doi:10.11959/j.issn.1000-436x.2015062

对 ARIA 算法中间相遇攻击的改进

李曼曼[1,2]，陈少真[1,2]
(1. 解放军信息工程大学 网络空间安全学院，河南 郑州 450001；2. 数学工程与先进计算国家重点实验室，河南 郑州 450001)

摘 要：对 ARIA 算法的结构特征进行了研究，利用“多重集”并结合截断差分的性质，将预计算的参数由 30 个减少到 16 个，构造新的 4 轮中间相遇区分器，有效地改进了 ARIA-192 算法的 7 轮中间相遇攻击。新攻击的预计算复杂度为 $2^{135.3}$，时间复杂度约为 2^{123}。
关键词：分组密码；ARIA 算法；中间相遇攻击；时间复杂度
中图分类号：TN918.1 **文献标识码**：A

Improved meet-in-the-middle attack on ARIA cipher

LI Man-man[1,2], CHEN Shao-zhen[1,2]
(1. Institute of Cyberspace Security, The PLA Information Engineering University，Zhengzhou 450001，China;
2. State Key Laboratory of Mathematical Engineering and Advanced Computing，Zhengzhou 450001，China)

Abstract: A study on the structure of ARIA cipher is presented. A new 4-round distinguishing property for the meet-in-the-middle attack on ARIA cipher is presented by making use of the multiset and the truncated differential characteristic. The new distinguishing property improves the meet-in-the-middle attack on 7 rounds of ARIA-192 cipher effectively by reducing the 30 parameters to 16. The new attack requires a precomputation complexity of $2^{135.3}$ and a time complexity of about 2^{123}.
Key words: block cipher; ARIA cipher; meet-in-the-middle attack; time complexity

1 引言

ARIA 算法[1]是由韩国学者在 2003 年设计的一种分组密码算法，并于 2004 年被选为韩国分组密码标准。由于 ARIA 算法与 AES 算法在设计结构上的相似性，使很多攻击 AES 算法的方法可能对 ARIA 算法产生威胁。但 ARIA 算法的扩散层采用了具有良好扩散效果的对合运算，分支数达到 8，这使 ARIA 算法具有很高的安全性。设计者给出了 ARIA 算法的安全性评估报告[2]，分析了 ARIA 算法的截断差分密码的安全性，导致 ARIA 算法的轮数由最初建议的 10/12/14 轮增加到 12/14/16 轮，而且设计者称 ARIA 算法不存在 4 轮不可能差分。之后，对 ARIA 算法安全性分析的主要结果有：文献[3]找到了一些 4 轮不可能差分，从而对 ARIA-128 进行了 6 轮不可能差分攻击；文献[4]对 6 轮不可能差分攻击提出了改进方案；文献[5]对 ARIA 算法进行了飞来去器攻击；文献[6]对 ARIA-256 提出了 7 轮积分攻击；文献[7]对 ARIA-256 进行了 7 轮不可能差分攻击；文献[8]对 ARIA-192 进行了 7 轮中间相遇攻击，且对 ARIA-256 进行了 8 轮中间相遇攻击；文献[9]对 ARIA-256 进行了 7 轮不可能差分新攻击。

中间相遇攻击这个思想早在 30 年前就被提出，经过长期发展，以及 20 世纪 90 年代差分攻击的提出，很多新的攻击方法也逐渐融合，例如不可能差分攻击就是利用了截断差分在中间相遇产生矛盾的思想而提出来的，其本质也是中间相遇攻击。而中间相遇攻击的本质，就是利用正向与逆向的若干字节或者比特在经过若干轮的

收稿日期：2013-10-31；**修回日期**：2014-01-02
基金项目：信息保障技术重点实验室开放基金资助项目（KJ-13-010）
Foundation Item: Foundation of Science and Technology on Information Assurance Laboratory (KJ-13-010)

数据膨胀之后，中间的数据相遇并找出一个合适的碰撞，然后形成一个有效的攻击。事实上，找到一个有效的碰撞是密码学界一直不懈探讨的问题，而在形成一个碰撞之后，使其涉及猜测的轮子密钥数量尽量少，从而减少攻击的时间复杂度是一个难点问题。本文所要解决的最核心的问题，就是对 ARIA 算法用最少的数据量来构成一个有效的碰撞。

本文主要分析了 ARIA 算法的中间相遇攻击，利用 ARIA 算法的结构特点，采用仅仅存储无序的输出序列的方法，减少猜测参数，进一步利用截断差分的性质，将预计算的参数由 30 个减少到 16 个，构造了新的中间相遇区分器，从而降低了攻击的预计算复杂度，同时也对时间复杂度进行了优化。新的攻击需要 2^{113} 个选择明文，预计算复杂度为 $2^{135.3}$，时间复杂度约为 2^{123}。表 1 给出了改进后的攻击同以往攻击结果的比较。

2 ARIA 加密算法

ARIA 算法的明文分组长度为 128 bit，密钥长度为 128、192 和 256 bit 的可变密钥长度，对应的轮数分别为 12,14,16 轮。轮密钥生成算法使用了一个 256 bit 的 Feistel 密码函数。

2.1 ARIA 算法的轮函数

ARIA 算法的整体结构为 SP(substitution permutation)结构，轮函数由以下 3 个操作构成。

1) 轮密钥加(AK)：将中间状态与 128 bit 轮密钥异或，轮密钥由密钥扩展算法产生。

2) 置换层(SL)： ARIA 算法选用了 2 个 S 盒(S_1 和 S_2)，构成 2 种类型的 SL(SL_1 和 SL_2)，分别在奇数轮和偶数轮使用。2 类置换层 SL_1 和 SL_2 由 2 种 S 盒 S_1 和 S_2 及其逆 S_1^{-1} 和 S_2^{-1} 构成，如图 1 所示。

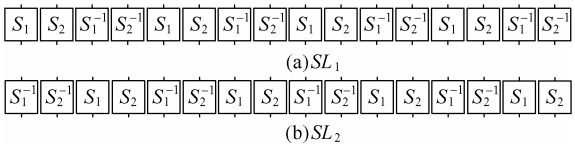


图 1 置换层构成

3) 扩散层(DL)：扩散层是有限域 $(F_2^8)^{16} \to (F_2^8)^{16}$ 上一个对和的线性映射。即为

$$(x_0, x_1, \cdots, x_{15}) \to (y_0, y_1, \cdots, y_{15})$$

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{pmatrix} = \begin{pmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix}$$

最后一轮扩散层用一个轮密钥加代替。

2.2 ARIA 算法的密钥编排方案

ARIA 算法的密钥编排方案由 2 部分组成：初始化阶段与轮密钥产生阶段。

1) 初始化阶段

使用 3 轮 256 bit 的 Feistel 结构，由主密钥 MK 产生 4 个各 128 bit 的 W_0, W_1, W_2, W_3。

表 1 本文与现有 ARIA 攻击结果的比较

文献	对象	攻击方法	轮数	预计算	数据	时间
文献[3]	ARIA-128	不可能差分攻击	6	—	2^{121}	2^{112}
文献[4]	ARIA-128	不可能差分攻击	6	—	2^{120}	2^{96}
文献[9]	ARIA-192	中间相遇攻击	7	2^{187}	2^{120}	$2^{185.3}$
文献[15]	ARIA-192	中间相遇攻击	7	$2^{174.2}$	2^{120}	$2^{184.7}$
本文	ARIA-192	中间相遇攻击	7	$2^{135.3}$	2^{113}	2^{123}
文献[2]	ARIA-256	积分攻击	7	—	$2^{100.6}$	$2^{225.8}$
文献[7]	ARIA-256	不可能差分攻击	7	—	2^{125}	2^{238}
文献[9]	ARIA-256	不可能差分攻击	7	—	2^{120}	2^{219}

2) 轮密钥产生阶段

ARIA-128/192/256 算法的轮数为 12/14/16，分别需要 13/15/17 个轮密钥，每个轮密钥为 128 bit。设轮密钥为 ek_i，$i=1,2,\cdots,17$，其产生方式如下

$$ek_1=(W_0)\oplus(W_1^{>>>19}),ek_2=(W_1)\oplus(W_2^{>>>19}),$$
$$ek_3=(W_2)\oplus(W_3^{>>>19}),ek_4=(W_0)\oplus(W_0^{>>>19})\oplus(W_3)$$
$$ek_5=(W_0)\oplus(W_1^{>>>31}),ek_6=(W_1)\oplus(W_2^{>>>31}),$$
$$ek_7=(W_2)\oplus(W_3^{>>>31}),ek_8=(W_0)\oplus(W_0^{>>>31})\oplus(W_3)$$
$$ek_9=(W_0)\oplus(W_1^{<<<61}),ek_{10}=(W_1)\oplus(W_2^{<<<61}),$$
$$ek_{11}=(W_2)\oplus(W_3^{<<<61}),ek_{12}=(W_0)\oplus(W_0^{<<<61})\oplus(W_3)$$
$$ek_{13}=(W_0)\oplus(W_1^{<<<31}),ek_{14}=(W_1)\oplus(W_2^{<<<31}),$$
$$ek_{15}=(W_2)\oplus(W_3^{<<<31}),ek_{16}=(W_0)\oplus(W_0^{<<<31})\oplus(W_3)$$
$$ek_{17}=(W_0)\oplus(W_1^{<<<19})$$

3　中间相遇攻击

中间相遇攻击是一种选择明文攻击，其思想是由 Diffie 和 Helman 提出的，包含 2 部分：首先建立特殊明文或密文到中间值的一一对应，其次选出满足特殊明文或密文要求的明密文对，再猜测密钥加解密对应的密文或明文得到中间值。如果猜测密钥是正确的，那么中间值是一致的；如果中间值不一致，那么密钥错误，就排除这个错误的猜测密钥。具体描述为：将分组密码看作由 2 部分构成，$C=E_{(K_1,K_2)}(P)=E_{K_2}^2(E_{K_1}^1(P))$，其中 $K_1\in GF(2)^{n_1},K_2\in GF(2)^{n_2}$。设存在 n 个明密文对$(P_1,C_1),(P_2,C_2),\cdots,(P_n,C_n)$，攻击过程如下。

1) 设明文 P_i 的某一个位置输入遍历 256 个所有可能的值(假设第 2 字节，记作 x)，其他字节为固定值，计算 $C_i^*=E_{K_1}^1(P_i)$。C_i^*某个位置的输出(假设第 2 字节，记作 y)可由函数 y 表示: $y=f(x)$，其中 f 由一些固定值和 K_1 决定。

2) 用明文 P_i 可以得到相应的密文 C_i，猜测所有可能的 K_2 计算 $C_i^{**}=E_{K_2}^{2\ -1}(C_i)$，记 C_i^{**}的第 2 字节为 y'，检测是否存在 $y'=y$，筛除不满足这个式子的 K_2。对所有的 i=1, 2,⋯, n 重复以上步骤，过滤掉错误的密钥，最终得到正确密钥。

至今，这种攻击方法已经对许多分组密码算法进行了有效分析。进一步，在过去几年里，这种攻击方法被改进成一种名为原像攻击的方法，并应用于散列函数的攻击，许多新的技术被提出。尽管中间相遇攻击需要大量的预计算复杂度和存储复杂度，但预计算只需要计算一次，而且可以大大降低攻击的时间复杂度。在预计算阶段，参数的个数不能太大，如果参数太大，预计算复杂度就超过了穷举搜索攻击的复杂度。通过减少预计算的复杂度或猜测的密钥量，从而降低攻击的复杂度。中间相遇攻击常常用来攻击 KATAN32[10]、Camellia[11,12]等分组密码算法。

4　中间相遇区分器

中间相遇攻击中的“多重集”是由 Dunkelman 和 Shamir 等人在分析 AES 时引入的[13]，由于 ARIA 算法和 AES 有相似的结构，类比多重集在 AES 分析中的应用[14]，文献[15]给出了针对 ARIA 算法分析的 4 轮“多重集”。

本节对“多重集”的概念作了详细的描述，文献[15]基于这个概念构造了 4 轮中间相遇区分器，对 ARIA 算法进行了有效的攻击。在文献[15]的基础上，本文将预计算的参数由 30 个减少到 16 个，构造了新的中间相遇区分器，并对 ARIA-192 进行了攻击。

在定义多重集前，首先给出 ARIA 算法的一个性质和 δ-集的定义。

性质 1　任取 ARIA 算法的 2 个 S 盒(S_1 和 S_2)之一，给定大量 $S_i(i=1,2)$ 的非零输入差分和输出差分 Δ_i 和 Δ_0，那么等式

$$S(x)+S(x+\Delta_i)=\Delta_0 \tag{1}$$

平均有一个解。这个结果对 S_i^{-1} 同样适用。

证明　如果固定非零的输入差分 Δ_i，取遍输出差分 Δ_0 的 2^8-1 个非零值，那么有 2^7 个输入输出差分使性质 1 中的式子解个数为 0，有 2^7-2 个输入输出差分使性质 1 中的式子解个数为 2，有 1 个输入输出差分使性质 1 中的式子解个数为 4。总的解的个数为$(2^7-2)\times2+4\times1=2^8$，由此可知，等式 $S(x)+S(x+\Delta_i)=\Delta_0$ 平均有（$2^8/(2^8-1)\approx1$）个解。

定义　ARIA 算法的 1 byte 上遍历 256 个所有可能值，其他 15 byte 上取固定值（可以相同），这样的结构称为δ-集，可以表示为 $\{X_m^0,X_m^1,\cdots,X_m^{255}\}$，其中 $X_m^{\ i}$ 是 128 bit，0≤i≤255，m 为 ARIA 算法的轮数。

由定义可以看出，一个δ-集 $\{X_m^0,X_m^1,\cdots,X_m^{255}\}$ 总的分组长度是 $2^8\times2^8=2^{16}$，如果攻击过程中所需

存储的多重集数量大，那么所需的预计算是巨大的，为了减少预计算存储量，引入如下性质。

性质 2 (多重集的存储) 任给一个 2^{16} bit 分组长度的多重集 M，其所有可能的取值数量为 $2^{506.17}$，而 M 仅需要用 512 bit 长度来存储。

证明 不妨设多重集组成的排列为 $(x_0,x_1,x_2,\cdots,x_{255})$，这个排列按照从大到小的顺序排列，当然里面还有一些相等的值，$x_{i+1}\geqslant x_i$ $(i=0,1,\cdots,255)$。利用组合数学的相关结论，可以构造如下等价关系

$$(x_0,x_1,x_2,\cdots,x_{255})\Leftrightarrow(x_0+0,x_1+1,x_2+2,\cdots,x_{255}+255)$$

于是 $(x_0+0,x_1+1,x_2+2,\cdots,x_{255}+255)$ 这个集合是严格递增的，$(x_0+0,x_1+1,x_2+2,\cdots,x_{255}+255)$ 所有可能的取值个数为 $\binom{2^8+2^8-1}{2^8}\approx 2^{506.17}$。

对于任给的多重集 M，在 M 的序列中，表示成 $M=\{x_1^{n_1},x_2^{n_2},\cdots,x_m^{n_m}\}$，其中有 $\sum_{i=1}^{m}n_i=256$，M 的序列中可以表示成如下形式

$$\underbrace{x_1x_1x_1x_1}_{n_1}\mid\underbrace{x_2x_2x_2}_{n_2}\mid\cdots\mid\underbrace{x_mx_mx_mx_m}_{n_m}$$

首先将 256 byte 中不相等的元素放入一个集合 $S=\{x_1,x_2,\cdots,x_m\}$ 中，并将 256 bit 编号为 $e_0=(0,0,\cdots,0)$，…，$e_{255}=(1,1,\cdots,1)$，其中 $e_i\in F_{2^8}$ $(i=0,1,\cdots,255)$。那么就可以将 M 的序列中的 $(x_1,x_2,\cdots,x_m)$ 用 $(e_0,e_1,\cdots,e_{255})$ 以 0 或者 1 对应表出。接下来就是用另外 256 bit 存储 $\{n_1,n_2,\cdots,n_m\}$，第一步是 256 个数值分别对应于 $(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_{256})$，然后采取的操作是将序列 $\{n_1,n_1+n_2,n_1+n_2+n_3,\cdots,\sum_{j=1}^{i}n_j\}$ 用 $(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_{256})$ 以 0 或者 1 对应表出，从而 512 bit 可以表出多重集 M。

文献[15]的工作是由 δ-集的概念得到 ARIA 算法的 4 轮“多重集”，“多重集”由 30 byte 变量决定，并由得到的多重集构造了 ARIA 算法的 4 轮中间相遇区分器，对 ARIA 算法进行了中间相遇攻击，得到了较好的分析结果。

结合本文选取的活动字节，文献[15]的中间相遇区分器可以描述为：中间相遇攻击使用的多重集是 $\{\Delta X_{6,1}^0,\Delta X_{6,1}^1,\cdots,\Delta X_{6,1}^{255}\}$，对应的 δ-集是 $\{X_{2,1}^0,X_{2,1}^1,\cdots,X_{2,1}^{255}\}$（$\delta$-集活动字节是第 1 byte）。对 δ-集进行 4 轮 ARIA 算法加密，（无序的）多重集 $\{\Delta X_{6,1}^0,\Delta X_{6,1}^1,\cdots,\Delta X_{6,1}^{255}\}$ 由状态 $X_3^0(IN)$ 的第 2, 5, 7, 8, 9, 12, 15 byte，状态 $X_4^0(IN)$的全部 16 byte 和状态 $X_5^0(IN)$的第 2, 5, 7, 8, 9, 12, 15 byte 这 30 byte 变量决定。$X_m^i(IN)$表示第 i 个状态在第 m 轮置换层 (SL) 变换前的状态，$X_{m,n}^i$ 表示 X_m^i 的第 n byte。

事实上，精简参数个数可以达到更好的攻击效果，为了进一步改进 ARIA 算法中间相遇攻击，引入下面的重要性质，如图 2 所示。

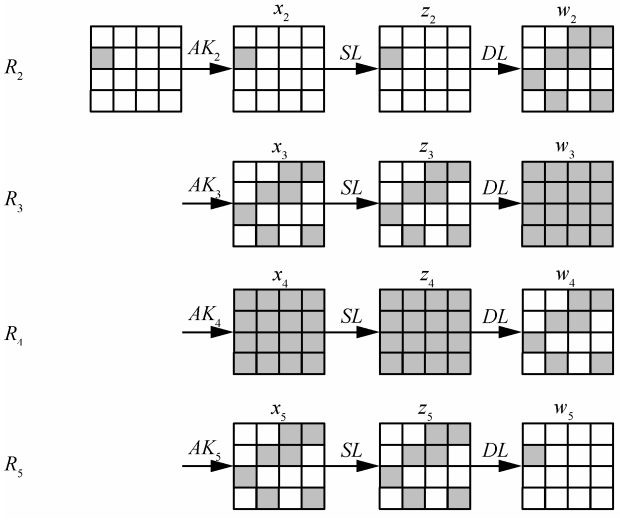


图 2 ARIA 算法的 4 轮中间相遇区分器

性质 3 （新的 4 轮中间相遇区分器）对δ-集进行 4 轮 ARIA 算法加密。如δ-集活动字节是第 1 byte，（无序的）多重集 $\{\Delta X_{6,1}^0,\Delta X_{6,1}^1,\cdots,\Delta X_{6,1}^{255}\}$ 由下列 16 byte 变量决定。

1) $\Delta z_2[1]$；

2) x_3[2, 5, 7, 8, 9, 12, 15]（即状态 X_3^0 (*IN*) 的第 2, 5, 7, 8, 9, 12, 15 byte）；

3) $\Delta w_5[1]$；

4) z_5[2, 5, 7, 8, 9, 12, 15]（即状态 X_5^0 (*IN*) 的第 2, 5, 7, 8, 9, 12, 15 byte）。

其中，$\Delta z_2[1]$是置换层 (SL) 变换后的差分，$\Delta w_5[1]$是扩散层(DL)变换前的差分。

证明 只需要证明通过$\Delta z_2[1]$，x_3[2, 5, 7, 8, 9, 12, 15]，$\Delta w_5[1]$，z_5[2, 5, 7, 8, 9, 12, 15]这 16 byte 来求出状态 $X_3^0(IN)$的第 2, 5, 7, 8, 9, 12, 15 byte，状态 $X_4^0(IN)$的全部 16 byte 和状态 $X_5^0(IN)$的第 2, 5, 7, 8, 9, 12, 15 byte 这 30 byte 即可。

对于上述所有可能的 2^{16} 种取值，由给出的 $\Delta z_2[1]$，可以得到Δx_3[2, 5, 7, 8, 9, 12, 15]，已知 x_3[2, 5, 7, 8, 9, 12, 15]这 7 byte，经过轮函数加密后，可以得到 x_4 的 16 byte 的差分值。同样由给出的

$\Delta w_5[1]$，$z_5[2, 5, 7, 8, 9, 12, 15]$这 8 byte，容易得到 z_4 的 16 byte 的差分值，现在知道 S 盒的输入差分与输出差分，根据性质 1，平均可以得到 x_4 的全部 16 byte 的一个状态值。对应于状态 X_3^0 (*IN*) 的第 2, 5, 7, 8, 9, 12, 15 byte，状态 X_4^0 (*IN*) 的全部 16 byte，状态 X_5^0(*IN*) 的第 2, 5, 7, 8, 9, 12, 15 byte 这 30 byte。

5　攻击过程

本节利用新的 4 轮中间相遇区分器对 7 轮 ARIA 算法进行中间相遇攻击。在新的 4 轮中间相遇区分器的前面加 1 轮，后面加 2 轮，从而构成 7 轮攻击。攻击需要 2^{113} 个选择明文，预计算复杂度为 $2^{135.3}$ 次 7 轮 ARIA 加密运算，时间复杂度约为 2^{123} 次 7 轮 ARIA 加密运算。

5.1　7 轮攻击过程

为了实现对 ARIA 算法的 7 轮攻击，在新的 4 轮中间相遇区分器的前面加 1 轮，后面加 2 轮。由于 AK 变换和 DL 变换的线性特性，交换第 6 轮 DL 变换与 7 轮 AK 变换的次序，得到等价算法，同时可以得到 K_7 的等价密钥 K_7'，如图 3 所示。

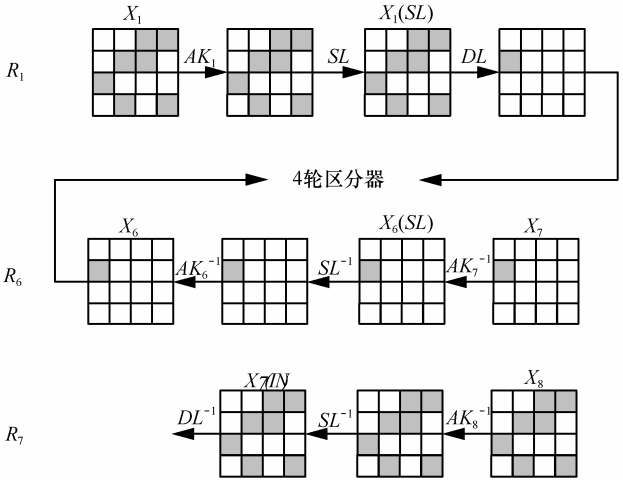


图 2　对 ARIA-192 的中间相遇攻击路径

具体攻击步骤如下。

步骤 1　找到一对使其满足图 3 截断差分路径，其中第 2~5 轮为 7 轮 ARIA 算法中间相遇攻击的预计算路径。

(a) 在第 1 轮和第 6 轮中，扩散层操作均以概率成立，所以需要增加选择明文量，第 1 轮和第 6 轮扩散层操作成立的概率均为 2^{-48}，并且第 7 轮操作成立的概率为 $2^{7\times8}/2^{16\times8}=2^{-72}$。定义一个明文空间结构：满足在字节(2, 5, 7, 8, 9, 12, 15)取遍所有值，其余字节取固定值。一个明文结构有 $2^{56}\times(2^{56}-1)/2=2^{111}$ 对明文差分，那么一共需要选择明文对 $2^{48}\times2^{48}\times2^{72}=2^{168}=2^{111+n}$，$n$=57，所以一共需要选择明文 $2^{57+56}=2^{113}$。

(b) 预计算过程

1) 穷举性质 3 的 16 byte（2^{128} 个所有可能的取值），求出 30 个参数所有可能的值。

2) 根据所求得 30 个参数，求出 $\{\Delta X_{6,1}^0, \Delta X_{6,1}^1, \cdots, \Delta X_{6,1}^{255}\}$，即多重集的个数为

$$2^{128}\times\left(4\times\frac{2^8-1}{(2^8-1)^2}+2\times\frac{(2^8-1)(2^7-1-1)}{(2^8-1)^2}\right)^{16}\approx 2^{128.09} \tag{2}$$

利用性质 2，大约需要 2^{130} 个 128 byte 分组长度的存储。

步骤 2　首先利用步骤 1 中找到的对构造出一个δ-集 $\{X_{2,1}^0, X_{2,1}^1, \cdots, X_{2,1}^{255}\}$：取这对的其中一个元素记作 P^0，找到 P^0 对应的值 $X_{2,1}^0$，然后用 255 个所有可能的非零值与 $X_{2,1}^0$ 异或，得到δ-集的其他 255 个值。猜测 k_1 的 7 byte (2, 5, 7, 8, 9, 12, 15)值，将得到的δ-集的 255 个值代入下面的式子，可以得到相应的 $P^1, P^2, \cdots, P^{255}$ 值。

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

其中，

$$\begin{aligned} p_2 &= S_1(X_{2,1}^i)\oplus k_{1,2} \\ p_5 &= S_2^{-1}(X_{2,1}^i)\oplus k_{1,5} \\ p_7 &= S_2(X_{2,1}^i)\oplus k_{1,7} \\ p_8 &= S_1^{-1}(X_{2,1}^i)\oplus k_{1,8} \\ p_9 &= S_2^{-1}(X_{2,1}^i)\oplus k_{1,9} \\ p_{12} &= S_1^{-1}(X_{2,1}^i)\oplus k_{1,12} \\ p_{15} &= S_2(X_{2,1}^i)\oplus k_{1,15} \end{aligned}$$

其他所有 $p_i(i\in\{0, 1, 3, 4, 6, 10, 11, 13, 14\})$ 取常数。这样就得到了 $P^0, P^1, \cdots, P^{255}$。

然后猜测 k_7' 的第 2 byte 值和 k_8 的 7 byte (2, 5, 7, 8, 9, 12, 15）值，利用猜测的这 8 byte 部分解密 $P^0, P^1, \cdots, P^{255}$ 对应的密文，得到多重集 $\{\Delta X_{6,1}^0,$

$\Delta X_{6,1}^{1},\cdots,\Delta X_{6,1}^{255}\}$。

步骤 3 将计算出来的多重集和预计算中的 $2^{128.09}$ 个值进行比较，如果碰撞成功就恢复出相关的子密钥。对步骤 2 中猜测的 15 byte，一共可以得到 $(2^8)^{15}=2^{120}$ 个多重集。根据性质 2 可知，一个含有 256 byte 的多重集所有取值的个数约为 $2^{506.17}$。由于多重集一共有 $2^{506.17}$ 种取法，错误密钥形成碰撞的概率为 $2^{120}\times 2^{-506.17}\approx 0$。因此认为满足碰撞的密钥就是正确的密钥。

5.2 7 轮攻击复杂度分析

预计算复杂度为 $2^{128.09}\times 2^8\times\frac{4}{7}=2^{135.3}$。

首先加密 $2^{57+56}=2^{113}$ 个选择明文，需要 2^{113} 次 7 轮 ARIA 加密。由构造出来的δ-集计算出多重集，需要的时间复杂度为 $2^{120}\times 2^8\times\frac{1}{16}\times\frac{1}{2}=2^{123}$ 次 7 轮 ARIA 加密（计算一个多重集需要约 1/2 的 7 轮 ARIA 加解密）。因此总的时间复杂度为 $2^{113}+2^{123}\approx 2^{123}$ 次 7 轮 ARIA 加密。

综上可知，改进后的攻击所需明文量为 2^{113} 个选择明文，预计算复杂度为 $2^{135.3}$ 次 7 轮 ARIA 加密运算，时间复杂度约为 2^{123} 次 7 轮 ARIA 加密运算。

6 结束语

本文研究了 ARIA 算法的中间相遇攻击，利用“多重集”和截断差分的性质，优化了多重集的存储过程，将预计算的参数减少到 16 个，提出了新的 4 轮中间相遇区分器，并利用此区分器对 7 轮 ARIA-192 算法的中间相遇攻击做了改进。改进后的攻击数据复杂度为 2^{113}，预计算复杂度和时间复杂度分别约为 $2^{135.3}$ 和 2^{123} 次 7 轮 ARIA 加密运算。改进后的方法降低了攻击的预计算复杂度和时间复杂度。如何构造较长轮数的中间相遇区分器，并结合其他攻击方法，从而得到更好的结果，将是值得进一步研究的工作。

参考文献：

[1] KWON D, KIM J.Specification of ARIA[EB/OL]. http://www.nsri.re.kr/ARIA/doc/ARIA-specification-e.pdf.2003.08.

[2] BIRYUKOV A, CANNIERE C, *et al*. Security and performance analysis of ARIA[EB/OL]. http://cloud.ttongfly.net/t/attachment/1321529635.pdf.2004.07.

[3] WU W, ZHANG W, FENG D. Impossible differential cryptanalysis of reduced-round ARIA and Camellia[J]. Journal of Computer Science and Technology, 2007, 22(3): 449-456.

[4] LI S, SONG C. Improved impossible differential cryptanalysis of ARIA[A]. ISA 2008, IEEE Computer Society[C]. Los Alamitos, 2008.129-132.

[5] FLEISCHMANN E, GORSKI M, LUCKS S. Attacking reduced rounds of the ARIA block cipher[EB/OL]. http://eprint.iacr.org/2009/334.pdf.2009.07.

[6] LI Y, WU W, ZHANG L. Integral attacks on reduced-round ARIA block cipher[A]. ISPEC 2010[C]. 2010. 19-29.

[7] DU C, CHEN J. Impossible differential cryptanalysis of ARIA reduced to 7 rounds[A]. CANS 2010[C]. 2010.20-30.

[8] TANG X, SUN B, LI R. A meet-in-the-middle attack on reduced-round ARIA[J]. Journal of Systems and Software, 2011, 84(10): 1685-1692.

[9] 苏崇茂. 7 轮 ARIA-256 的不可能差分新攻击[J]. 计算机应用, 2012,32(1):45-48.
SU C M. New impossiblediferential attack on 7-round reduced ARIA[J]. Journal of Computer Applications, 2012, 32(1):45-48.

[10] ZHANG W, LIU F, LIU X, MENG S. Differential fault analysis and meet-in-the-middle attack on the block cipher KATAN32[J]. Journal of Shanghai Jiaotong University (Science), 2013, 18(2): 147-152.

[11] LU J, WEI Y. The higher-order meet-in-the-middle attack and its application to the camellia block cipher[A]. INDOCRYPT 2012[C]. 2012. 244-264.

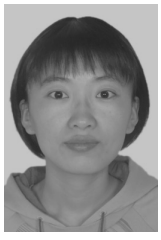
[12] LU J, WEI Y. Meet-in-the-middle attack on reduced versions of the camellia block cipher[A]. IWSEC 2012[C]. 2012. 197-215.

[13] DUNKELMAN O, KELLER N, SHAMIR A. Improved single-key attacks on 8-round AES[A]. ASIACRYPT 2010[C]. 2010. 158-176.

[14] JOHANSSON T, NGUYEN P.Improved key recovery attacks on reduced-round AES in the single-key setting[A]. EUROCRYPT 2013[C]. 2013. 371-387.

[15] DU C, CHEN J. Improved meet-in-the-middle attacks on ARIA[A]. ISAI 2010[C]. 2010.306-310.

作者简介：



李曼曼（1986-），女，河南开封人，解放军信息工程大学硕士生，主要研究方向为信息安全。

陈少真（1967-），女，江苏无锡人，博士，解放军信息工程大学教授、博士生导师，主要研究方向为密码学和信息安全。