Software Engineering and Applications 软件工程与应用, 2012, 1, 17-23
http://dx.doi.org/10.12677/sea.2012.12004 Published Online December 2012 (http://www.hanspub.org/journal/sea.html)

A Meet-in-the-Middle Attack on Reduced-Round Crypton*

Chao Liu[1], Fucheng Liao[1], Hongru Wei[2]

[1]School of Mathematics and Physics, University of Science and Technology Beijing, Beijing
[2]State Key Laboratory of Information Security, Beijing
Email: fcliao@ustb.edu.cn

Received: Nov. 8th, 2012; revised: Nov. 16th, 2012; accepted: Nov. 30th, 2012

Abstract: Crypton, one of AES candidates, is a 128 bit block cipher of SPN structure proposed by Lim. By means of the concept of Multiset, this paper evaluates the security of the reduced-round Crypton against meet-in-the-middle attack, constructs two categories of distinguishers of 4/5 round used to the attack on Crypton algorithm of 7/8/9 round. All the attack instances give the complexity analysis. The results demonstrate that Crypton reduced to 9 rounds is not immune to meet-in-the-middle attacks, and new attacks reduced the data complexity efficiently.

Keywords: Crypton Algorithm; Meet-in-the-Middle Attack; Multiset; Distinguisher

对简化轮数的 Crypton 算法的中间相遇攻击*

刘　超[1]，廖福成[1]，卫宏儒[2]

[1]北京科技大学数理学院，北京
[2]信息安全国家重点实验室，北京
Email: fcliao@ustb.edu.cn

收稿日期：2012 年 11 月 8 日；修回日期：2012 年 11 月 16 日；录用日期：2012 年 11 月 30 日

摘　要：Crypton 算法是一种 SPN 型分组密码，它是分组长度为 128 bit 的 AES 候选算法之一。本文借助于多重集的概念，评估了简化轮数的 Crypton 算法对中间相遇攻击的抵抗能力，设计出两类 4/5 轮区分器，对 7/8/9 轮的 Crypton 算法实施了攻击。所有的攻击实例都给出了复杂度分析，攻击结果表明 9 轮的 Crypton 算法对中间相遇攻击是不免疫的，而且新攻击有效地降低了攻击所需的数据复杂度。

关键词：Crypton 算法；中间相遇攻击；多重集；区分器

1. 引言

文献[1]设计了一种 AES 候选算法，该算法被学术界用文献[1]作者的名字命名为 Crypton 算法。该算法中密码采用分组长度为 128 bit，密钥长度可变且最多为 256 bit 的 SPN 型结构。文献[1]指出，Crypton 对差分密码分析和线性密码分析是安全的，但是其密钥扩展算法相对比较简单。文献[2]对其进行了改进，得到 Crypton v1.0。由于 Crypton 具有加密过程严格一致、结构高度灵活、并且可并行运算等特点，使其倍受密码分析者的关注。目前人们所做的工作主要有：H'Halluin 等在 FSE1999 上提出了对 6 轮 Crypton 算法修正的 Square 攻击[3]，2009 年王兵等分别对 5，6 轮的 Crypton 算法实施了新的积分攻击[4]，2010 年 Mala 等找到了 Crypton 算法的 4 轮不可能差分，给出了 7 轮算法的两个攻击[5]，2011 年魏悦川等构造了一系列 6 轮相关密钥不可能差分，并提出了对 9 轮 Crypton 算法的两个攻击[6]。

中间相遇攻击由 Dilfie 和 Hellman 针对 Two-DES 分析时提出的一种时空折中的攻击方法[7]，其实质是

*资助信息：内蒙古自治区科技创新引导奖励资金项目(2012)；国家自然科学基金项目(61174209)；信息安全国家重点实验室 2011 年开放课题(02-04-3)。

用存储复杂度来换取时间复杂度。与其他的分析方法相比，在攻击相同轮数的情况下，该方法所需的数据量相对较少。其基本思想是：首先找到一种区分器，然后猜测相应的密钥对特定的明文和密文分别向下加密和向上解密若干轮，最后与区分器的首尾相接，构成完整的攻击线路。如果上述对接能够成功，则说明猜测的密钥是正确的，需要保留，否则是错误的，于是将其淘汰。由此可知当所有的错误密钥都被淘汰后，剩下的密钥即为正确密钥。这种方法尽管有大量的预计算复杂度和存储复杂度，但预计算只需进行一次，而且可以降低攻击时的时间复杂度，所以还是比较有效的。截止到目前，这种方法已被广泛应用于许多分组密码的分析中，如见文献[8-11]。

中间相遇攻击中的“多重集”[12]概念最初是由 Dunkelman 和 Shamir 在分析 AES 时引入的，后来文献[13]又将其引入到了与 AES 有着相似结构的 ARIA 算法的分析中。

本文将借助多重集的概念，研究 Crypton 算法抵抗中间相遇攻击的能力。通过对 Crypton 算法的结构特点的分析，设计出该算法的两类 4/5 轮区分器，并用于分别对 7/8/9 轮的 Crypton 实施攻击。

2. 预备知识：Crypton 算法简介

Crypton 密码采用 SPN 结构，标准加密轮数为 12 轮。它的分组长度为 128 bit，密钥长度可变且最多为 256 bit。对于 128 bit 的分组数据 $A=\left(a_{0,0},a_{0,1},\cdots,a_{3,3}\right)$ 可表示为

$$A=\begin{bmatrix} a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \end{bmatrix} \quad \begin{bmatrix} 12 & 13 & 14 & 15 \\ 8 & 9 & 10 & 11 \\ 4 & 5 & 6 & 7 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

Crypton 密码的轮函数由 4 种变换组成：

1) 字节代替变换(γ)：将数据每一字节操作非线性变换(S 盒)，其中 γ_0 和 γ_e 分别作用于奇数轮和偶数轮。

2) 列混合变换(π)：将状态矩阵中的每一列分别与已知字节作用。其中用于奇数轮的 π_0 定义为 $B_{i,j}=\oplus_{k=0}^{3}\left(A_{k,j}\cdot m_{(i+j+k)\bmod 4}\right)$，用于偶数轮的 π_e 定义为 $B_{i,j}=\oplus_{k=0}^{3}\left(A_{k,j}\cdot m_{(i+j+k+2)\bmod 4}\right)$。这里 $m_0=0\times fc$，$m_1=0\times f3$，$m_2=0\times cf$，$m_3=0\times 3f$。而且容易验证矩阵 $\boldsymbol{\pi}_0$ 和 $\boldsymbol{\pi}_e$ 都是自逆的，即 $\boldsymbol{\pi}_0=\boldsymbol{\pi}_0^{-1}$，$\boldsymbol{\pi}_e=\boldsymbol{\pi}_e^{-1}$。

3) 字节对称变换(τ)：将在(i,j)位置上的字节变换到(j,i)位置，即 $B=\tau(A)\Leftrightarrow b_{i,j}=a_{i,j}$，明显地有 $\tau^{-1}=\tau$。

4) 轮密钥加变换(σ_K)：将 128 比特的轮密钥 K_i 直接与数据按比特位异或。

令 K_i 为第 i 轮的轮密钥，它是通过主密钥 K 经过密钥扩展算法得到的，于是完整的 Crypton 算法可描述为：$E_K=\phi_e\rho_eK_{12}\circ\rho_0K_{11}\circ\cdots\circ\rho_eK_2\circ\rho_1K_1\circ\sigma_{K_0}$。其中奇数轮：$\rho_0K=\sigma_K\circ\tau\circ\pi_0\circ\gamma_0$，偶数轮：$\rho_eK=\sigma_K\circ\tau\circ\pi_e\circ\gamma_e$，而线性变换 $\phi_e=\tau\circ\pi_e\circ\tau$ 为作用在最后一轮之后的末变换，当最后一轮为奇数轮时，相应的变换定义为 $\phi_0=\tau\circ\pi_0\circ\tau$。

另外，在一些情况下，可以将轮函数 $\sigma_K\circ\tau\circ\pi_t\circ\gamma_t$ 写成其等价形式 $\tau\circ\pi_t\circ\sigma_{K^{eq}}\circ\gamma_t$，$t\in\{0,e\}$，相应地密钥 K 也替换为等价密钥 K^{eq}，其中 $K^{eq}=\pi^{-1}\circ\tau^{-1}(K)$。由于密钥扩展算法和 S 盒具体的构造与本文分析无关，可将两个版本的算法 Crypton 和 Crypton v1.0 统称为 Crypton。

3. Crypton 多重集与区分器的设计

本节利用 Crypton 算法的结构特点，首先给出 Crypton 的 4/5 轮多重集，然后设计相应的 4/5 轮中间相遇区分器。

3.1. Crypton 的多重集

3.1.1. Crypton 的 4 轮多重集

首先考虑 δ-集[12]的第 0 字节为活跃字节时进行 4 轮加密的情况(其他位置的字节情况类似)。我们用 $T_u(\gamma),T_u(\pi),T_u(\tau),T_u(\sigma_K)$ 分别表示经过 γ,π,τ,σ_K 运算后的中间输出状态。

今从低轮开始推导。在第 u 轮时，差分 $\left\{\Delta T_u^0(\gamma),\Delta T_u^1(\gamma),\cdots,\Delta T_u^{255}(\gamma)\right\}$ 是已知的，因为这些正是第 0 字节的 256 种可能的差分值(其他 15 个字节都相等，没有差分)。我们只关心多重集而不关心差分顺序。由于 π、τ 和 σ_K 都是线性变换，所以经过上面的操作就得到了差分 $\left\{\Delta T_{u+1}^0(\sigma_K),\Delta T_{u+1}^1(\sigma_K),\cdots,\Delta T_{u+1}^{255}(\sigma_K)\right\}$ 的值。由于 δ-集的活跃字节是第 0 字节，所以上述差分的活跃字节

为$\{0,1,2,3\}$，其他字节均为非活跃的。

把$T_{u+1}^0(\sigma_K)$的 0, 1, 2, 3 字节值作为参数的一部分，可以得到$\{T_{u+1}^1(\sigma_K),T_{u+1}^2(\sigma_K),\cdots,T_{u+1}^{255}(\sigma_K)\}$的 0, 1, 2, 3 各字节的值，当然也得到了$\{T_{u+1}^0(\gamma),T_{u+1}^1(\gamma),\cdots,T_{u+1}^{255}(\gamma)\}$的 0, 1, 2, 3 字节值。又差分$\Delta T_{u+1}^i(\gamma)\ (i=1,2,\cdots,255)$除了 0, 1, 2, 3 之外的字节全为零，并且$\Delta T_{u+1}^0(\gamma)=T_{u+1}^0(\gamma)\oplus T_{u+1}^0(\gamma)=0$，所以可推得$\{\Delta T_{u+1}^0(\gamma),\Delta T_{u+1}^1(\gamma),\cdots,\Delta T_{u+1}^{255}(\gamma)\}$的值。

由于变换π,τ,σ_K的线性，所以知道了差分$\{\Delta T_{u+1}^0(\gamma),\Delta T_{u+1}^1(\gamma),\cdots,\Delta T_{u+1}^{255}(\gamma)\}$也就知道了差分$\{\Delta T_{u+2}^0(\sigma_K),\Delta T_{u+2}^1(\sigma_K),\cdots,\Delta T_{u+2}^{255}(\sigma_K)\}$。如果把$T_{u+2}^0(\sigma_K)$的 16 字节值作为变量的一部分给出，就可以得到$T_{u+2}^i(\sigma_K)$的值$(i=1,2,\cdots,255)$。

然后由状态$\{T_{u+2}^0(\sigma_K),T_{u+2}^1(\sigma_K),\cdots,T_{u+2}^{255}(\sigma_K)\}$(经过变换$\gamma,\pi,\tau$作用后)和变量$\sigma_{K_{u+3}}$的 0, 4, 8, 12 的 4 个字节值，可以得到$\{T_{u+3}^0(\sigma_K),T_{u+3}^1(\sigma_K),\cdots,T_{u+3}^{255}(\sigma_K)\}$的 0, 4, 8, 12 字节的值，通过变换$\gamma,\pi,\tau$，就得到$\{T_{u+4}^0,\cdots,T_{u+4}^{255}\}$的第 0 个字节值，也就得到了差分$\{\Delta T_{u+4,0}^0,\Delta T_{u+4,0}^1,\cdots,\Delta T_{u+4,0}^{255}\}$的值。

综上可得多重集

$$\left[T_{u+4,v}^0\oplus T_{u+4,v}^0,T_{u+4,v}^1\oplus T_{u+4,v}^0,\cdots,T_{u+4,v}^{255}\oplus T_{u+4,v}^0\right],$$

它完全由以下 24 个字节变量决定：

- 状态$T_{u+1}^0(\sigma_K)$的 4 个字节(当δ-集的活跃字节是第 0 个字节时，则这 4 字节即为 0, 1, 2, 3)；
- 状态$T_{u+2}^0(\sigma_K)$的全部 16 个字节；
- 轮密钥$\sigma_{K_{u+3}}$的 4 个字节(这 4 个字节分别为 0, 4, 8, 12)，而且最多有2^{192}种可能的取值。不过，由于T_u^0有 256 种可能选择，每个δ-集对应2^8个多重集，因此可以通过选择T_u^0使得$T_{u+1,0}^0=0$，来使变量字节数降为 23 个(这是可能的，由于状态T_{u+1}的第 0 字节是活跃字节)。这就把可能的多重集数量降到了$\left(2^8\right)^{23}=2^{184}$。

本文中使用多重集$\{\Delta T_{6,0}^0,\Delta T_{6,0}^1,\cdots,\Delta T_{6,0}^{255}\}$，取$u=2$，$v=0$，对应的$\delta$-集为$\{T_2^0,T_2^1,\cdots,T_2^{255}\}$。

3.1.2. Crypton 的 5 轮多重集

考虑对δ-集进行 5 轮加密的情况。对每个$0\le v\le 15$，(无序)的多重集

$$\left[T_{u+5,v}^0\oplus T_{u+5,v}^0,T_{u+5,v}^1\oplus T_{u+5,v}^0,\cdots,T_{u+5,v}^{255}\oplus T_{u+5,v}^0\right],$$

完全由以下 40 个字节变量决定：

- 状态$T_{u+1}^0(\sigma_K)$的 4 个字节(当δ-集的活跃字节是第 0 个字节时，则这 4 字节即为 0, 1, 2, 3)；
- 状态$T_{u+2}^0(\sigma_K)$的全部 16 个字节；
- 状态$T_{u+3}^0(\sigma_K)$的所有 16 个字节；
- 轮密钥$\sigma_{K_{u+4}}$的 4 个字节(这 4 字节分别为 0, 4, 8, 12)。

分析过程与 4 轮多重集类似。此外，这一多重集最多只有2^{312}个值。本文使用多重集$\{\Delta T_{7,0}^0,\Delta T_{7,0}^1,\cdots,\Delta T_{7,0}^{255}\}$，取$u=2$，$v=0$，对应的$\delta$-集为$\{T_2^0,T_2^1,\cdots,T_2^{255}\}$。

3.2. Crypton 基于多重集的区分器

3.2.1. Crypton 基于多重集的 4 轮区分器

区分器 1

若δ-集的活跃字节是第 0 字节，用 4 轮 Crypton 加密δ-集，则(无序)的多重集$\{\Delta T_{6,0}^0,\Delta T_{6,0}^1,\cdots,\Delta T_{6,0}^{255}\}$完全由以下 24 个字节变量决定：

- 状态$T_3^0(\sigma_K)$的 4 个字节(当δ-集的活跃字节是第 0 个字节时，则这 4 个字节即为 0, 1, 2, 3)；
- 状态$T_4^0(\sigma_K)$的全部 16 个字节；
- 轮密钥σ_{K_5}的 4 个字节(这 4 个字节分别为 0, 4, 8, 12)。

由此可知，多重集完全由 184 字节变量决定。这一多重集共有2^{184}个可能值，所以如果密钥的猜测值使得对应的多重集产生了上述的2^{184}个值中的一个值，那么猜测的这个密钥就很有可能是正确密钥。

3.2.2. Crypton 基于多重集的 5 轮区分器

区分器 2

若δ-集的活跃字节是第 0 字节，用 5 轮 Crypton 加密δ-集，则(无序)的多重集$\{\Delta T_{7,0}^0,\Delta T_{7,0}^1,\cdots,\Delta T_{7,0}^{255}\}$完全由以下 40 个字节变量决定：

- 状态$T_3^0(\sigma_K)$的 4 个字节(当δ-集的活跃字节是第 0 个字节时，则这 4 个字节即为 0, 1, 2, 3)；
- 状态$T_4^0(\sigma_K)$的全部 16 个字节；
- 状态$T_5^0(\sigma_K)$的所有 16 个字节；
- 轮密钥σ_{K_6}的 4 个字节(这 4 个字节分别为 0, 4, 8, 12)。

同样地，由上可知多重集完全由 312 字节变量决定。这一多重集共有2^{312}个可能值，所以如果密钥的

猜测值使得对应的多重集产生了上述的 2^{312} 个值中的一个值，那么猜测的这个密钥就很有可能是正确密钥。

3.2.3. Crypton 基于多重集的改进的 5 轮区分器

区分器 3

在区分器 2 的基础上，我们令变量中几个字节取相同的值以便减少变量个数，从而降低预计算复杂度，使攻击 9 轮的 Crypton 成为可能。但这样做也降低了区分器的成功率，而且增加了选择明文数量。

方案一：选择

$$T_{4,0}^0 = T_{4,1}^0 = T_{4,2}^0 = T_{4,3}^0 = T_{4,4}^0 = T_{4,5}^0 = T_{4,6}^0 = T_{4,7}^0,$$

$$T_{5,0}^0 = T_{5,1}^0 = T_{5,2}^0 = T_{5,3}^0 = T_{5,4}^0,$$

此时的概率为

$$\Pr\left(T_{4,0}^0 = T_{4,1}^0 = T_{4,2}^0 = T_{4,3}^0 = T_{4,4}^0 = T_{4,5}^0 = T_{4,6}^0 = T_{4,7}^0,\right.$$
$$\left. T_{5,0}^0 = T_{5,1}^0 = T_{5,2}^0 = T_{5,3}^0 = T_{5,4}^0\right) = 2^{-8\times 7-8\times 4} = 2^{-88}.$$

方案二：选择

$$T_{4,0}^0 = T_{4,1}^0 = T_{4,2}^0 = T_{4,3}^0 = T_{4,4}^0 = T_{4,5}^0 = T_{4,6}^0,$$
$$T_{5,0}^0 = T_{5,1}^0 = T_{5,2}^0 = T_{5,3}^0,$$

此时的概率为

$$\Pr\left(T_{4,0}^0 = T_{4,1}^0 = T_{4,2}^0 = T_{4,3}^0 = T_{4,4}^0 = T_{4,5}^0 = T_{4,6}^0,\right.$$
$$\left. T_{5,0}^0 = T_{5,1}^0 = T_{5,2}^0 = T_{5,3}^0\right) = 2^{-8\times 6-8\times 3} = 2^{-72}.$$

这时，多重集就完全由 29 或 31 字节变量决定了。此外，多重集最多有 2^{224} 或 2^{240} 个可能值。因此，如果密钥的猜测值使对应的多重集取到上述 2^{224} 或 2^{240} 个值中的一个，那么这一猜测的密钥则就很有可能为正确密钥。

4. 对 7/8/9 轮 Crypton 的中间相遇攻击

4.1. 对 7 轮 Crypton 的中间相遇攻击

首先定义一个明文结构，它是由 2^{32} 个在 4 字节(0, 4, 8, 12)处取固定值的组成的集合。我们利用区分器 1 对 7 轮的 Crypton 实施中间相遇攻击：在 4 轮区分器的前边添 1 轮，后边添 2 轮，构成 7 轮攻击线路，如图 1 所示。在这一攻击中，首先预计算上述区分器中用到的多重集的所有可能值，把它们存储到一个哈希表 T 中。然后选择并加密足量的合乎条件的明文集。搜索特定的密钥字节，对密文进行部分解密，来获得多重集，并检测由解密所得多重集是否在预计算时生成的哈希表中。如果是，则很有可能是正确密钥。

4.1.1. 7 轮攻击过程

步骤 1 预计算阶段

计算由区分器 1 定义的 4 轮多重集的 2^{184} 个可能值，并存储在一个哈希表中。

步骤 2 在线阶段

由第 2 部分的分析可知，分别交换第 6 轮的轮密钥加 K_6^{eq} 与列混合变换、字节对称变换的复合变换 $\tau\circ\pi$，及第 7 轮的轮密钥加 K_7 与列混合变换、字节对称变换的复合变换 $\tau\circ\pi$ 的顺序，即可以得到 K_6^{eq} 是 K_6^{eq} 的等价密钥，K_7^{eq} 是 K_7 的等价密钥。

步骤 2.1. 猜测 K_1 的 1 字节(0)值以及 K_0 的 4 个字节(0, 4, 8, 12)值。对 $T_{2,0}$ 的 256 个可能值，我们令 $T_{2.0}^0 = 0, T_{2.0}^1 = 1, \cdots, T_{2.0}^{255} = 255$。他们组成了 δ - 集 $\left[T_{2,0}^0, T_{2,0}^1, \cdots, T_{2,0}^{255}\right]$。我们通过把 $T_{2,0}^i$ 代入下面状态，得到对应的 $P^0, P^1, \cdots, P^{255}$ 的值。

$$P = \begin{bmatrix} p_{12} & p_{13} & p_{14} & p_{15} \\ p_8 & p_9 & p_{10} & p_{11} \\ p_4 & p_5 & p_6 & p_7 \\ p_0 & p_1 & p_2 & p_3 \end{bmatrix}$$

其中

$$p_0 = \left[\gamma_0^{-1}\circ\pi_0^{-1}\circ\tau^{-1}\left(T_{2,0}^i \oplus K_{1,0}\right)\right]_0 \oplus K_{0,0},$$
$$p_4 = \left[\gamma_0^{-1}\circ\pi_0^{-1}\circ\tau^{-1}\left(T_{2,0}^i \oplus K_{1,0}\right)\right]_4 \oplus K_{0,4},$$
$$p_8 = \left[\gamma_0^{-1}\circ\pi_0^{-1}\circ\tau^{-1}\left(T_{2,0}^i \oplus K_{1,0}\right)\right]_8 \oplus K_{0,8},$$
$$p_{12} = \left[\gamma_0^{-1}\circ\pi_0^{-1}\circ\tau^{-1}\left(T_{2,0}^i \oplus K_{1,0}\right)\right]_{12} \oplus K_{0,12},$$

其他所有 p_i(除 0, 4, 8, 12 字节位置外)均为常量字节，并且不同的位置不一定相等。

步骤 2.2. 猜测 K_6^{eq} 第 0 字节值及 K_7^{eq} 4 字节(0, 1, 2, 3)值。部分解密 $P^0, P^1, \cdots, P^{255}$ 对应的密文，得到多重集 $\left[\Delta T_{6,0}^0, \Delta T_{6,0}^1, \cdots, \Delta T_{6,0}^{255}\right]$。

$$\Delta T_6^i = \gamma_e^{-1}\left(\pi_e^{-1}\circ\tau^{-1}\left(T_7^0\right)\oplus K_6^{eq}\right)\oplus\gamma_e^{-1}\left(\pi_e^{-1}\circ\tau^{-1}\left(T_7^i\right)\oplus K_6^{eq}\right)$$
$$= \gamma_e^{-1}\left(\pi_e^{-1}\circ\tau^{-1}\left(\gamma_0^{-1}\left(\pi_0^{-1}\circ\tau^{-1}\circ\phi_0^{-1}\left(C^0\right)\oplus K_7^{eq}\right)\right)\oplus K_6^{eq}\right)$$
$$\oplus\gamma_e^{-1}\left(\pi_e^{-1}\circ\tau^{-1}\left(\gamma_0^{-1}\left(\pi_0^{-1}\circ\tau^{-1}\circ\phi_0^{-1}\left(C^i\right)\oplus K_7^{eq}\right)\right)\oplus K_6^{eq}\right)$$

步骤 2.3. 检查多重集是否存在于预计算生成的哈

希表中。如果没有，就舍弃这一组密钥猜测值。由于一组错误密钥的匹配概率接近 $2^{8\times 24}\times 2^{-8\times(256-1)}=2^{-1848}$，而且猜测的轮密钥一共有 $\left(2^8\right)^{10}=2^{80}$ 个。因此，一旦有一个密钥猜测值能保留下来，则认为它是正确密钥。

步骤 2.4. 剩余的密钥字节可以通过穷搜索的方法得到。

4.1.2. 7 轮攻击的复杂度分析

7 轮攻击所需的数据复杂度为 $2^{8\times 4}=2^{32}$ 选择明文。预计算的复杂度约为 $2^{184}\times 2^8\times 3/2\times 1/7\approx 2^{189.78}$ 次 7 轮加密运算(计算一个多重集的复杂度相当于 3/2 轮加密运算)。完成步骤 2.1 需要 $2^8\times\left(2^8\right)^5\times 5/16\times 5/4=25\times 2^{42}$ 一轮加密运算；在步骤 2.2 中，为了得到 $\Delta T_{6,0}^i$ 的值，只需知道 T_7^i 的第 0 字节即可。而要得到 $\Delta T_{7,0}^i$，需要计算 $C^i\left(T_8^i\right)$ 的 4 个字节。所以这一步需要

$$2^{40}\times\left[2^8\times\left(2^8\right)^4\times\frac{1}{4}\times\frac{5}{4}+2^8\times 2^{32}\times 2^8\times\frac{1}{16}\right]\approx 2^{84}$$

一轮运算。因此，总的时间复杂度为 $\left(25\times 2^{42}+2^{84}\right)\times 1/7\approx 2^{81.19}$ 次 7 轮 Crypton 加密运算。

4.2. 对 8 轮 Crypton 的中间相遇攻击

在 7 轮的基础上，后面再添 1 轮，构成 8 轮攻击路径，其攻击的结构类似于图 1。不同的是，8 轮攻击时还需要猜测 K_8 全部的 16 字节值，分析过程也相似。这里只给出 8 轮攻击复杂度的分析结果。

8 轮攻击所需的数据复杂度为 $2^{8\times 4}=2^{32}$ 选择明文。预计算的复杂度接近 $2^{184}\times 2^8\times 3/2\times 1/8\approx 2^{189.58}$ 次 8 轮加密运算。总的时间复杂度约为 2^{209} 次 8 轮 Crypton 加密运算。

4.3. 对 9 轮 Crypton 的中间相遇攻击

本节利用区分器 3 来实施 9 轮 Crypton 的中间相遇攻击。在 5 轮区分器的前面添 1 轮，后面添 3 轮，从而构成 9 轮攻击路径，其攻击结构与类似于 7，8 轮的攻击，分析过程也相似。

4.3.1. 9 轮攻击过程

方案一：

步骤 1 预计算阶段

计算由区分器 3 定义的“特殊”多重集的 2^{224} 个可能值，并存储在一个哈希表中。

步骤 2 在线阶段

由第 2 部分的分析可知，分别交换第 7 轮的轮密钥加 K_7 与列混合变换、字节对称变换的复合变换 $\tau\circ\pi$，及第 8 轮的轮密钥加 K_8 与列混合变换、字节对称变换的复合变换 $\tau\circ\pi$ 的顺序，即可以得到 K_7^{eq} 是 K_7 的等价密钥，K_8^{eq} 是 K_8 的等价密钥。

步骤 2.1. 选择 2^{88} 个含有 2^{32} 个与 7 轮攻击定义相同的明文结构，并加密这 2^{120} 个明文。

步骤 2.2. 同 7 轮攻击的步骤 2.1。

步骤 2.3. 猜测 K_7^{eq} 第 0 字节值，K_8^{eq} 第 4 字节(0, 1, 2, 3)值以及 K_9 所有 16 字节值，并部分解密 $P^0,P^1,\cdots,P^{255}$ 对应的密文，得到多重 $\left[\Delta T_{7,0}^0,\Delta T_{7,0}^1,\cdots,\Delta T_{7,0}^{255}\right]$。

步骤 2.4. 检查多重集的值是否在哈希表中出现。如果没有，就删除对应的密钥猜测值。如果在表中，

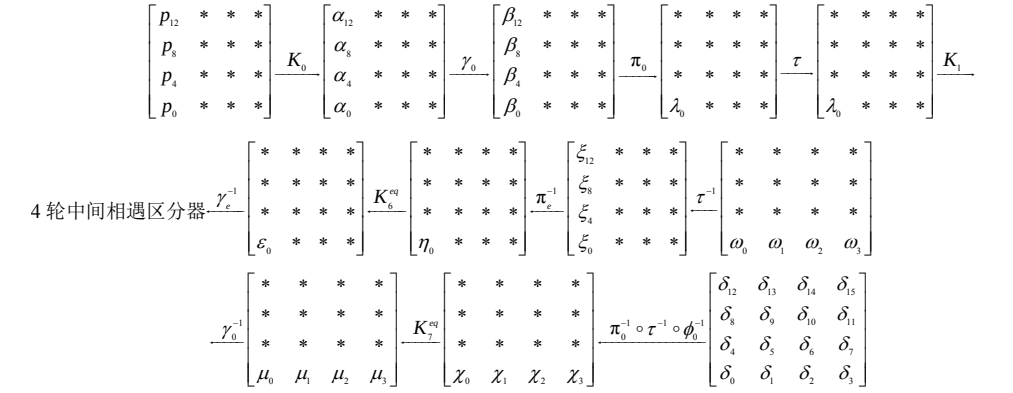

Figure 1. The meet-in-the-middle attack on 7-round Crypton
图 1. 7 轮 Crypton 的中间相遇攻击

则猜测的密钥很可能为正确密钥。原因同 7 轮攻击。

步骤 2.5. 剩余密钥字节可以通过穷搜索的方法得到。

方案二：

步骤 1 预计算阶段

计算由区分器 3 定义的“特殊”多重集的 2^{240} 个可能值，并存储在一个哈希表中。

步骤 2 在线阶段

步骤 2.1. 选择 2^{72} 个含有 2^{32} 个上述明文结构[同方案一]，并加密这 2^{104} 个明文。

步骤 2.2.，2.3.，2.4.，2.5.同方案一。

4.3.2. 9 轮攻击的复杂度分析

方案一：9 轮攻击所需的数据复杂度为 2^{120} 选择明文。预计算的复杂度接近 $2^{224}\times 2^8\times 5/2\times 1/9\approx 2^{230.15}$ 次 9 轮加密运算(计算一个多重集的复杂度相当于 5/2 轮加密运算)。步骤 2.1 需要 2^{120} 次 9 轮 Crypton 加密；步骤 2.2 需要 $2^{120}\times 2^8\times 5/16\times 5/4=25\times 2^{122}$ 一轮加密运算；步骤 2.3 需要

$$\left(2^8\right)^5\times 2^8\times\left(2^8\right)^{16}\times\frac{5}{4}+\left(2^8\right)^5\times 2^8\times 2^{128}\times\left(2^8\right)^4\times\frac{1}{4}$$
$$+\left(2^8\right)^5\times 2^8\times 2^{128}\times 2^{32}\times 2^8\times\frac{1}{16}\approx 2^{212}$$

一轮加密运算，所以总的时间复杂度为 $\left(25\times 2^{122}+2^{212}\right)\times 1/9\approx 2^{208.83}$ 次 9 轮 Crypton 加密运算。

方案二：9 轮攻击所需的数据复杂度为 2^{104} 选择明文。预计算的复杂度接近 $2^{240}\times 2^8\times 5/2\times 1/9\approx 2^{246.15}$ 次 9 轮加密运算(计算一个多重集的复杂度相当于 5/2 轮加密运算)。步骤 2.1 需要 2^{104} 次 9 轮 Crypton 加密；步骤 2.2 需要 $2^{104}\times 2^8\times 5/16\times 5/4=25\times 2^{106}$ 一轮加密运算；步骤 2.3 需要

$$\left(2^8\right)^5\times 2^8\times\left(2^8\right)^{16}\times\frac{5}{4}+\left(2^8\right)^5\times 2^8\times 2^{128}\times\left(2^8\right)^4\times\frac{1}{4}$$
$$+\left(2^8\right)^5\times 2^8\times 2^{128}\times 2^{32}\times 2^8\times\frac{1}{16}\approx 2^{212}$$

一轮加密运算，所以总的时间复杂度为 $\left(25\times 2^{106}+2^{212}\right)\times 1/9\approx 2^{208.83}$ 次 9 轮 Crypton 加密运算。

表 1 给出了本文对 Crypton 算法实施中间相遇攻击及利用其他方法分析时所得的比较结果。通过对比发现：在攻击相同轮数的情况下，中间相遇攻击的计算量有所增加，但有效地降低了攻击所需的数据复杂度。

Table 1. The comparison of the cryptanalysis results of Crypton
表 1. Crypton 算法的分析结果比较

出处	攻击方法	攻击轮数	数据复杂度	时间复杂度	预计算复杂度
文献[5]	不可能差分	7	2^{121}	$2^{116.2}$	-
本文	中间相遇	7	2^{32}	$2^{81.19}$	$2^{189.78}$
本文	中间相遇	8	2^{32}	2^{209}	$2^{189.58}$
文献[6]	相关密钥不可能差分	9	$2^{124.5}$	$2^{176.3}$	-
本文	中间相遇	9	2^{120}	$2^{208.83}$	$2^{230.15}$
文献[6]	相关密钥不可能差分	9	2^{105}	$2^{243.8}$	-
本文	中间相遇	9	2^{104}	$2^{208.83}$	$2^{246.15}$

5. 结束语

本文利用中间相遇攻击的方法，并借助多重集的概念，首次评估了 Crypton 算法对中间相遇攻击的抵抗能力。通过对 Crypton 算法的结构进行分析，构造出该算法的两类 4/5 轮区分器，并由此给出了针对 7/8/9 轮 Crypton 的中间相遇攻击。表 1 给出了一些针对 Crypton 算法的攻击结果。结果显示 9 轮的 Crypton 算法对中间相遇攻击是不免疫的，同现有密码分析方法相比，在攻击相同轮数的情况下，中间相遇攻击的计算量有所增加，但有效地降低了攻击所需的数据复杂度。

参考文献 (References)

[1] C. Lim. Crypton: A new 128-bit block cipher. The First Advanced Encryption Standard Candidate Conference, 1998.

[2] C. Lim. A revised version of crypton-crypton v1.0. Rome: Proceedings of Conference on Fast Software Encryption. Berlin: Springer-Verlag, 1999: 31-45.

[3] G. D'Halluin, G. Bijnens, V. Rijmen, et al. Attack on six rounds of crypton. Rome: Proceedings of Conference on Fast Software Encryption. Berlin: Springer-Verlag, 1999: 46-59.

[4] H. R. Wei, B. Wang. Integral cryptanalysis of reduced-round crypton block cipher. International Symposium on Computer Network and Multimedia Technology, 2009, 2: 792-795.

[5] H. Mala, M. Shakiba and M. Dakhilalian. New impossible differential attacks on reduced-round crypton. Computer Standards & Interfaces, 2010, 32(4): 222-227.

[6] Y. C. Wei, C. Li and B. Sun. Related-key impossible differential cryptanalysis on crypton and crypton v1.0. Xi' an: IEEE International Conference on Signal Processing, Communications and Computing, 2011: 227-232.

[7] H. Diffie, M. Hellman. Exhaustive cryptanalysis of the NBS data encryption standard. IEEE Computer, 1977, 10(6):74-84.

[8] H. Demirci, H. Selcuk. A meet-in-the-middle attack on 8-round

AES. Lausanne: Proceedings of Conference on Fast Software Encryption. Springer-Verlag, 2008: 116-126.
[9] 唐学海, 孙兵, 李超. 对 8 轮 CLEFIA 算法的一种现实攻击[J]. 电子学报, 2011, 39(7): 1608-1612.
[10] 苏崇茂, 韦永壮, 马春波. 10 轮 3D 分组密码算法的中间相遇攻击[J]. 电子与信息学报, 2012, 34(3): 694-697.
[11] 海昕, 唐学海, 李超. 对 Zodiac 算法的中间相遇攻击[J]. 电子与信息学报, 2012, 34(9): 2259-2262.
[12] O. Dunkelman, N. Keller and A. Shanmir. Improved single-key attack on 8-round AES-192 and AES-256. Singapore: Proceedings of Conference on Theory and Application of Cryptology and Information Security, 2010: 158-176.
[13] 杜承航. 分组密码算法 ARIA 的不可能差分分析和中间相遇攻击[D]. 山东大学, 2011.

