25,7})$ 共 48 bit, 最多 2^{48} 个可能值, 最多需 $2^{16} \times 2^{48} \times 4 = 2^{66}$ 次模 2^8 加运算,

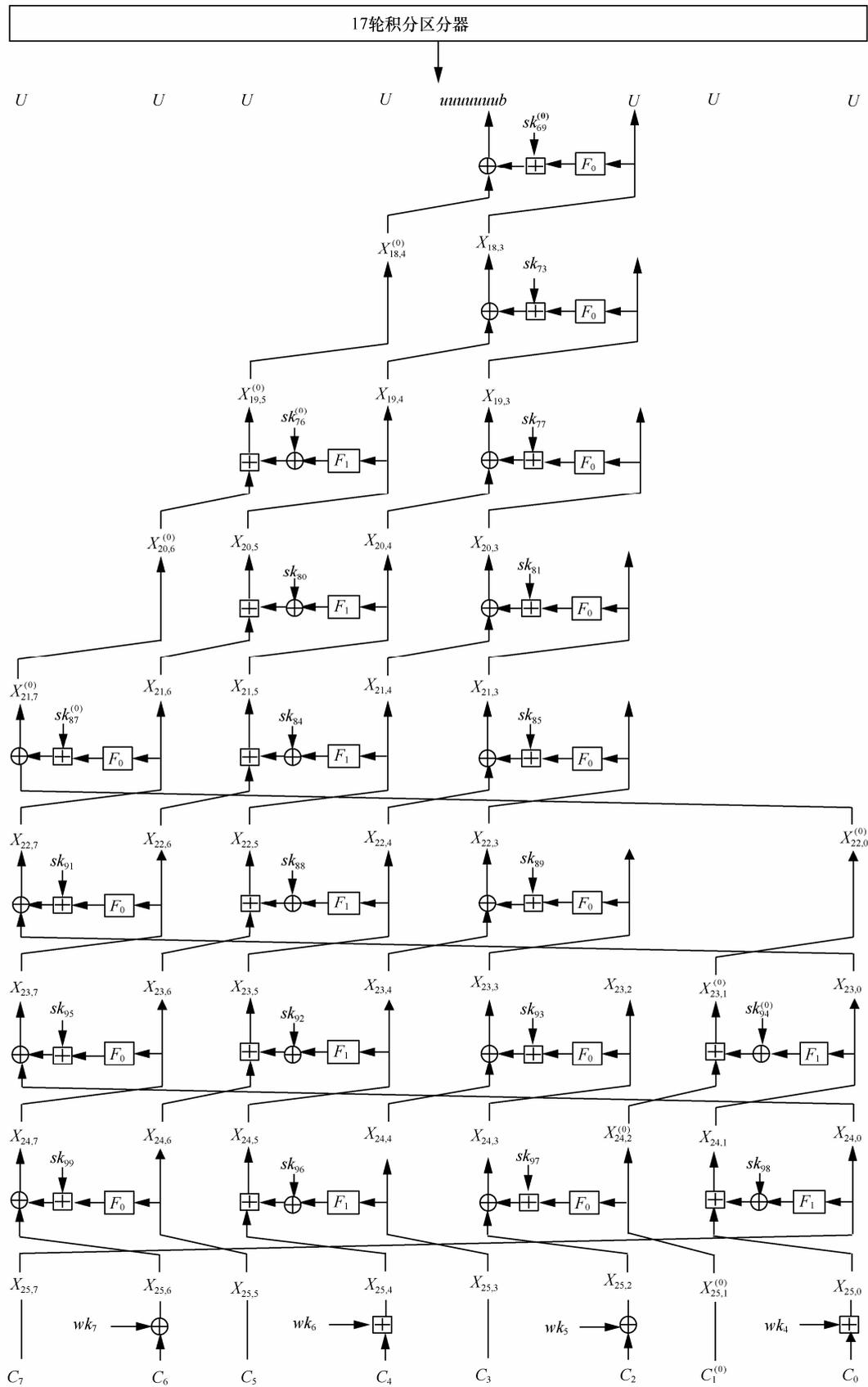


图 4 25 轮 HIGHT 算法的积分攻击

表 1 算法 1 所猜密钥与主密钥的关系

wk_7	wk_6	wk_5	wk_4	sk_{73}	sk_{77}	sk_{80}	sk_{81}	sk_{84}	sk_{85}	sk_{88}	sk_{89}
MK_3	MK_2	MK_1	MK_0	MK_{13}	MK_9	MK_3	MK_4	MK_7	MK_0	MK_{11}	MK_{12}
sk_{91}	sk_{92}	sk_{93}	sk_{95}	sk_{96}	sk_{97}	sk_{98}	sk_{99}	$sk_{69}^{(0)}$	$sk_{76}^{(0)}$	$sk_{87}^{(0)}$	$sk_{94}^{(0)}$
MK_{14}	MK_{15}	MK_8	MK_{10}	MK_2	MK_3	MK_4	MK_5	$MK_1^{(0)}$	$MK_8^{(0)}$	$MK_2^{(0)}$	$MK_9^{(0)}$

步骤 4 需要猜测 $sk_{95}, sk_{92}, sk_{93}, sk_{94}^{(0)}$ 共 25 bit 密钥, 共 2^{25} 个可能值, 在密钥固定的情况下, 模 2^8 加运算依赖于 $F_0(X_{24,7}), X_{24,6}, F_1(X_{24,5}), F_0(X_{24,3})$ 共 32 bit, 最多 2^{32} 个可能值, 最多需要 $2^{25} \times 2^{32} \times 3 = 2^{58.58}$ 次模 2^8 加运算, 同理, 步骤 5 需要猜测 $sk_{91}, sk_{88}, sk_{89}$ 共 24 bit 密钥, 最多需要 $2^{24} \times 2^{32} \times 3 = 2^{57.58}$ 次模 2^8 加运算, 步骤 6 需要猜测 sk_{84} 这 8 bit 密钥, 最多需 $2^8 \times 2^{24} \times 2 = 2^{33}$ 次模 2^8 加运算, 步骤 7 需要猜测 0 bit 密钥, 最多需 $2^{24} \times 2 = 2^{25}$ 次模 2^8 加运算, 步骤 8 需要猜测 sk_{77} 的前 7 bit 密钥, 最多需 $2^7 \times 2^8 = 2^{15}$ 次模 2^8 加运算, 步骤 9 需要猜测 sk_{73} 这 8 bit 密钥, 最多需 $2^8 \times 2^8 = 2^{16}$ 次模 2^8 加运算, 步骤 10 计算量可忽略不计, 由于 25 轮算法一共有 $4 \times 25 + 4 = 104$ 次模 2^8 加运算, 因此, 在忽略其他运算所耗时间的情况下, 模 2^8 加运算转换为 25 轮加密操作, 相当于 $2^{56} + \frac{2^{49} + 2^{66} + 2^{58.58} + 2^{57.58} + 2^{33} + 2^{25} + 2^{15} + 2^{16}}{104} \approx 2^{59.45}$

次 25 轮加密; 处理完第一组明文后, 错误密钥还剩 2^{119} 个 (大于 2^{32}), 同理处理第二组明文最多需 $2^{59.45}$ 次 25 轮加密; 只要候选密钥剩余个数不小于 2^{32} , 处理每组明文均最多需 $2^{59.45}$ 次 25 轮加密, 因此当处理完 89 组明文后, 剩余候选密钥 2^{31} 个, 处理第 90 组最多需 $2^{56} + \frac{2^{48} + 2^{66} + 2^{58.58} + 2^{57.58} + 2^{33} + 2^{25} + 2^{15} + 2^{16}}{104} \approx 2^{59.45}$

次 25 轮加密; 处理完 90 组明文后, 剩余候选密钥 2^{30} 个, 处理第 91 组最多仍需 $2^{59.45}$ 次 25 轮加密; 类似地, 处理后 30 组明文分别最多需 $2^{59.45}, 2^{59.45}, 2^{59.45}, 2^{59.45}, 2^{59.44}, 2^{59.44}, 2^{59.44}, 2^{59.44}, 2^{59.44}, 2^{59.44}, 2^{58.57}, 2^{57.79}, 2^{57.16}, 2^{56.69}, 2^{56.39}, 2^{56.21}, 2^{56.17}, 2^{56.05}, 2^{56.03}, 2^{56.01}, 2^{56.01}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}, 2^{56}$ 次 25 轮加密。所以攻击过程时间复杂度不超过 $2^{59.45} \times 97 + 2^{59.44} \times 8 + 2^{58.57} + 2^{57.79} + 2^{57.16} + 2^{56.69} + 2^{56.39} + 2^{56.21} + 2^{56.17} + 2^{56.05} + 2^{56.03} + 2^{56.01} \times 2 + 2^{56} \times 5 \approx 2^{66.20}$ 。此外, 攻击过程中还需要 2^{119} 的存储空间存储候选密钥, 模 2^8 加运算的表以及攻击过程中每一步骤的某些中间数据。

证毕

同理, 可利用区分器 I 对 25 轮 HIGHT 算法进行攻击。

5 结束语

本文对 HIGHT 算法在积分攻击下的安全性进行了研究, 纠正了文献[2]中构造 11 轮区分器时的错误, 通过向前做高阶积分将 11 轮区分器扩展至 17 轮。依据 17 轮区分器, 利用“时空折中”的原理降低计算复杂度, 攻击了 25 轮 HIGHT 算法。攻击算法的数据复杂度、时间复杂度和空间复杂度分别为 $2^{62.92}, 2^{66.20}$ 和 2^{119} 。攻击轮数和时间复杂度都要优于文献[2]的结果。

本文结果表明, 针对广义 Feistel 结构的算法, 在拥有足够的存储空间的情况下, 可以利用“时空折中”的原理攻击更多轮算法结构。在未来的工作中, 将利用这一原理, 对其他结构分组密码算法 (SPN 结构、Feistel 结构等) 进行分析, 以达到在降低时间复杂度的同时, 攻击更多轮算法的研究目的, 进一步推广积分攻击算法的应用范围。

表 2 将本文积分攻击的结果与文献[2]的结果做了比较。

表 2 HIGHT 算法积分攻击的结果比较

积分攻击来源	攻击轮数	数据复杂度	时间复杂度	空间复杂度
文献[2]	22	$2^{62.04}$	$2^{118.71}$	2^{64} (存储候选密钥)
本文	25	$2^{62.92}$	$2^{66.20}$	2^{119}

参考文献:

[1] HONG D, SUNG J, HONG S, et al. HIGHT: a new block cipher suitable for low-resource device[C]//Cryptographic Hardware and Embedded Systems - CHES 2006. c2006: 46-59.
 [2] ZHANG P, SUN B, LI C. Saturation attack on the block cipher HIGHT[C]//The 8th International Conference on Cryptology and Network Security. c2009:76-86.
 [3] KOO B, HONG D, KWON D. Related-key attack on the full HIGHT[C]//Information Security and Cryptology - ICISC 2010. c2010:

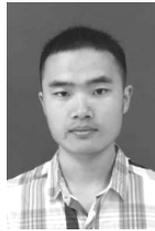
49-67.

- [4] HONG D, KOO B, KWON D. Biclique attack on the full HIGHT[C]// Information Security and Cryptology - ICISC 2011. c2011: 365-374.
- [5] CHEN J, WANG M, PRENEEL B. Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT[C]// AF-RICACRYPT 2012. c2012: 117-137.
- [6] IGARASHI Y, SUEYOSHI R, KANEKO T, et al. Meet-in-the-middle attack with splice-and-cut technique on the 19-round variant of block cipher HIGHT[J]. Information Science and Applications, 2015, 339: 423-429.
- [7] 范伟杰, 吴文玲, 张蕾. HIGHT 算法的差分故障攻击[J]. 中国科学院研究生院学报, 2012, 29(2): 271-276.
FAN W J, WU W L, ZHANG L. Differential fault analysis on HIGHT[J]. Journal of Graduate University of Chinese Academy of Science. 2012, 29(2): 271-276.
- [8] 陈浩, 王韬, 张帆, 等. HIGHT 密码代数故障分析[J]. 上海交通大学学报. 2015, 49(12): 1817-1825.
CHEN H, WANG T, ZHANG F, et al. Algebraic fault analysis of HIGHT[J]. Journal of Shanghai Jiaotong University, 2015, 49(12): 1817-1825.
- [9] KNUDSEN L, WAGNER D. Integral cryptanalysis[C]//FSE 2002. Leuven, Belgium, c2002: 112-127.
- [10] MINER M, PHAN R W, POUSSE B. On integral distinguishers of rijndael family of ciphers[J]. Cryptologia, 2012, 36(2): 104-118.
- [11] YU S, LEI W. Meet-in-the-middle technique for integral attacks against feistel ciphers[C]//Selected Areas in Cryptography 2012. c2012: 234-251.
- [12] YI W, CHEN S. Integral cryptanalysis of the block cipher E2[EB/OL]. <http://arxiv.org/pdf/1404.6100.pdf>.
- [13] YI W, CHEN S. Improved results on integral and zero-correlation linear cryptanalysis of the block cipher MIBS[EB/OL]. <http://arxiv.org/pdf/1404.6100.pdf>.
- [14] 李超, 孙兵, 李瑞林. 分组密码的攻击方法与实例分析[M]. 北京: 北京科学出版社, 2010: 175-207.
LI C, SUN B, LI R L. Block cipher attack method and example analysis[M]. Beijing: Beijing Science Press, 2010:175-207.

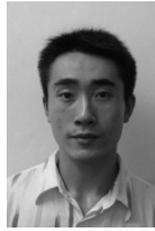
作者简介:



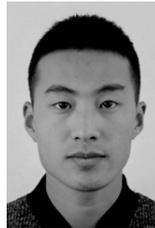
郭建胜 (1972-), 男, 河南沁阳人, 解放军信息工程大学教授, 主要研究方向为信息安全与密码学。



崔竞一 (1992-), 男, 河南登封人, 解放军信息工程大学硕士生, 主要研究方向为分组密码设计与分析。



潘志舒 (1985-), 男, 江苏镇江人, 西安卫星测控中心助理工程师, 主要研究方向为分组密码设计与分析。



刘翼鹏 (1992-), 男, 山东烟台人, 解放军信息工程大学硕士生, 主要研究方向为信息安全与量子密码。