國立臺灣大學電機資訊學院電子工程學研究所

碩士論文

Graduate Institute of Electronics Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

利用旁道資訊對 RC6 進行的代數攻擊分析

Algebraic Cryptanalysis of RC6 with Side Channel Information

林珍綺

Chen-Chi Lin

指導教授：鄭振牟 博士

Advisor: Chen-Mou (Doug) Cheng, Ph.D.

中華民國 103 年 8 月

August 2014

摘要

本篇論文詳述一種新的 *CCA*(*Chosen Ciphertext Attack*)攻擊法，可利用旁道攻擊(*Side Channel Attack*)所獲得的資訊，建立減法差分代數式並以 SAT Solver 工具求解。經學理分析驗證，本論文所提方法可在 $O(2^{43})$ 的資料量與 $O(2^{78})$ 的計算複雜度內成功破譯 RC6 最末回合之加密密鑰(*round key*)，並利用末回合密鑰還原對應的漂白密鑰值(*whitening key*)，其計算複雜度僅約 $O(2^{32})$。另外，本論文亦在不同強度之資訊假設（又稱 *oracle*）下，分析攻擊所需資料量與計算複雜度之間的 *trade-off* 關係。

- 關鍵字：RC6、旁道攻擊、減法差分、代數攻擊、破密分析

Abstract

This paper details a novel chosen ciphertext attack on RC6 cipher which has been chosen as one of the finalists for AES competition (March 1999) and has been declared to be resistant to all known cryptanalysis since then. In this paper, it'll be shown that with the aid of side channel information and algebraic analysis the attacker can recover all round keys and whitening keys by using at most $O(2^{43})$ ciphertext pairs and $O(2^{78})$ computations. Moreover, this paper also provides theoretic analysis of the trade-off between different oracles and the general assumption (without any side channel information given), and then proves that the distribution of round key candidates may not be uniformly random.

- **Keyword:** RC6, side channel information, algebraic analysis, chosen ciphertext attack, cryptanalysis

目　錄

圖 目 錄

表 目 錄

第壹章、 簡介

RC6 是 RSA 實驗室為參加 AES(*Advanced Encryption Standard*)徵選所提出的新一代區塊加密演算法，亦為 AES 競賽決選前 5 名之一，其架構承襲了 RC5 的簡約風格，採用硬體執行速度極快的低階運算如「異或和(*xor*)」、「模加法(*modular addition*)」及「循環旋轉(*circular rotation*)」等，可大幅節省實作成本，特別適用於晶片與 *smart card* 等運算資源有限的環境。

RC6 的前身－即 RC5－自 1994 年提出後，學術界已發表多篇論文論證其安全性，唯一可稱為缺陷之處，乃 RC5 具有資料相依旋轉量不完全之現象，幾乎所有理論上可行的攻擊法皆肇因於此，即使如此，迄今仍無任何有效攻擊法被提出。作為 RC5 改良版的 RC6，除了保留 RC5 對各種攻擊的抵抗能力外，另利用「二次函式」與「定量旋轉」之效果，徹底解決 RC5 設計中資料相依旋轉量不完全之缺陷，進一步提昇其安全強度，已知現行任何主流攻擊法如「差分攻擊法」、「線性攻擊法」、「關聯密鑰攻擊法」及「差分－線性攻擊法」等皆無法有效破解之。

在本篇論文中，我們將依據旁道攻擊所能提供之資訊種類訂定各種 *oracle*，並證明在某些特殊 *oracle* 下，若能蒐集足夠多的密文差分，即可利用代數與統計方法成功破譯 RC6 最末兩回合之密鑰（含 *whitening key* 及 *round key*），其原理類似傳統差分攻擊，惟此攻擊法僅需密文對，並不受加密回合數影響，對於採用 *Feistel* 架構的 RC6 而言，此攻擊法之安全威脅比傳統差分攻擊更大。

本篇論文結構如下：第貳章為預備知識，第參章為 RC6 加密演算法簡介與傳統破密分析；第肆章闡述 RC6 的密文現象與攻擊漏洞，並詳述本論文之攻擊方法，包括回合密鑰之破譯與後續 *whitening key* 之還原；第伍章則依據不同 *oracle* 與攻擊所需資料量進行 *trade-off* 分析；最後為總結。

第貳章、 預備知識

第一節、 區塊密

對稱式加密演算法主要分兩大類：串流密與區塊密。前者一次以一個符號（例如一個字元或位元）進行加解密，優點是速度快，可廣泛應用於即時性較強之加密需求，如音訊、視訊等，缺點是易遭代數攻擊法破解。後者則將明文切割成長度為 *n* 位元之區塊，每個區塊獨立加密，經回合函數作用後，產生長度亦為 *n* 位元之密文。

根據 *Shannon* 理論，安全的區塊加密演算法須具備兩個重要功能：「混淆(*confusion*)」與「擴散(*diffusion*)」。前者目的在於隱藏明文、密文與密鑰之間的關係，使攻擊者無法利用統計方法從明密文中推算出密鑰，常見的混淆元件為「替代盒(S-box)」或「資料相依旋轉(*data dependent rotation*)」；後者可將明文的統計特性散布至密文中，使明文之每個位元盡可能地影響密文的每個位元，讓攻擊者無法逐段破譯(*divide and conquer*)，常見的擴散元件為「排列(*permutation*) 」或「可逆線性變換(*invertible linear transform*)」。

第二節、 加密結構

1. *Feistel Network*

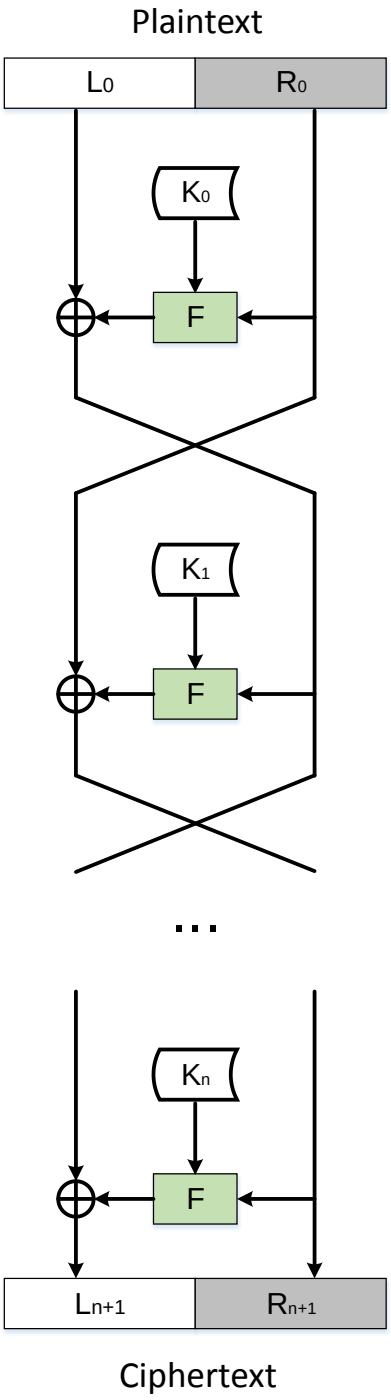
   Feistel Network 是目前最廣為研究的區塊密架構之一，始於 1973 年 *Horst Feistel* 所設計的 *Lucifer* 加密演算法，藉由重複執行基本密碼系統，使最後產生的密碼強度高於單系統產生的結果。其優點在於加解密過程相同，十分利於軟硬體實作，且無須考慮「回合函數(*round function*)」之可逆性，故在函數設計與選擇上較具彈性；缺點則為擴散速度慢，所需回合數較多且耗時，難以達到 「快速雪崩效應(*Fast Avalanche Criteria, FAC*)」。常見的 *Feistel Network* 演算法包括 DES、DEAL、TwiFish 與 RC6 等。

2. *SPN*(*Substitution-Permutation Network*)

SPN 沿襲了 *Shannon* 的想法，將主結構分為兩層：「替換層（*Substitution layer*）」與「排列層（*Permutation layer*）」，以分別達到混淆與擴散的效果。與 *Feistel Network* 不同的是，*SPN* 在架構設計上具有較大彈性，擴散速度亦較快，故所需回合數較少，整體效率較高；缺點是此架構不具可逆性，回合函數無法任意選取，以致加解密演算法不盡相同，實作時往往須用兩種元件。常見的 *SPN* 演算法包括 Serpent 與 SAFER+等。

3. *Square*

Square 為 *Joan Daemen* 與 *Vincent Rijmen* 兩人於 1997 年提出的一個特殊 *SPN* 架構，可將每個區塊以位元組為單位排成矩形，以提供簡潔代數結構與良好運算效能。常見的 *Square* 演算法包括 AES(Rijndael)及 Crypton 等。



【圖 2-2-1】 *Feistel Network* 加密示意圖

第三節、 AES 候選演算法

1. 背景

隨著硬體技術日趨成熟，電腦運算能力亦獲得指數型成長， 對於金鑰長度僅有 56 位元的 DES 演算法而言，其安全性已面臨嚴苛考驗。鑑於此，美國國家技術標準局（*National Institute of Standards and Technology*，簡稱 NIST）於 1997 年 1 月公開徵選新一代加密標準，計畫逐步取代當前主流 DES 演算法，以保護國家機敏等級之資料，此即現今所稱「高級加密標準(*Advance Encryption Standard*, AES)」。

2. 五名決選入圍者

歷時約 2 年的評比階段，NIST 終於在 1998 年 8 月公告出最後五名決選入圍者，即 MARS、RC6、Rijndael、Serpent 與 TwoFish，依序簡介如下：

(1) MARS：採 *Feistel Network* 架構，共加密 32 回合，可支援 128、192 及 256 位元的密鑰長度，提供優於 3-DES 之安全性但速度卻比一次 DES 快許多。此演算法由 IBM 公司提出，設計者之一的 *Don Coppersmith* 當初亦曾參與 DES 加密演算法的設計。

(2) RC6：由 RC5 研改而來，故亦採 *Feistel Network* 架構，可支援 128、192 及 256 位元的密鑰長度，加密回合數為 20，能有效抵擋「差分攻擊」、「線性攻擊」及「關聯密鑰攻擊」等現有已知攻擊。此演算法由 RSA 實驗室提出，適用於運算資源有限的環境。

(3) Rijndael：採 *SPN* 架構，共加密 10 回合，可支援 128、192 及 256 位元的密鑰長度，其設計簡單且代數結構明顯，對於所有已知攻擊具有免疫性，為 AES 加密標準最終勝選者。此演算法由比利時密碼學家 *Joan Daemen* 及 *Vincent Rijmen* 提出，原型即為「Square」演算法。

(4) Serpent：採 *SPN* 架構，共加密 32 回合，可支援 128、192 及 256 位元的密鑰長度，共使用 8 個 4 進 4 出替代盒，其替代盒設計與 DES 類似但具有較佳雪崩效果。由於整體設計理念較保守，故安全性極高，足以抵擋所有已知攻擊，為 AES 最終決選第二名。

(5) TwoFish：前身即為 Blowfish，故亦採 *Feistel Network* 架構，共加密 16 回合，可支援 128、192 及 256 位元的密鑰長度，最大特色是採用「密鑰相依替代盒(*Key -dependent S-box*)」，並可藉由增加記憶體（空間成本）大幅提高運算效能（時間成本），為 AES 最終決選第三名。

第四節、 密碼分析

1. 密碼系統的安全定義

(1) 無條件安全(*Unconditionally Secure*)

在密碼分析者擁有無限計算資源之前提下，不管分析者截獲多少密文，皆無法取得足夠資訊以推出明文內容，如 One-time Pad 密碼系統。

(2) 計算安全(*Computationally Secure*)

以現今計算資源作預測評估，欲破解該密碼系統需耗費數十年或數百年以上（視安全需求而訂）之時間，現今常見密碼系統如：區塊密、公鑰密等多屬此類。

2. 密碼攻擊類型（攻擊強度依序遞增）

(1) 唯密文攻擊(*Ciphertext Only Attack, COA*)

可用資源僅有已截獲之密文，密碼分析者無法任意挑選所需密文，屬於難度最高之攻擊情形。一般攻擊方法為「窮舉密鑰法」，即依序猜測所有可能密鑰並試譯密文，直到出現有意義的明文為止。

(2) 已知明文攻擊(*Known Plaintext Attack, KPA*)

除了已截獲之密文外，密碼分析者另有一些已知的「明密文對」可供參考，例如 Word 文件或 e-mail 的標頭格式皆相同等。

(3) 選擇明文攻擊(*Chosen Plaintext Attack, CPA*)

除了已截獲之密文與已知「明密文對」外，密碼分析者另可選擇有利於攻擊的明文並獲知其對應密文。

(4) 選擇密文攻擊(*Chosen Ciphertext Attack, CCA*)

密碼分析者可任意選擇有利於攻擊的密文並獲知其對應明文，為四種攻擊類型中假設最強的一種。若密碼系統能抵禦選擇密文攻擊，必可抵禦上述三種攻擊。

3. 傳統攻擊方法

(1) 窮舉密鑰攻擊法(*Exhaustive Search Attack*)

又稱「暴力攻擊法」，即一一猜測所有可能的密鑰值，直至破解為止。若密鑰長度為 n 位元，平均而言攻擊者需猜測 2^{n-1} 次才可獲得真正密鑰。

(2) 差分攻擊法(*Differential Attack*)

此攻擊法為一種「選擇明文攻擊」，主要利用明文差分與密文差分之關聯性，試圖找出一條表現不夠隨機的差分路徑（即發生機率較高者）以獲得密鑰位元資訊。此攻擊法除了可對區塊密進行破密分析外，其「差分分析」概念亦適用於串流密、雜湊函數以及替代盒(*S-box*)之安全性分析上，缺點是必須累積大量「明密文對」資料。

(3) 線性攻擊法(*Linear Attack*)

此攻擊法為一種「已知明文攻擊」，主要利用輸入位元線性組合與輸出位元線性組合之關聯性，試圖求解線性方程組以獲得密鑰位元資訊，屬統計型攻擊之一（亦即，此技巧無法保證每次攻擊皆可奏效，僅當兩者線性組合之偏差值(*bias*)夠高時方可適用），故須累積大量「明密文對」資料。與差分攻擊類似的是，其「線性分析」概念亦適用於雜湊函數及替代盒之安全性分析上。

(4) 代數攻擊法(*Algebraic Attack*)

利用代數式表示其加密過程，並將某些已知值代入變數之中，如：密文、初始值或某些狀態值等，再藉由求解代數式而還原出密鑰值。一般而言，其攻

擊效果只略優於暴力攻擊法，或僅適用於特定結構之加密算法；但在破解傳統型 LFSR 串流密時，因 LFSR 的代數結構完整且特徵明顯，故破譯效果極佳。

(5) 關聯密鑰攻擊法(*Related Key Attack*)

假設攻擊者可觀察相同明文在不同密鑰下的運算情形，且得知各密鑰間的關聯為何（但密鑰值未知），例如：「密鑰 K_1 與密鑰 K_2 的末 80 位元皆相同」或「密鑰 K_1 為密鑰 K_2 位移 1 位元的結果」等情形，其所發動的攻擊法皆可稱為關聯密鑰攻擊法。鑒於密鑰交換協議之設計與實作可能存在瑕疵，此攻擊前提並非完全不可能成立。

4. 旁道攻擊法(*Side Channel Attack*)

相較於暴力攻擊法(*Brute force attack*)或學理分析，任何針對硬體實作破綻或系統漏洞而造成資訊外洩情形所進行的攻擊皆可稱為旁道攻擊，如時序、時間差、電力、電磁現象甚或聲音等皆可提供額外資訊，以助密碼分析者破譯整個密碼系統。（利用統計技巧）常見攻擊法包括 *Timing attack*、P*ower monitoring attack*、*Electromagnetic attacks*、*Acoustic cryptanalysis*、*Differential fault* 及 *Data remanence analysis* 等。

第參章、 RC6 加密演算法

第一節、 RC6 規格

RC6 是一個可完全參數化的加密演算法，如字元大小為 w 位元、加密回合數 r 以及金鑰長度為 b 位元組等皆可作為參數調整。其正式寫法為 RC6-w/r/b，表示明文長度共 $4w$ 位元，分拆於 4 個 w 位元暫存器 A、B、C、D 內進行運算，經過 r 回合加密後，最終輸出 $4w$ 位元的密文；另外，金鑰長度共 $8b$ 位元，經密鑰排程演算後，可擴張為 $2r+4$ 把回合子密鑰，分別以 S[0]、…、S[$2r$+3]表示。

第二節、 基本運算

在完整介紹 RC6 加密演算法之前，須先介紹基本運算如下：

【表 3-2-1】RC6 基本運算

運算符號	說明
$a+b$	模加法(mod 2^{32})
$a-b$	模減法(mod 2^{32})
$a \oplus b$	異或和(*bitwise exclusive or*)
$a \times b$	模乘法(mod 2^{32})
$a \lll b$	取 b 最低 5 位元轉成 10 進位，此值即為 a 向左位移旋轉之旋轉量。
$a \ggg b$	取 b 最低 5 位元轉成 10 進位，此值即為 a 向右位移旋轉之旋轉量。

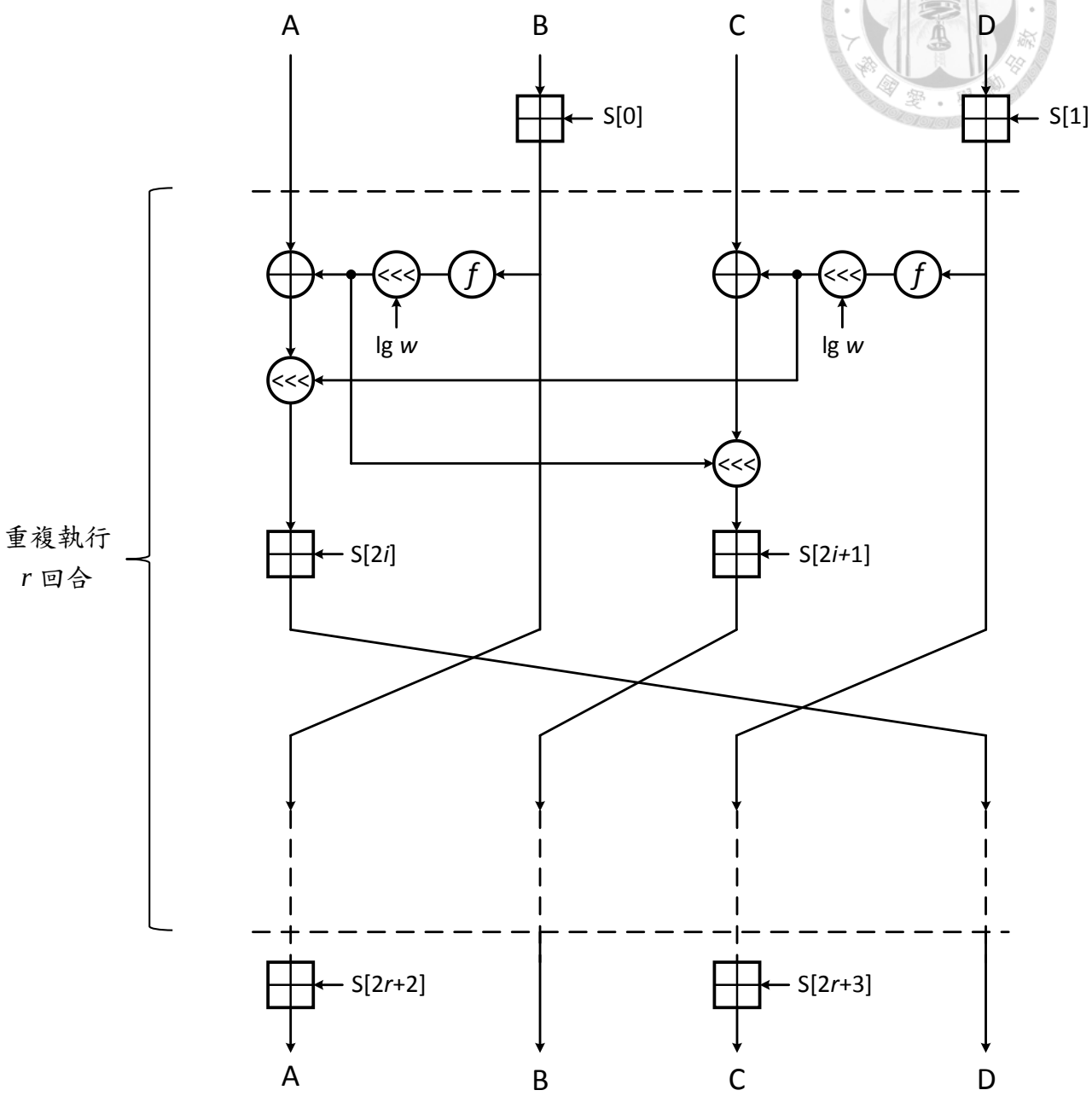
第三節、 虛擬碼與流程示意圖

【表 3-2-2】RC6 加密演算法虛擬碼

輸入	將 128 位元明文(*plaintext*)拆成 4 個字元(*words*)，分別存入 A、B、C、D 共 4 個暫存器之中。
	給定金鑰(128、192 或 256 位元)經密鑰擴張演算後，可得到一組回合密鑰 $S[0]\ldots S[43]$，每把回合密鑰皆為 32 位元，合計 1408 位元。
輸出	暫存器 A、B、C、D 內存密文(*ciphertext*)共 4 字元，合計 128 位元。
回合數	20 回

演算過程	$B = B + S[0]$
	$D = D + S[1]$
	for $i = 1$ to 20 do
	{
	$t = (B \times (2B + 1)) <<< 5$
	$u = (D \times (2D + 1)) <<< 5$
	$A = ((A \oplus t) <<< u) + S[2i]$
	$C = ((C \oplus u) <<< t) + S[2i + 1]$
	$(A, B, C, D) = (B, C, D, A)$
	}
	$A = A + S[42]$
	$C = C + S[43]$

RC6 加密流程示意圖如下，其中 $f(x) = x(2x+1)$ ：



【圖 3-3-1】 RC6 加密流程示意圖

第四節、 傳統安全性分析方法

相較於決選名單中的其它候選者，RC6 的架構簡單且元件單純，可視為兩組 RC5 平行化運算的結果，並藉由「資料相依旋轉(*data dependent rotation*)」將兩者資料充分混和，因此有不少密碼學家將 RC5 視為 RC6 簡化版進行安全分析。然而，與 RC5 相關的破密攻擊分析雖然層出不窮[10, 13, 12, 1, 14, 15]，至今仍然無法找到有效攻擊方法，唯一可稱為缺陷之處，即 RC5 具有資料相依旋轉量不完全之現象，已知幾乎所有理論上可行的攻擊法皆肇因於此。

鑒於 RC5 與 RC6 兩者在結構與元件上的高度相似，RC6 基本上承襲了 RC5 對現行主流攻擊的抵抗能力，如差分攻擊、線性攻擊、關聯密鑰攻擊以及差分－線性攻擊等。但與 RC5 最大不同之處在於，RC6 另利用了「定量旋轉(*fixed rotation*)」與「二次函式(*quadratic function*)」兩元件修正了 RC5 設計中資料相依旋轉量不完全之瑕疵，同時更提高了線性攻擊難度，使其安全性進一步得到強化提昇，即使是目前表現最佳的統計攻擊法[9]，亦無法成功破譯加密 15 回合以上之密文。以下簡述傳統差分攻擊與線性攻擊對 RC6 所進行的破密分析。

1. 差分攻擊

首先，考慮兩組密文差分特徵如下：

$0 \quad 0 \quad e_{t+5} \quad e_t$ $\downarrow$ $0 \quad 0 \quad e_t \quad 0$	$0 \quad 0 \quad 0 \quad e_t$ $\downarrow$ $0 \quad e_s \quad e_t \quad 0$

其中，e_t 表示 2^t，亦即字元中僅第 t 位元為 1 而其它位元皆為 0 之表示法。根據[4]可知，左、右兩邊差分特徵的發生機率分別為 q_t 及 $\rho \times p_t$，今考慮 6 回合的最佳差分路徑：

$$
\begin{array}{cccc}
e_{t+5} & e_t & 0 & 0 \\
 & \downarrow & & \\
e_t & 0 & 0 & 0 \\
 & \downarrow & & \\
0 & 0 & 0 & e_s \\
 & \downarrow & & \\
0 & e_u & e_s & 0 \\
 & \downarrow & & \\
e_u & e_{u-5} & 0 & e_v \\
 & \downarrow & & \\
e_{u-5} & e_{u-10} & e_v & 0 \\
 & \downarrow & & \\
e_{u-10} & e_{u-15} & 0 & 0
\end{array}
$$

欲滿足此路徑，變數t、s、v與u須滿足以下條件：

$$0 \le t,s,v \le 26 \quad \text{且} \quad 15 \le u \le 26$$

可知其發生機率為

$$\rho \times (q_t \times p_s \times p_v \times p_u \times q_{u-5} \times q_{u-10})$$

為使上述差分路徑具有最大發生機率，今選定$t = 11$且$u - 15 = 11$（s與v則可自由取值），則可得差分路徑機率如下：

$$
\begin{aligned}
& \rho^6 \times q_{11} \times \left(\sum_{i=0}^{26} p_i \right)^2 \times p_{26} \times q_{21} \times q_{16} \\
& \approx 2^{-30} \times 2^{-33} \times (2^{-4})^2 \times 2^{-5} \times 2^{-10} \times 2^{-15} \\
& = 2^{-91}
\end{aligned}
$$

假設此路徑連續循環3次（共計$6 \times 3 = 18$回合，尚未滿20回合），則其發生機率為$(2^{-91})^3 = 2^{-273}$，亦即，平均需有2^{273}筆密文差分才可發動有效差分攻擊，然此攻擊成本已遠高於窮舉法（窮舉法平均複雜度僅為$O(2^{127})$）與所有密文差分數量（約$O(2^{255})$），可知RC6加密算法對差分攻擊的抵禦能力極強。

2. 線性攻擊

假設資料相依旋轉 $A = B <<< C$ 的兩種線性逼近式如下：

$$A\cdot,_a = B\cdot,_b \oplus C\cdot,_c$$

$$A\cdot,_a = B\cdot,_b$$

其中，$,_a$ 是由一些字元遮罩(*mask*)所構成的集合，而 $,_b$ 是由集合 $,_a$ 中各元素經循環旋轉（包含 0 位元循環旋轉）後所構成的另一個集合；至於 $,_c$ 則是由最末 5 位元遮罩所構成的集合（否則其偏差值 *bias* 必為 0）。

若集合 $,_a$ 可經由 t 種循環旋轉變成集合 $,_b$，根據[4]可知，第一種線性逼近式的偏差值必為 $bias_I = 2^{-6}$，而第二種線性逼近式的最佳偏差值亦為 $bias_{II} = 2^{-6}$。為降低後續運算（如進位加法或二次函式等）對兩偏差值的弱化效果，以下分析皆以單位元之線性逼近為前提，以求最佳逼近效果。

第一型攻擊

首先，考慮加密 2 回合後之線性逼近：

$$(A\cdot e_t)\oplus(C\cdot e_s) = (A''\cdot e_t)\oplus(C''\cdot e_s)$$

其詳細路徑如下：

$$\begin{array}{cccc} e_t & - & e_s & - \\ & & \downarrow & \\ - & e_u & - & e_v \\ & & \downarrow & \\ e_u & - & e_v & - \end{array}$$

由 *piling-up* 引理可知，上述路徑的偏差值為

$$\alpha_u \times \rho \times \alpha_v \times \rho \times 2^3$$

其中，α_u 與 α_v 分別代表第 u、v 位元的單位元線性逼近在進行進位加法後的偏差值，ρ 則為單位元線性逼近在進行資料相依旋轉後的偏差值。根據[4]可知，當 $u=v=0$ 時，上述偏差值為 2^{-11}；當 u、v 其中之一為 0 而另一方不為 0 且值小於 5 時，偏差值為 2^{-12}；至於 $1 \le u,v \le 4$ 時，偏差值則為 2^{-13}。今考慮 $t=s=u=v=0$ 的最佳情形，則可得到一個 6 回合的線性逼近，其偏差值為 $(2^{-11})^3 \times 2^2 = 2^{-31}$；若為 r 回合的線性逼近，偏差值可估算如下：

$$(2^{-11})^{\left\lfloor \frac{r}{2} \right\rfloor} \times 2^{\left\lfloor \frac{r}{2} \right\rfloor - 1}$$

已知 RC6 共加密 20 回合，而在最佳情形下，其線性逼近偏差值約為

$$\varepsilon_0 = (2^{-11})^{10} \times 2^9 = 2^{-101}$$

已知線性攻擊所需的明文資料與偏差值平方 $\varepsilon_0^{\,2}$ 成正比，可知在 20 回合加密前提下，共需要 $O(2^{202})$ 筆資料才可有效發動攻擊，其成本已遠高於窮舉法與所有明文數量，故 RC6 可有效抵禦第一型線性攻擊。

第二型攻擊

首先，考慮非循環的 6 回合最佳路徑：

$$
\begin{array}{cccc}
e_5 & e_0 & e_5 & e_5 \oplus e_0 \\
 & & \downarrow & \\
- & e_5 & e_5 & e_0 \\
 & & \downarrow & \\
e_5 & e_0 & - & - \\
 & & \downarrow & \\
- & - & - & e_{10} \\
 & & \downarrow & \\
- & - & e_{10} & - \\
 & & \downarrow & \\
- & e_0 & e_5 & - \\
 & & \downarrow & \\
e_0 & e_0 & e_0 & -
\end{array}
$$

根據[4]可知，其線性逼近偏差值為

$$2^{-36} \times 2^{-8} \times 2^{-10} \times 2^{17} = 2^{-37}$$

同理，考慮循環的 6 回合最佳路徑：

$$
\begin{array}{cccc}
e_5 & e_0 & - & - \\
 & & \downarrow & \\
- & - & - & e_{20} \\
 & & \downarrow & \\
- & - & e_{20} & - \\
 & & \downarrow & \\
- & e_5 & e_{15} & - \\
 & & \downarrow & \\
e_5 & e_0 & e_{10} & - \\
 & & \downarrow & \\
- & e_5 & e_5 & e_0 \\
 & & \downarrow & \\
e_5 & e_0 & - & -
\end{array}
$$

其線性逼近偏差值為

$$2^{-36} \times 2^{-9} \times 2^{-23.2} \times 2^{17} \approx 2^{-51}$$

經由各種可能路徑之組合與循環後，可得 14 回合最佳路徑之線性逼近偏差值約為 2^{-106}，可知至少需要 $O(2^{2!2})$ 筆明文資料才可有效發動線性攻擊，由於此成本已遠高於窮舉法與所有明文數量，故 RC6 可有效抵禦第二型線性攻擊。

第肆章、 本論文攻擊方法

第一節、 RC6 的密文現象

假設任兩筆資料在第 $r-1$ 回合之暫存值如下：

A	B	C	D
$\overline{A}$	$\overline{B}$	$\overline{C}$	$\overline{D}$

經過最末回(*r-th inner loop*)加密後可得

B	$[(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1]$	D	$[(A \oplus f(B)_5) <<< f(D)_5] + S[2r]$
$\overline{B}$	$[(\overline{C} \oplus f(\overline{D})_5) <<< f(\overline{B})_5] + S[2r+1]$	$\overline{D}$	$[(\overline{A} \oplus f(\overline{B})_5) <<< f(\overline{D})_5] + S[2r]$

其中

$$f(x)_5 = x(2x+1) <<< 5$$

意即 x 先經過二次函式運算（在模 2^{32} 運算底下），再向左循環旋轉 5 位元之結果。

上述暫存值經過 *whitening* 作用後可得

$B + S[2r+2]$	$[(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1]$	$D + S[2r+3]$	$[(A \oplus f(B)_5) <<< f(D)_5] + S[2r]$
$\overline{B} + S[2r+2]$	$[(\overline{C} \oplus f(\overline{D})_5) <<< f(\overline{B})_5] + S[2r+1]$	$\overline{D} + S[2r+3]$	$[(\overline{A} \oplus f(\overline{B})_5) <<< f(\overline{D})_5] + S[2r]$

狀況 1：假設 $\Delta C = \Delta D = 0$，則此兩筆密文資料分別為

$B + S[2r+2]$	$[(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1]$	$D + S[2r+3]$	$[(A \oplus f(B)_5) <<< f(D)_5] + S[2r]$
$\overline{B} + S[2r+2]$	$[(C \oplus f(D)_5) <<< f(\overline{B})_5] + S[2r+1]$	$D + S[2r+3]$	$[(\overline{A} \oplus f(\overline{B})_5) <<< f(D)_5] + S[2r]$

狀況 2：假設 $\Delta A = \Delta B = 0$，則此兩筆密文資料分別為

$B + S[2r+2]$	$[(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1]$	$D + S[2r+3]$	$[(A \oplus f(B)_5) <<< f(D)_5] + S[2r]$
$B + S[2r+2]$	$[(\overline{C} \oplus f(\overline{D})_5) <<< f(B)_5] + S[2r+1]$	$\overline{D} + S[2r+3]$	$[(A \oplus f(B)_5) <<< f(\overline{D})_5] + S[2r]$

已知狀況 1 與狀況 2 互為對稱情形，以下僅針對狀況 1 進行說明。

- 簡化分析版：無 *whitening* 作用之情形。

已知密文資料分別為

B	$[(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1]$	D	$[(A \oplus f(B)_5) <<< f(D)_5] + S[2r]$
$\overline{B}$	$[(C \oplus f(D)_5) <<< f(\overline{B})_5] + S[2r+1]$	D	$[(\overline{A} \oplus f(\overline{B})_5) <<< f(D)_5] + S[2r]$

首先，透過觀察兩暫存器之值

$$Cipher_B = [(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1]$$

與

$$\overline{Cipher_B} = [(C \oplus f(D)_5) <<< f(\overline{B})_5] + S[2r+1]$$

可發現此二者之減法差分為

$$\begin{aligned} Diff &= Cipher_B - \overline{Cipher_B} \\ &= [(C \oplus f(D)_5) <<< f(B)_5] - [(C \oplus f(D)_5) <<< f(\overline{B})_5] + S[2r+1] - S[2r+1] \\ &= [(C \oplus f(D)_5) <<< f(B)_5] - [(C \oplus f(D)_5) <<< f(\overline{B})_5] \\ &= [(C \oplus f(D)_5) <<< f(B)_5] - [((C \oplus f(D)_5) <<< f(B)_5) <<< (f(\overline{B})_5 - f(B)_5)] \\ &= X - (X <<< offset) \end{aligned}$$

其中

$$X = (C \oplus f(D)_5) <<< f(B)_5$$

且

$$offset = f(\overline{B})_5 - f(B)_5$$

由於 $Diff$ 、$offset$ 與 $X + S[2r+1]$（即 $Cipher_B$）皆已知，若能根據上述關係求得 X 的可能值，則可利用下式

$$\begin{aligned} S[2r+1] &= [(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1] - [(C \oplus f(D)_5) <<< f(B)_5] \\ &= [(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1] - X \\ &= Cipher_B - X \end{aligned}$$

求得回合密鑰 $S[2r+1]$ 的可能值。（同理，狀況 2 可求得回合密鑰 $S[2r]$ 的可能值）

□

➢ 進階分析版：受 *whitening* 作用之情形。

已知密文資料分別為

$B+S[2r+2]$	$[(C\oplus f(D)_5)<<<f(B)_5]+S[2r+1]$	$D+S[2r+3]$	$[(A\oplus f(B)_5)<<<f(D)_5]+S[2r]$
$\overline{B}+S[2r+2]$	$[(C\oplus f(D)_5)<<<f(\overline{B})_5]+S[2r+1]$	$D+S[2r+3]$	$[(\overline{A}\oplus f(\overline{B})_5)<<<f(D)_5]+S[2r]$

首先，透過觀察兩暫存器之值

$$Cipher_B=[(C\oplus f(D)_5)<<<f(B)_5]+S[2r+1]$$

與

$$\overline{Cipher_B}=[(C\oplus f(D)_5)<<<f(\overline{B})_5]+S[2r+1]$$

可發現此二者之減法差分為

$$\begin{aligned}Diff &= Cipher_B-\overline{Cipher_B}\\ &= [(C\oplus f(D)_5)<<<f(B)_5]-[(C\oplus f(D)_5)<<<f(\overline{B})_5]+S[2r+1]-S[2r+1]\\ &= [(C\oplus f(D)_5)<<<f(B)_5]-[(C\oplus f(D)_5)<<<f(\overline{B})_5]\\ &= [(C\oplus f(D)_5)<<<f(B)_5]-[((C\oplus f(D)_5)<<<f(B)_5)<<<(f(\overline{B})_5-f(B)_5)]\\ &= X-(X<<<offset)\end{aligned}$$

其中

$$X=(C\oplus f(D)_5)<<<f(B)_5$$

且

$$offset\in\{0,\ldots,31\}$$

由於 *Diff* 與 $X+S[2r+1]$（即 $Cipher_B$）皆已知，且 *offset* 僅有 32 種可能，若能根據上述關係求得 X 的可能值，則可利用下式

$$
\begin{aligned}
S[2r+1] &= [(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1] - [(C \oplus f(D)_5) <<< f(B)_5] \\
&= [(C \oplus f(D)_5) <<< f(B)_5] + S[2r+1] - X \\
&= Cipher_B - X
\end{aligned}
$$

求得回合密鑰$S[2r+1]$的可能值。（同理，狀況 2 可求得回合密鑰$S[2r]$的可能值）

□

第二節、 攻擊條件(Oracle)

☆Oracle 1：最強假設

假設目前回合為第 t 回合，攻擊者可得知任兩筆資料在第 $t-1$ 回合狀態是否出現「理想碰撞」，亦即，攻擊者可得知第 $t-1$ 回合是否發生 $\Delta C_{t-1} = \Delta D_{t-1} = 0$ 或 $\Delta A_{t-1} = \Delta B_{t-1} = 0$ 之情形。

☆Oracle 2：一般假設

假設目前回合為第 t 回合，攻擊者可得知任兩筆資料在第 $t-1$ 回合狀態的「漢明權重值(*Hamming Weight*)」是否出現碰撞，亦即，攻擊者可得知第 $t-1$ 回合是否發生 $w(A_{t-1}) = w(\overline{A_{t-1}})$ 或 $w(C_{t-1}) = w(\overline{C_{t-1}})$ 之情形。

☆Oracle 3：最弱假設

除了目前回合外，攻擊者無法得知資料在前回合狀態的任何資訊。

第三節、 代數式求解

1. 條件限制與列式

給定 $Diff = (d_{31},...,d_0)$ 與 $offset = k$，欲求解

$$Diff = X - (X <<< offset) \quad X = (x_{31},...,x_0) \in \{0,1\}^{32}$$

等同於求解下式：

$$Diff = X + \overline{X <<< offset} + 1 \quad X = (x_{31},...,x_0) \in \{0,1\}^{32}$$

其中，$\overline{X <<< offset}$ 為 $X <<< offset$ 的補數(*complement*)。為方便求解，上式可改寫成以下聯立條件式，亦即，當上式成立則以下各條件式同時成立，反向亦然：

$$\begin{cases} x_0 \oplus x_{(-k)\bmod 32} = d_0 \\ \sigma_0 \oplus x_1 \oplus x_{(1-k)\bmod 32} \oplus 1 = d_1 \\ \sigma_1 \oplus x_2 \oplus x_{(2-k)\bmod 32} \oplus 1 = d_2 \\ \qquad\qquad \cdots \\ \sigma_{30} \oplus x_{31} \oplus x_{(31-k)\bmod 32} \oplus 1 = d_{31} \\ \sigma_0 = x_0 x_{(-k)\bmod 32} \\ \sigma_1 = \sigma_0 x_1 \oplus \sigma_0 x_{(1-k)\bmod 32} \oplus x_1 x_{(1-k)\bmod 32} \\ \qquad\qquad \cdots \\ \sigma_{30} = \sigma_{29} x_{30} \oplus \sigma_{29} x_{(30-k)\bmod 32} \oplus x_{30} x_{(30-k)\bmod 32} \end{cases}$$

其中 $\sigma_0,...,\sigma_{30}$ 分別為原式加法運算中第 0 ~ 30 個進位位元(*carry bit*)。

★求解工具：可利用 SAT Solver 或 *Gröbner Basis* 求解。

2. 解的分佈情形

☆定理 1

$X-(X<<<k)=D$ 和 $X-(X<<<(n-k))=D$ 有相同的解個數。

【證明】欲證明此定理，須先證明引理 1.1 與引理 1.2。

☆引理 1.1

$X-(X<<<k)=D$ 和 $X-(X<<<k)=-D$ 有相同的解個數。

【證明】首先，假設 x 是 $X-(X<<<k)=D$ 的解，則 x 滿足

$$x-(x<<<k)=D$$

令 $x'=\bar{x}$（即 x 的補數），已知

$$x+\bar{x}=2^n-1$$

且

$$\overline{x<<<k}=\bar{x}<<<k$$

則可推得

$$\bar{x}<<<k=2^n-1-(x<<<k)$$

考慮下式

$$\begin{aligned}x'-(x'<<<k)&=\bar{x}-(\bar{x}<<<k)\\&=(2^n-1-x)-[2^n-1-(x<<<k)]\\&=-[x-(x<<<k)]\\&=-D\end{aligned}$$

可知 x' 必為 $X-(X<<<k)=-D$ 的解。

由於對任意 $x \in \{0,1\}^n$ 而言，必存在唯一的補數 $\bar{x} \in \{0,1\}^n$ 使得

$$x \oplus \bar{x} = (1,..,1)$$

可知 $f(x) = \bar{x}$ 是個一對一函式(*injective function*)。假設 $X - (X <<< k) = D$ 之解集合為 A，$X - (X <<< k) = -D$ 之解集合為 B，根據上述，可推得

$$|A| \le |B| \qquad \cdots\cdots(1)$$

同理，假設 x 是 $X - (X <<< k) = -D$ 的解，則 x 滿足

$$x - (x <<< k) = -D$$

令 $x' = \bar{x}$，考慮下式

$$\begin{aligned} x' - (x' <<< k) &= \bar{x} - (\bar{x} <<< k) \\ &= -[x - (x <<< k)] \\ &= -(-D) \\ &= D \end{aligned}$$

可知 x' 必為 $X - (X <<< k) = D$ 的解。由於 $f(x) = \bar{x}$ 是個一對一函式，故可推得

$$|B| \le |A| \qquad \cdots\cdots(2)$$

根據(1)、(2)之結論，可知 $|B| = |A|$，即 $X - (X <<< k) = D$ 和 $X - (X <<< k) = -D$ 有相同的解個數。

□

☆引理 1.2

$X-(X<<<(n-k))=D$和$X-(X<<<k)=-D$有相同的解個數。

【證明】首先，假設x是$X-(X<<<(n-k))=D$的解，則x滿足

$$x-(x<<<(n-k))=D$$

即

$$x-(x>>>k)=D$$

令$x'=(x>>>k)$，考慮下式

$$\begin{aligned} x'-(x'<<<k) &= (x>>>k)-x \\ &= -[x-(x>>>k)] \\ &= -D \end{aligned}$$

可知x'必為$X-(X<<<k)=-D$的解。

由於對任意$x\in\{0,1\}^n$而言，必存在唯一的$y\in\{0,1\}^n$使得

$$y=(x>>>k)$$

可知$f(x)=(x>>>k)$是個一對一函式。假設$X-(X<<<(n-k))=D$之解集合為C，根據上述，可推得

$$|C|\le|B| \qquad \cdots\cdots(3)$$

同理，假設x是$X-(X<<<k)=-D$的解，則x滿足

$$x-(x<<<k)=-D$$

令$x'=(x<<<k)$，考慮下式

$$\begin{aligned} x'-[x'<<<(n-k)] &= x'-(x'>>>k) \\ &= (x<<<k)-x \\ &= -[x-(x<<<k)] \\ &= -(-D) \\ &= D \end{aligned}$$

可知 x' 必為 $X-(X<<<(n-k))=-D$ 的解。由於對任意 $x\in\{0,1\}^n$ 而言，必存在唯一的 $y\in\{0,1\}^n$ 使得

$$y=(x<<<k)$$

可知 $f(x)=(x<<<k)$ 是個一對一函式，故可推得

$$|B|\le|C| \qquad \cdots\cdots(4)$$

根據(3)、(4)之結論，可知 $|B|=|C|$，即 $X-(X<<<(n-k))=D$ 和 $X-(X<<<k)=-D$ 有相同的解個數。

□

已知 $|B|=|A|$ 且 $|B|=|C|$，可推得 $|A|=|C|$，即 $X-(X<<<k)=D$ 和 $X-(X<<<(n-k))=D$ 有相同的解個數。

□

3. 解的重複性

☆引理 2.1

給定旋轉量 $k \in \{0,...,n-1\}$，則對任意 $x \in \{0,1\}^n$ 而言，必存在唯一減法差分值 $diff \in \{0,1\}^n$ 滿足

$$diff = x - (x <<< k)$$

但反向並不成立。

【證明】給定旋轉量 $k \in \{0,...,n-1\}$，已知對任意 $x \in \{0,1\}^n$ 而言，必存在唯一的 $y \in \{0,1\}^n$ 使得 $y = (x <<< k)$，故可知 $diff = x - (x <<< k) = x - y$ 亦為唯一值。

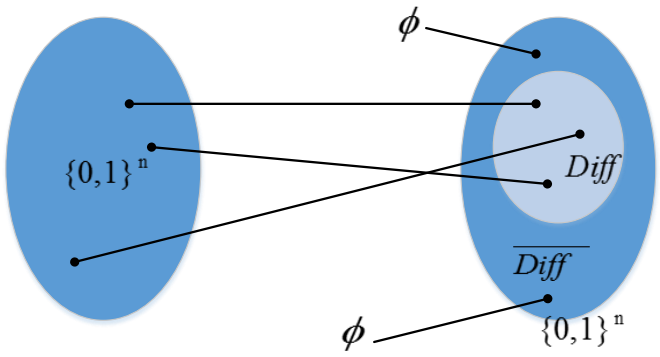
另一方面，若給定旋轉量 $k = 2$ 並考慮 8 位元的減法差分值

$$diff = (1,1,0,1,0,0,0,1)$$

則 $diff = x - (x <<< k)$ 無解，亦即，對任意 $x \in \{0,1\}^n$ 而言，

$$x - (x <<< 2) \neq (1,1,0,1,0,0,0,1)$$

可知並非所有減法差分值 $diff$ 皆可表示成 $x - (x <<< k)$，如下圖所示：



【圖 4-3-1】 解法差分之解空間示意圖

□

【定義 1】

給定減法差分 $diff \in \{0,1\}^n$ 與旋轉量 $k \in \{0,...,n-1\}$，則 $\#X(k, diff)$ 表示其對應的解個數，亦即

$$\#X(k, diff) = \#\{x \in \{0,1\}^n \mid x - (x <<< k) = diff\}$$

☆引理 2.2

給定旋轉量 $k \in \{0,...,n-1\}$，則所有減法差分值 $diff \in \{0,1\}^n$ 之對應解數量 $\#X(k, diff)$ 加總後必為 2^n，亦即

$$\sum_{\forall diff \in \{0,1\}^n} \#X(k, diff) = 2^n$$

【證明】根據引理 2.1，已知

$$\{0,1\}^n = Diff \cup \overline{Diff}$$

其中，$Diff = \{diff \in \{0,1\}^n \mid \exists x \in \{0,1\}^n \ s.t. \ diff = x - (x <<< k)\}$ 且

$\overline{Diff} = \{diff \in \{0,1\}^n \mid \forall x \in \{0,1\}^n \ s.t. \ diff \neq x - (x <<< k)\}$，由於

$$Diff \cap \overline{Diff} = \phi$$

可知

$$\begin{aligned}\sum_{\forall diff \in \{0,1\}^n} \#X(k, diff) &= \sum_{\forall diff \in Diff} \#X(k, diff) + \sum_{\forall diff \in \overline{Diff}} \#X(k, diff) \\ &= \#\{x \in \{0,1\}^n\} + \#\{\phi\} \\ &= 2^n + 0 \\ &= 2^n\end{aligned}$$

□

☆引理 2.3

給定旋轉量 $k \in \{0,...,n-1\}$ 與減法差分值 $diff \in \{0,1\}^n$，若 x 滿足

$$\begin{cases} diff = x - (x <<< k) \\ diff = x - (x <<< (k+1)) \end{cases}$$

則 $x = \{0,...,0\}$ 或 $x = \{1,...,1\}$。

【證明】已知 $diff = x - (x <<< k)$ 且 $diff = x - (x <<< (k+1))$，則兩式相減後可得

$$(x <<< k) - (x <<< (k+1)) = 0$$

亦即

$$(x <<< k) = (x <<< (k+1)) \quad \cdots\cdots(1)$$

令 $x' = (x <<< k+1)$，則式(1)可化簡成

$$x' = (x' >>> 1) \quad \cdots\cdots(2)$$

考慮 $x' = (x_0,...,x_{n-1}) \in \{0,1\}^n$，根據式(2)可得關係式如下：

$$x_i = x_{i+1 (\mathrm{mod}\, n)}$$

亦即

$$\begin{cases} x_0 = x_1 \\ x_1 = x_2 \\ \quad ... \\ x_{n-2} = x_{n-1} \\ x_{n-1} = x_0 \end{cases}$$

可知

$$x_0 = x_1 = ... = x_{n-2} = x_{n-1}$$

故 $x'=\{0,...,0\}$ 或 $x'=\{1,...,1\}$。由於 $x=(x'<<<(k+1))$，可知 $x=\{0,...,0\}$ 或 $x=\{1,...,1\}$。

□

☆引理 2.4

給定旋轉量 $k\in\{0,...,n-1\}$、減法差分值 $diff\in\{0,1\}^n$ 與奇數 odd，若 x 滿足

$$\begin{cases} diff = x-(x<<<k) \\ diff = x-(x<<<(k+odd)) \end{cases}$$

則 $x=\{0,...,0\}$ 或 $x=\{1,...,1\}$。

【證明】已知 $diff = x-(x<<<k)$ 且 $diff = x-(x<<<(k+odd))$，則兩式相減後可得

$$(x<<<k)-(x<<<(k+odd))=0$$

亦即

$$(x<<<k)=(x<<<(k+odd)) \qquad \cdots\cdots(1)$$

令 $x'=(x<<<k+odd)$，則式(1)可化簡成

$$x'=(x'>>>odd) \qquad \cdots\cdots(2)$$

考慮 $x'=(x_0,...,x_{n-1})\in\{0,1\}^n$，根據式(2)可得關係式如下：

$$x_i = x_{i+odd(\mathrm{mod}\,n)}$$

亦即

$$\begin{cases} x_0 = x_{odd(\bmod n)} \\ x_{odd(\bmod n)} = x_{2\cdot odd(\bmod n)} \\ \quad \dots \\ x_{(n-2)\cdot odd(\bmod n)} = x_{(n-1)\cdot odd(\bmod n)} \\ x_{(n-1)\cdot odd(\bmod n)} = x_{n\cdot odd(\bmod n)} = x_0 \end{cases}$$

可知

$$x_0 = x_{odd} = \dots = x_{(n-1)\cdot odd} \qquad \cdots\cdots(3)$$

由於 $n = 2^m$ ，可推得 $\gcd(odd, n) = 1$ ，因此

$$< odd >= \{0, odd, 2\cdot odd, \dots, (n-1)\cdot odd\} = \{0, 1, \dots, n\text{-}1\}$$

亦即，式(3)可改寫成

$$x_0 = x_{odd} = \dots = x_{(n-1)\cdot odd}$$

可知 $x' = \{0,\dots,0\}$ 或 $x' = \{1,\dots,1\}$ 。由於 $x = (x' <<< (k + odd))$ ，可推得 $x = \{0,\dots,0\}$ 或 $x = \{1,\dots,1\}$ 。

□

☆引理 2.5

已知 $n=2^m$，今給定旋轉量 $k\in\{0,...,n-1\}$、減法差分值 $diff\in\{0,1\}^n$ 與正整數 $j\in\{1,...,m\}$，若 x 滿足

$$\begin{cases} diff = x-(x<<<k) \\ diff = x-(x<<<(k+2^j)) \end{cases}$$

則 x 亦滿足 $diff = x-(x<<<(k+2\cdot 2^j))$、$diff = x-(x<<<(k+3\cdot 2^j))\cdots$

、$diff = x-(x<<<(k+(2^{m-j}-1)\cdot 2^j))$。

【證明】已知 $diff = x-(x<<<k)$ 且 $diff = x-(x<<<(k+2^j))$，則兩式相減後可得

$$(x<<<k)-(x<<<(k+2^j))=0$$

亦即

$$(x<<<k)=(x<<<(k+2^j)) \qquad \cdots\cdots(1)$$

令 $x'=(x<<<k+2^j)$，則式(1)可化簡成

$$x'=(x'>>>2^j) \qquad \cdots\cdots(2)$$

考慮 $x'=(x_0,...,x_{n-1})\in\{0,1\}^n$，根據式(2)可得關係式如下：

$$x_i = x_{i+2^j \pmod n}$$

整理可得

$$\begin{cases} x_0 = x_{2^j} = x_{2\cdot 2^j} = ... = x_{(2^{m-j}-2)\cdot 2^j} = x_{(2^{m-j}-1)\cdot 2^j} \\ x_1 = x_{1+2^j} = x_{1+2\cdot 2^j} = ... = x_{1+(2^{m-j}-2)\cdot 2^j} = x_{1+(2^{m-j}-1)\cdot 2^j} \\ \qquad\qquad ..... \\ x_{2^j-1} = x_{(2^j-1)+2^j} = x_{(2^j-1)+2\cdot 2^j} = ... = x_{(2^j-1)+(2^{m-j}-2)\cdot 2^j} = x_{(2^j-1)+(2^{m-j}-1)\cdot 2^j} \end{cases}$$

其中

$$
\begin{aligned}
H_0 &= \{0, 0+2^j, 0+2\cdot 2^j, \ldots, 0+(2^{m-j}-1)\cdot 2^j\},\\
H_1 &= \{1, 1+2^j, 1+2\cdot 2^j, \ldots, 1+(2^{m-j}-1)\cdot 2^j\},\\
&\quad\ldots\ldots\\
H_t &= \{t, t+2^j, t+2\cdot 2^j, \ldots, t+(2^{m-j}-1)\cdot 2^j\},\\
&\quad\ldots\ldots\\
H_{2^j-1} &= \{(2^j-1), (2^j-1)+2^j, (2^j-1)+2\cdot 2^j, \ldots, (2^j-1)+(2^{m-j}-1)\cdot 2^j\}
\end{aligned}
$$

恰形成$\{0,1,\ldots,n-1\}$的一組分割(*partition*)，亦即

$$H_i \cap H_j = \phi, \quad \forall i \neq j$$

且

$$H_0 \cup H_1 \cup \ldots \cup H_{2^j-1} = \{0,1,\ldots,n-1\}$$

由於每一子群H_i皆可指定其值為 0 或 1，故x'共有2^{2^j}種可能值，且皆滿足

$$x' = (x' >>> s\cdot 2^j) \quad , \forall s \in \{0,1,\ldots,\ 2^{m-j}-1\}$$

由於 $x' = (x <<< k+2^j)$，可推得

$$
\begin{aligned}
(x <<< k+2^j) &= (x <<< k+2^j) >>> s\cdot 2^j\\
&= x <<< (k+(1-s)\cdot 2^j)\\
&= x <<< (k+(2^{m-j}+1-s)\cdot 2^j)
\end{aligned}
$$

其中，$2^{m-j}+1-s \in \{0,1,\ldots,2^{m-j}-1\}$。

□

☆引理 2.6

已知 $n=2^m$，當旋轉量 k 為奇數時，可知

$$\#X(k,0)=2$$

而當旋轉量 k 為偶數時，可知

$$\#X(k,0)=2^{\frac{n}{o(k)}}$$

【證明】已知 $x-(x<<<k)=0$，則可得關係式如下：

$$x=(x<<<k) \qquad \cdots\cdots(1)$$

令 $x'=(x<<<k)$，則式(1)可化簡成

$$(x'>>>k)=x' \qquad \ldots\ldots(2)$$

考慮 $x'=(x_0,...,x_{n-1})\in\{0,1\}^n$，根據式(2)可得關係式如下：

$$x_i = x_{i+k(\bmod n)}$$

亦即

$$\begin{cases} x_0 = x_{k(\bmod n)} \\ x_{k(\bmod n)} = x_{2\cdot k(\bmod n)} \\ \quad \ldots \\ x_{(n-2)\cdot k(\bmod n)} = x_{(n-1)\cdot k(\bmod n)} \\ x_{(n-1)\cdot k(\bmod n)} = x_{n\cdot k(\bmod n)} = x_0 \end{cases}$$

若 k 為奇數，根據引理 2.4 可知 $x=\{0,...,0\}$ 或 $x=\{1,...,1\}$，亦即 $\#X(k,0)=2$；若 k 為偶數，根據引理 2.5 可知，集合 $\{0,1,\ldots,n-1\}$ 恰可分割為 $\frac{n}{o(k)}$ 個子群，即 $H_0,H_1,...,H_{n/o(k)-1}$，其中 $o(k)$ 表示 k 的階度(*order*)，由於每一子群 H_i 皆可指定其

值為 0 或 1，可知 x 共有 $2^{\frac{n}{o(k)}}$ 種可能值，亦即 $\#X(k,0)=2^{\frac{n}{o(k)}}$。□

☆引理 2.7

考慮 $n=32$ 之情形，可知

$$\sum_{k=1}^{31}\#X(k,0)=66176$$

【證明】令 H_i 表示 $n=i$ 情形下之對應值，根據引理 2.6 可推得遞迴關係式如下：

$$H_{2i}=2^i+2H_i$$

則可知

$$\begin{aligned}
H_4&=2+4+2=8\\
H_8&=2^4+2H_4=16+2\times 8=32\\
H_{16}&=2^8+2H_8=256+2\times 32=320\\
H_{32}&=2^{16}+2H_{16}=65536+2\times 320=66176
\end{aligned}$$

□

☆引理 2.8

給定旋轉量 $k\in\{0,...,n-1\}$，則對任意減法差分值 $diff\in\{0,1\}^n$ 而言，下式成立：

$$\#X(k,diff)\le\#X(k,0)$$

【說明】根據引理 2.3~2.6 之證明過程可知，一旦旋轉量 $k\in\{0,...,n-1\}$ 選定後，$x=(x_0,...,x_{n-1})\in\{0,1\}^n$ 內部各位元之對應關係即已固定，例如以下關係鏈：

$$x_0 \to x_{k(\mathrm{mod}\, n)} \to x_{2k(\mathrm{mod}\, n)} \to \ldots$$
$$x_1 \to x_{k+1(\mathrm{mod}\, n)} \to x_{2k+1(\mathrm{mod}\, n)} \to \ldots$$
$$x_2 \to x_{k+2(\mathrm{mod}\, n)} \to x_{2k+2(\mathrm{mod}\, n)} \to \ldots$$
$$\ldots\ldots$$

事實上，當減法差分值 $diff = 0$ 時，x 內部位元值並不受其它條件影響，其對應值即為

$$x_0 = x_{k(\mathrm{mod}\, n)} = x_{2k(\mathrm{mod}\, n)} = \ldots$$
$$x_1 = x_{k+1(\mathrm{mod}\, n)} = x_{2k+1(\mathrm{mod}\, n)} = \ldots$$
$$x_2 = x_{k+2(\mathrm{mod}\, n)} = x_{2k+2(\mathrm{mod}\, n)} = \ldots$$
$$\ldots\ldots$$

已知不同關係鏈將形成不同子群，當各子群選定決定值（為 0 或 1）後，子群內其它位元之值亦對應而生，故解數量必為 $2\times 2\times \ldots \times 2 = 2^t$ 之形式。另一方面，當減法差分值 $diff \neq 0$ 時，x 內部位元值將同時受到差分位元與進位位元所控制，在相同關係鏈底下，由於限制條件增加，解空間受到限縮，故其解數量必不超過 $diff = 0$ 之情形（如附錄所示）。

□

第四節、 還原回合密鑰

1. 正解 v.s 錯誤解

☆Oracle 1：

攻擊者可挑選任兩筆資料滿足 $\Delta C_{r-1} = \Delta D_{r-1} = 0$（或 $\Delta A_{r-1} = \Delta B_{r-1} = 0$）之情形，並利用上述攻擊法求解代數式以獲得回合密鑰 $S[2r+1]$（或 $S[2r]$）的可能值，若因 *whitening* 作用使攻擊者無法得知 *offset* ，則須考慮 $offset = 0,...,31$ 的所有情形並進行加總統計。經過持續蒐集、求解並統計各可能值的出現次數後，其次數最高者即為真正回合密鑰。

【說明】欲證明此結論，須先證明定理 2 與定理 3。

☆定理 2：

在 *offset* 已知之前提下，「解出真正回合密鑰」與「發生『理想碰撞』」兩者互為等價關係。

【證明】考慮下式

$$(X_1 <<< k_1) - (X_2 <<< k_2) = X - (X <<< (k_2 - k_1))$$

其中 $X_1 <<< k_1 = Cipher_B - S[2r+1]$、$X_2 <<< k_2 = \overline{Cipher_B} - S[2r+1]$ 且 k_1、k_2 皆已知，則等價關係的「充分性」與「必要性」分別證明如下。

✧ 充分性證明：(發生「理想碰撞」⇒解出真正回合密鑰)

若發生「理想碰撞」，則 $X_1 = X_2$ 且

$$\begin{aligned} Cipher_B - \overline{Cipher_B} &= (X_1 <<< k_1) - (X_2 <<< k_2) \\ &= (X_1 <<< k_1) - (X_1 <<< k_2) \\ &= (X_1 <<< k_1) - ((X_1 <<< k_1) <<< (k_2 - k_1)) \end{aligned}$$

亦即， $(X_1 <<< k_1)$ 是 $Cipher_B - \overline{Cipher_B} = X - (X <<< (k_2 - k_1))$ 的解。

已知

$$(X_1 <<< k_1) = Cipher_B - S[2r+1]$$

可知

$$\begin{aligned} S_{cand} &= Cipher_B - X \\ &= Cipher_B - (X_1 <<< k_1) \\ &= Cipher_B - Cipher_B + S[2r+1] \\ &= S[2r+1] \end{aligned}$$

亦即，若發生「理想碰撞」，則可求出真正回合密鑰。

□

✧ 必要性證明：(發生「理想碰撞」⇐解出真正回合密鑰)

若未發生「理想碰撞」，則 $X_1 \neq X_2$ 。今假設

$$S_{cand} = Cipher_B - X = S[2r+1]$$

已知

$$S[2r+1] = Cipher_B - (X_1 <<< k_1)$$

則

$$Cipher_B - X = Cipher_B - (X_1 <<< k_1)$$

可推得

$$X = (X_1 <<< k_1)$$

考慮

$$Cipher_B - \overline{Cipher_B} = (X_1 <<< k_1) - (X_2 <<< k_2)$$

且

$$\begin{aligned} Cipher_B - \overline{Cipher_B} &= X - (X <<< (k_2 - k_1)) \\ &= (X_1 <<< k_1) - ((X_1 <<< k_1) <<< (k_2 - k_1)) \\ &= (X_1 <<< k_1) - (X_1 <<< k_2) \end{aligned}$$

可推得 $X_1 = X_2$，此結論與前提互為矛盾。

□

綜合上述，可知在 *offset* 已知之前提下，當「理想碰撞」發生時，必可解出真正回合密鑰，反向亦然，故兩者互為等價關係。

□

☆定理 3：

在考慮所有 *offset* 之前提下，「發生『理想碰撞』」為「解出真正回合密鑰」的充分條件而非必要條件。

【證明】考慮下式

$$(X_1 <<< k_1) - (X_2 <<< k_2) = X - (X <<< (k_2 - k_1))$$

其中 $X_1 <<< k_1 = Cipher_B - S[2r+1]$、$X_2 <<< k_2 = \overline{Cipher_B} - S[2r+1]$ 且 k_1、k_2 皆未知，則「充分性」證明如下。

✧ 發生「理想碰撞」=>解出真正回合密鑰：

若發生「理想碰撞」，則 $X_1 = X_2$ 且

$$\begin{aligned} Cipher_B - \overline{Cipher_B} &= (X_1 <<< k_1) - (X_2 <<< k_2) \\ &= (X_1 <<< k_1) - (X_1 <<< k_2) \\ &= (X_1 <<< k_1) - ((X_1 <<< k_1) <<< (k_2 - k_1)) \end{aligned}$$

因 $0 \le (k_2 - k_1) \bmod 32 \le 31$，可知必存在 $k \in \{0,1,...,31\}$ 使得 $(X_1 <<< k_1)$ 是 $Cipher_B - \overline{Cipher_B} = X^{(k)} - (X^{(k)} <<< k)$ 的解。故可知

$$\begin{aligned} S_{cand}(k) &= Cipher_B - X^{(k)} \\ &= Cipher_B - (X_1 <<< k_1) \\ &= Cipher_B - Cipher_B + S[2r+1] \\ &= S[2r+1] \end{aligned}$$

必為真正回合密鑰。

□

✧ 解出真正回合密鑰≠>發生「理想碰撞」：

今假設「發生『理想碰撞』」為「解出真正回合密鑰」的必要條件，則當 $X_1 \neq X_2$ 時，S_{cand} 不可能解出真正回合密鑰。

考慮以下情形：

$$X_2 = X_1 <<< t$$

其中 $t \neq 0$，則

$$\begin{aligned} Cipher_B - \overline{Cipher_B} &= (X_1 <<< k_1) - (X_2 <<< k_2) \\ &= (X_1 <<< k_1) - ((X_1 <<< t) <<< k_2) \\ &= (X_1 <<< k_1) - (X_1 <<< (t + k_2)) \\ &= (X_1 <<< k_1) - ((X_1 <<< k_1) <<< (t + k_2 - k_1)) \end{aligned}$$

因 $0 \le (t + k_2 - k_1) \bmod 32 \le 31$，可知必存在 $k \in \{0,1,...,31\}$ 使得 $(X_1 <<< k_1)$ 是 $Cipher_B - \overline{Cipher_B} = X^{(k)} - (X^{(k)} <<< k)$ 的解。故可知

$$\begin{aligned} S_{cand}(k) &= Cipher_B - X^{(k)} \\ &= Cipher_B - (X_1 <<< k_1) \\ &= Cipher_B - Cipher_B + S[2r+1] \\ &= S[2r+1] \end{aligned}$$

必為真正回合密鑰，此結論與前提互為矛盾。

□

綜合上述，可知在考慮所有 *offset* 之前提下，當「理想碰撞」發生時，必可解出真正回合密鑰，然反向並不成立。

□

- **Case 1：無 *whitening* 作用之情形**

給定 $Diff$ 、 $offset$ 與 $Cipher_B$ ，已知每求解一次代數式

$$Diff = X - (X <<< offset)$$

與

$$S[2r+1] = Cipher_B - X$$

即可求得 $S[2r+1]$ 的可能解集合 S_{cand} ，且對密鑰可能值 x 而言，以下機率式成立：

$$\Pr(x \in S_{cand}) = \begin{cases} 1 & if\ x = S[2r+1] \\ \dfrac{|S_{cand}| - 1}{2^{32} - 1} & if\ x \neq S[2r+1] \end{cases}$$

若連續求解 n 次代數式，得到 n 個可能解集合 $S_{cand}^{(1)}, S_{cand}^{(2)}, ..., S_{cand}^{(n)}$

，則可計算 x 出現次數之期望值 $Exp(\#x)$ 如下：

$$Exp(\#x) = \begin{cases} \sum_{i=1,...,n} 1 = n & if\ x = S[2r+1] \\ \sum_{i=1,...,n} \dfrac{|S_{cand}^{(i)}| - 1}{2^{32} - 1} < n & if\ x \neq S[2r+1] \end{cases}$$

（已知 $|S_{cand}^{(i)}| \leq 2^{16}$ ，可推得 $\sum_{i=1,...,n} \dfrac{|S_{cand}^{(i)}| - 1}{2^{32} - 1} < \sum_{i=1,...,n} \dfrac{1}{2^{16}} = \dfrac{n}{2^{16}} < n$ ；若僅考慮 *offset*

為奇數之情形，則可得 $\left|S_{cand}^{(i)}\right| \le 2$ ）

可知當 n 夠大時，出現次數最高者即為真正回合密鑰值。

□

- **Case 2：受 *whitening* 作用之情形**

給定 $Diff$ 、 $Cipher_B$ 與 $offset = k \in \{0,1,...,31\}$ ，已知每求解一次代數式

$$Diff = X - (X <<< offset)$$

與

$$S[2r+1] = Cipher_B - X$$

即可求得 $S[2r+1]$ 的可能解集合 $S_{cand}(k)$ 。考慮 $k = 0,...,31$ ，令

$$S_{cand} = \underset{k=0,...,31}{\cup} S_{cand}(k)$$

則對密鑰可能值 x 而言，以下機率估計式成立：

$$\Pr(x \in S_{cand}) = \begin{cases} \ge 1 & if\ x = S[2r+1] \\ \le \dfrac{\left|S_{cand}\right| - 1}{2^{32}} & if\ x \ne S[2r+1] \end{cases}$$

（當真正密鑰僅出現 1 次時，等號才成立）

若連續求解 n 次代數式，得到 n 個可能解集合 $S_{cand}^{(1)}, S_{cand}^{(2)}, ..., S_{cand}^{(n)}$ ，則可計算 x 出現次數之期望值 $Exp(\#x)$ 如下：

$$Exp(\#x) = \begin{cases} \ge \sum_{i=1,...,n} 1 = n & if\ x = S[2r+1] \\ \le \sum_{i=1,...,n} \dfrac{\left|S_{cand}^{(i)}\right| - 1}{2^{32}} < n & if\ x \ne S[2r+1] \end{cases}$$

（由於真正密鑰可能出現不只 1 次，故其統計個數應不只 n 個）

已知 $\left|S_{cand}^{(i)}\right| \le 66176$（如引理 2.7 所示；然而若僅考慮 $offset$ 為奇數之情形，則可

得$\left|S_{cand}^{(i)}\right| \leq 32$），可推得

$$\sum_{i=1,\ldots,n} \frac{\left|S_{cand}^{(i)}\right|-1}{2^{32}} \leq \sum_{i=1,\ldots,n} \frac{66175}{2^{32}} = \frac{66175}{2^{32}} \cdot n < n$$

可知當n夠大時，出現次數最高者即為真正回合密鑰值。

□

☆**Oracle 2：**

攻擊者可挑選任兩筆資料

$$(Cipher_A^{(r)}, Cipher_B^{(r)}, Cipher_C^{(r)}, Cipher_D^{(r)})$$

與

$$(\overline{Cipher_A^{(r)}}, \overline{Cipher_B^{(r)}}, \overline{Cipher_C^{(r)}}, \overline{Cipher_D^{(r)}})$$

同時滿足 $Cipher_C^{(r)} = \overline{Cipher_C^{(r)}}$ 且 $w(Cipher_C^{(r-1)}) = w(\overline{Cipher_C^{(r-1)}})$，並利用上述攻擊法求解代數式以獲得回合密鑰 $S[2r+1]$ 的可能值，若因 *whitening* 作用使攻擊者無法得知 *offset*，則須考慮 $offset = 1, 3, ..., 31$ 的所有奇數情形並進行加總統計。經持續蒐集求解且統計各可能值的出現次數後，其次數最高者即為真正回合密鑰。同理，若挑選任兩筆資料滿足 $Cipher_A^{(r)} = \overline{Cipher_A^{(r)}}$ 且 $w(Cipher_A^{(r-1)}) = w(\overline{Cipher_A^{(r-1)}})$，則可求得真正的回合密鑰 $S[2r]$。

【說明】已知 $Cipher_C^{(r)} = \overline{Cipher_C^{(r)}} \Leftrightarrow Cipher_D^{(r-1)} = \overline{Cipher_D^{(r-1)}}$，令 P_k 表示前回合密文 $Cipher_C^{(r-1)}$ 與 $\overline{Cipher_C^{(r-1)}}$ 在漢明權重值皆為 k 之前提下其實際值亦相等的機率，亦即

$$P_k = \Pr[Cipher_C^{(r-1)} = \overline{Cipher_C^{(r-1)}} \mid w(Cipher_C^{(r-1)}) = w(\overline{Cipher_C^{(r-1)}}) = k]$$

依據不同的漢明權重值 k，可得機率分佈表如下：

k	$Cipher_C^{(r-1)}$ 或 $\overline{Cipher_C^{(r-1)}}$ 的可能情形	$\#n$	P_k
0	(00...0)	C_0^{32}	1
1	(10...0),(010...0),...,(00...01)	C_1^{32}	≈ 0.0313
2	(110...0),(1010...0),...,(00...011)	C_2^{32}	≈ 0.0020
3	(1110...0),(11010...0),...,(00...0111)	C_3^{32}	$\approx 2.0161\times10^{-4}$
4	(11110...0),(111010...0),...,(00...01111)	C_4^{32}	$\approx 2.7809\times10^{-5}$
5	(111110...0),(1111010...0),...,(00...011111)	C_5^{32}	$\approx 4.9658\times10^{-6}$
6	(1111110...0),(11111010...0),...	C_6^{32}	$\approx 1.1035\times10^{-6}$
7	(11111110...0),(111111010...0),...	C_7^{32}	$\approx 2.9710\times10^{-7}$
8	(111111110...0),(1111111010...0),...	C_8^{32}	$\approx 9.5072\times10^{-8}$
9	(1111111110...0),(11111111010...0),...	C_9^{32}	$\approx 3.5652\times10^{-8}$
10	(1111111110...0),(11111111010...0),...	C_{10}^{32}	$\approx 1.5501\times10^{-8}$
…	…	...	..
i	(11...10...0),(11...1010...0),...,(00...01...1)	C_i^{32}	$1/C_i^{32}$
…	…	…	…
30	(001...1),(0101...1),...,(11...100)	C_2^{32}	≈ 0.0020
31	(01...1),(101...1),...,(11...10)	C_1^{32}	≈ 0.0313
32	(11...1)	C_0^{32}	1

由於 $C_0^{32} < C_1^{32} < C_2^{32} < ... < C_{16}^{32}$ ，可知當 k 越接近 16 時，$Cipher_C^{(r-1)}$ 與 $\overline{Cipher_C^{(r-1)}}$ 相等的機率即越低，故攻擊者解出正確回合密鑰的機率亦越低。為避免錯誤解累積速度過快，使統計結果不精準（難以有效區分正解與錯誤解），攻擊時僅考慮 *offset* 為奇數之情形，如此即可將 $\left|S_{cand}^{(i)}\right|$ 限制在 2（無 whitening 作用）或 32（受 whitening 作用）以內。

➢ **Case 1：無 *whitening* 作用之情形**

給定 $Diff$ 、$offset$ 、$Cipher_B$ 與漢明權重值 k ，已知每求解一次代數式

$$Diff = X - (X <<< offset)$$

與

$$S[2r+1] = Cipher_B - X$$

即可求得 $S[2r+1]$ 的可能解集合 S_{cand}，且對密鑰可能值 x 而言，以下機率式成立：

$$\Pr(x \in S_{cand}) = \begin{cases} P_k & if\ x = S[2r+1] \\ P_k \cdot \dfrac{(|S_{cand}| - 1)}{(2^{32} - 1)} + (1 - P_k) \cdot \dfrac{|S_{cand}|}{2^{32}} & if\ x \neq S[2r+1] \end{cases}$$

若連續求解 n 次代數式，並求得 n 個可能解集合 $S_{cand}^{(1)}, S_{cand}^{(2)}, ..., S_{cand}^{(n)}$（假設其碰撞機率值分別為 $P_{k_1}^{(1)}, P_{k_2}^{(2)}, ..., P_{k_n}^{(n)}$），則可計算 x 出現次數之期望值 $Exp(\#x)$ 如下：

$$Exp(\#x) = \begin{cases} \sum_{i=1,...,n} P_{k_i}^{(i)} & if\ x = S[2r+1] \\ \sum_{i=1,...,n} [P_{k_i}^{(i)} \cdot \dfrac{(|S_{cand}^{(i)}| - 1)}{(2^{32} - 1)} + (1 - P_{k_i}^{(i)}) \cdot \dfrac{|S_{cand}^{(i)}|}{2^{32}}] & if\ x \neq S[2r+1] \end{cases}$$

亦即

$$Exp(\#x) = \begin{cases} \sum_{i=1,...,n} P_{k_i}^{(i)} & if\ x = S[2r+1] \\ \approx \sum_{i=1,...,n} \dfrac{|S_{cand}^{(i)}| - P_{k_i}^{(i)}}{2^{32}} & if\ x \neq S[2r+1] \end{cases}$$

已知

$$\sum_{i=1,...,n} P_{k_i}^{(i)} \geq \sum_{i=1,...,n} \frac{1}{C_{16}^{32}} = \frac{n}{C_{16}^{32}}$$

根據引理 2.8 可知 $|S_{cand}^{(i)}| \leq 2$ ，故可推得

$$\sum_{i=1,\ldots,n}\frac{\left|S_{cand}^{(i)}\right|-P_{k_i}^{(i)}}{2^{32}}\approx\sum_{i=1,\ldots,n}\frac{\left|S_{cand}^{(i)}\right|}{2^{32}}\le\sum_{i=1,\ldots,n}\frac{2}{2^{32}}<\sum_{i=1,\ldots,n}\frac{1}{C_{16}^{32}}=\frac{n}{C_{16}^{32}}\le\sum_{i=1,\ldots,n}P_{k_i}^{(i)}$$

可知當 n 夠大時，出現次數最高者即為真正回合密鑰值。

□

- **Case 2：受 *whitening* 作用之情形**

給定 $Diff$ 、 $offset=t\in\{1,3,5,\ldots,31\}$ 、 $Cipher_B$ 與漢明權重值 k ，已知每求解一次代數式

$$Diff = X-(X<<<offset)$$

與

$$S[2r+1] = Cipher_B - X$$

即可求得 $S[2r+1]$ 的可能解集合 $S_{cand}(t)$ 。考慮 $t=1,3,5,\ldots,31$ ，令可能密鑰集合 $S_{cand}=\underset{t=1,3,\ldots,31}{\cup}S_{cand}(t)$ ，則對密鑰可能值 x 而言，以下機率式成立：

$$\Pr(x\in S_{cand})=\begin{cases}\frac{1}{2}[P_k+(1-P_k)\cdot\frac{\left|S_{cand}\right|}{2^{32}}] & if\ x=S[2r+1]\\ \frac{1}{2}[P_k\cdot\frac{(\left|S_{cand}\right|-1)}{(2^{32}-1)}+(1-P_k)\cdot\frac{\left|S_{cand}\right|}{2^{32}}] & if\ x\neq S[2r+1]\end{cases}$$

若連續求解 n 次代數式，並求得 n 個可能解集合 $S_{cand}^{(1)},S_{cand}^{(2)},\ldots,S_{cand}^{(n)}$（假設其碰撞機率值分別為 $P_{k_1}^{(1)},P_{k_2}^{(2)},\ldots,P_{k_n}^{(n)}$ ），則可計算 x 出現次數之期望值 $Exp(\#x)$ 如下：

$$Exp(\#x)=\begin{cases}\frac{1}{2}\cdot\sum_{i=1,\ldots,n}[P_{k_i}^{(i)}+(1-P_{k_i}^{(i)})\cdot\frac{\left|S_{cand}^{(i)}\right|}{2^{32}}] & if\ x=S[2r+1]\\ \frac{1}{2}\cdot\sum_{i=1,\ldots,n}[P_{k_i}^{(i)}\cdot\frac{(\left|S_{cand}^{(i)}\right|-1)}{(2^{32}-1)}+(1-P_{k_i}^{(i)})\cdot\frac{\left|S_{cand}^{(i)}\right|}{2^{32}}] & if\ x\neq S[2r+1]\end{cases}$$

已知

$$\frac{1}{2}\cdot\sum_{i=1,\ldots,n}[P_{k_i}^{(i)}+(1-P_{k_i}^{(i)})\cdot\frac{|S_{cand}^{(i)}|}{2^{32}}]\approx\frac{1}{2}\cdot\sum_{i=1,\ldots,n}P_{k_i}^{(i)}\approx\frac{n}{2}\cdot EXP(P_{k_i}^{(i)})=\frac{n}{2\cdot 2^{27}}$$

且

$$\frac{1}{2}\cdot\sum_{i=1,\ldots,n}[P_{k_i}^{(i)}\cdot\frac{(|S_{cand}^{(i)}|-1)}{(2^{32}-1)}+(1-P_{k_i}^{(i)})\cdot\frac{|S_{cand}^{(i)}|}{2^{32}}]\approx\frac{\sum|S_{cand}^{(i)}|}{i=1,\ldots,n\ 2\cdot 2^{32}}$$

其中

$$\sum_{i=1,\ldots,n}|S_{cand}^{(i)}|\approx\sum_{i=1,\ldots,n}EXP(|S_{cand}^{(i)}|)=16\cdot n$$

代入上式可得

$$\frac{1}{2}\cdot\sum_{i=1,\ldots,n}[P_{k_i}^{(i)}\cdot\frac{(|S_{cand}^{(i)}|-1)}{(2^{32}-1)}+(1-P_{k_i}^{(i)})\cdot\frac{|S_{cand}^{(i)}|}{2^{32}}]\approx\frac{\sum|S_{cand}^{(i)}|}{i=1,\ldots,n\ 2\cdot 2^{32}}\approx\frac{16\cdot n}{2\cdot 2^{32}}=\frac{n}{2\cdot 2^{28}}$$

故可推得

$$Exp(\#x_{wrong})<Exp(\#x_{S[2r+1]})$$

可知當n夠大時，出現次數最高者應為真正回合密鑰值。

□

☆Oracle 3：

攻擊者可任意選取滿足 $Cipher_C^{(r)} = \overline{Cipher_C^{(r)}}$ 之密文，並以上述攻擊法求解代數式，經統計工具分析篩濾後，其機率分布情形與「碰撞現象」最吻合(*fitting*)者，即為真正回合密鑰 $S[2r+1]$。同理，若挑選滿足 $Cipher_A^{(r)} = \overline{Cipher_A^{(r)}}$ 之密文進行篩濾，則可求得真正的回合密鑰 $S[2r]$。

【說明】假設每一次試驗時，攻擊者皆取 k 組密文，則根據「生日攻擊法」可知 $Cipher_C^{(r-1)}$ 與 $\overline{Cipher_C^{(r-1)}}$ 兩兩互不碰撞之機率為

$$q^{(k)} = \frac{k! C_k^{2^{32}}}{(2^{32})^k}$$

- **Case 1：無 *whitening* 作用之情形**

已知當 $Cipher_C^{(r-1)}$ 與 $\overline{Cipher_C^{(r-1)}}$ 兩兩互不碰撞時，真正回合密鑰 $S[2r+1]$ 必不出現，故令隨機變數 $X_i = 1$ 表示真正密鑰未出現之情形；而當 $X_i = 0$ 時，真正密鑰至少出現一次。今假設攻擊者進行 n 次獨立實驗(*experiment*)並計算真正回合密鑰之未出現次數 $X = \sum_{i=1}^{n} X_i$，則隨機變數 X 之機率分布情形應符合二項分配(*Binomial Distribution*)，亦即

$$X \sim B(n, q^{(k)})$$

另一方面，錯誤密鑰恰未出現之機率為

$$\overline{q^{(k)}} \approx \left(1 - \frac{1}{2^{32}}\right)^{C_2^k}$$

已知

$$\frac{q^{(k)}}{\overline{q^{(k)}}} \propto k$$

亦即，當 k 越大時，$q^{(k)}$ 與 $\overline{q^{(k)}}$ 之差異越明顯，攻擊者越容易區分出真正密鑰與錯誤密鑰之分布情形。令隨機變數 $X_{cand}^{(k)}$ 表示密鑰可能值 S_{cand} 在「n 次實驗」中的出現次數，若 $X_{cand}^{(k)}$ 之機率分布情形越接近二項分配 $B(n, q^{(k)})$，則 $S_{cand} = S[2r+1]$ 之可能性越高。

□

- **Case 2：受 *whitening* 作用之情形**

考慮 $offset = 1, 3, ..., 31$，可知真正密鑰未出現之機率為

$$Q^{(k)} \approx q^{(k)}\left(1-\frac{16}{2^{32}}\right)^{C_2^k} + \sum_{t=1}^{C_2^k}\left(1-q^{(k)}\right)\cdot\left(\frac{1}{2}\right)^t \cdot\left(1-\frac{16}{2^{32}}\right)^{C_2^k - t}$$

其中，$t \geq 1$ 為實際發生碰撞之密文對數量。另一方面，錯誤密鑰恰未出現之機率為

$$\overline{Q^{(k)}} \approx \left(1-\frac{16}{2^{32}}\right)^{C_2^k}$$

已知

$$\frac{Q^{(k)}}{\overline{Q^{(k)}}} \propto k$$

可知當 k 越大時，$Q^{(k)}$ 與 $\overline{Q^{(k)}}$ 之差異越明顯，攻擊者越容易區分出真正密鑰與錯誤密鑰之分布情形。令隨機變數 $X_{cand}^{(k)}$ 表示密鑰可能值 S_{cand} 在「n 次實驗」中的出

現次數，若 $X_{cand}^{(k)}$ 之機率分布情形越接近二項分配 $B(n,Q^{(k)})$，則 $S_{cand}=S[2r+1]$ 之可能性越高。

□

第五節、 還原漂白密鑰

- Case 1： $S[2r+1]$ 已知

1. 利用 $S[2r+1]$ 對 $Cipher_B$ 與 $\overline{Cipher_B}$ 進行解密，並挑選存在「理想碰撞」的密文對，並記錄其對應的 $Cipher_A$ 、 $\overline{Cipher_A}$ 與 $offset$ 。

2. 依序猜測可能的漂白密鑰值 S_{cand}（最初共有 2^{32} 個可能值，即 $S_{cand}[0,1,...,2^{32}-1]$，此後逐漸遞減），若

$$f(\overline{Cipher_A} - S_{cand}[i])_5 - f(Cipher_A - S_{cand}[i])_5 \neq offset$$

則淘汰 $S_{cand}[i]$。（註： $f(x)_5 = x(2x+1) \bmod 2^5$ ）

3. 重複步驟 1~2，直到所有錯誤密鑰值皆被汰選掉為止（已知每次汰選通過率為 2^{-5} ，則錯誤密鑰 $\overline{S}$ 連續 k 次皆通過汰選之機率為 2^{-5k} ，可知當 k 夠大時，應可汰選掉所有錯誤密鑰值）。

- Case 2： $S[2r]$ 已知

1. 利用 $S[2r]$ 對 $Cipher_D$ 與 $\overline{Cipher_D}$ 進行解密，並挑選存在「理想碰撞」的密文對，並記錄其對應的 $Cipher_C$ 、 $\overline{Cipher_C}$ 與 $offset$ 。

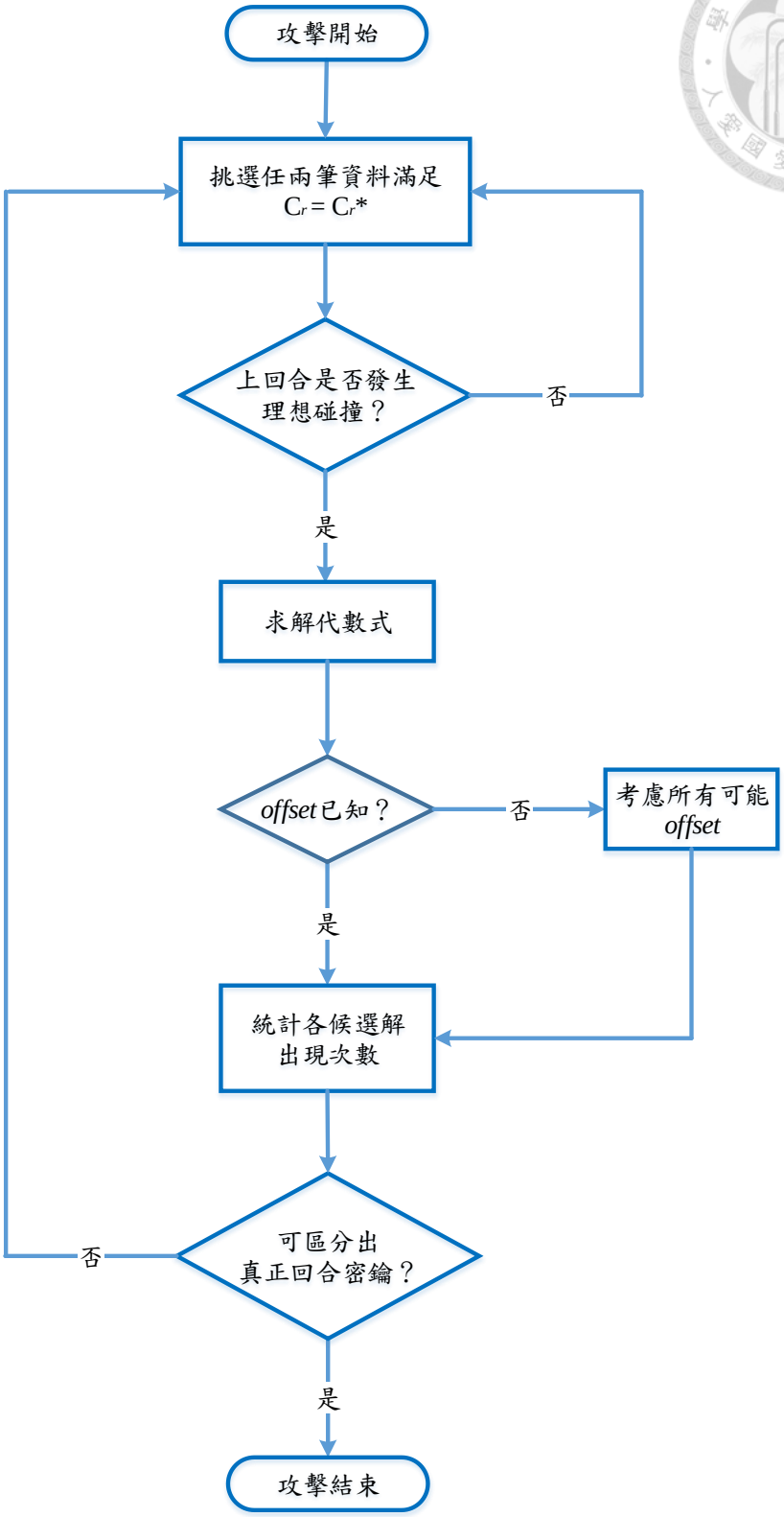
2. 依序猜測可能的漂白密鑰值 S^*_{cand}（最初共有 2^{32} 個可能值，即 $S^*_{cand}[0,1,...,2^{32}-1]$，此後逐漸遞減），若

$$f(\overline{Cipher_C} - S^*_{cand}[i])_5 - f(Cipher_C - S^*_{cand}[i])_5 \neq offset$$

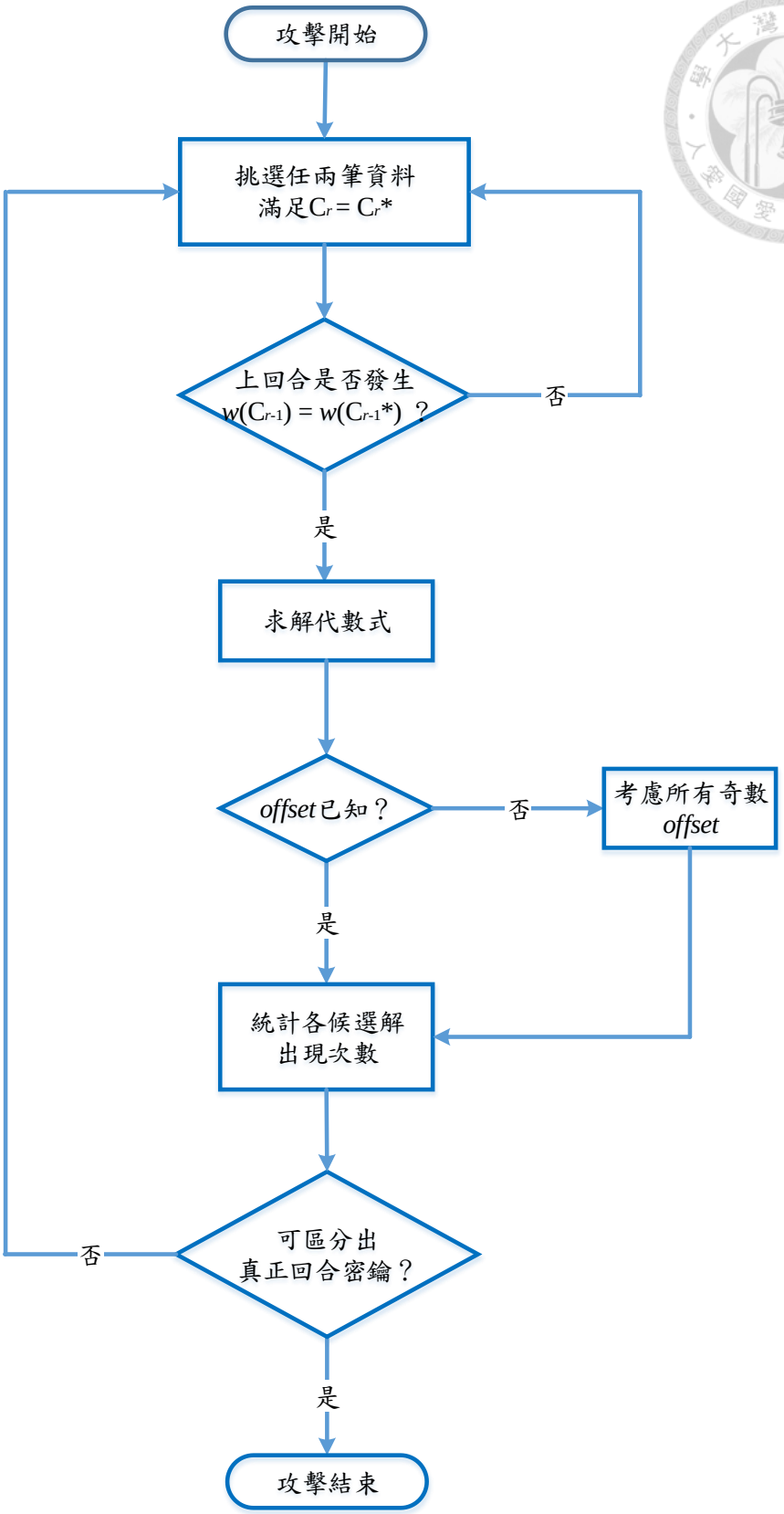
則淘汰 $S^*_{cand}[i]$。（註： $f(x)_5 = x(2x+1) \bmod 2^5$ ）

3. 重複步驟 1~2，直到所有錯誤密鑰值皆被汰選掉為止（已知每次汰選通過率為 2^{-5}，則錯誤密鑰 $\overline{S}$ 連續 k 次皆通過汰選之機率為 2^{-5k}，可知當 k 夠大時，應可汰選掉所有錯誤密鑰值）。

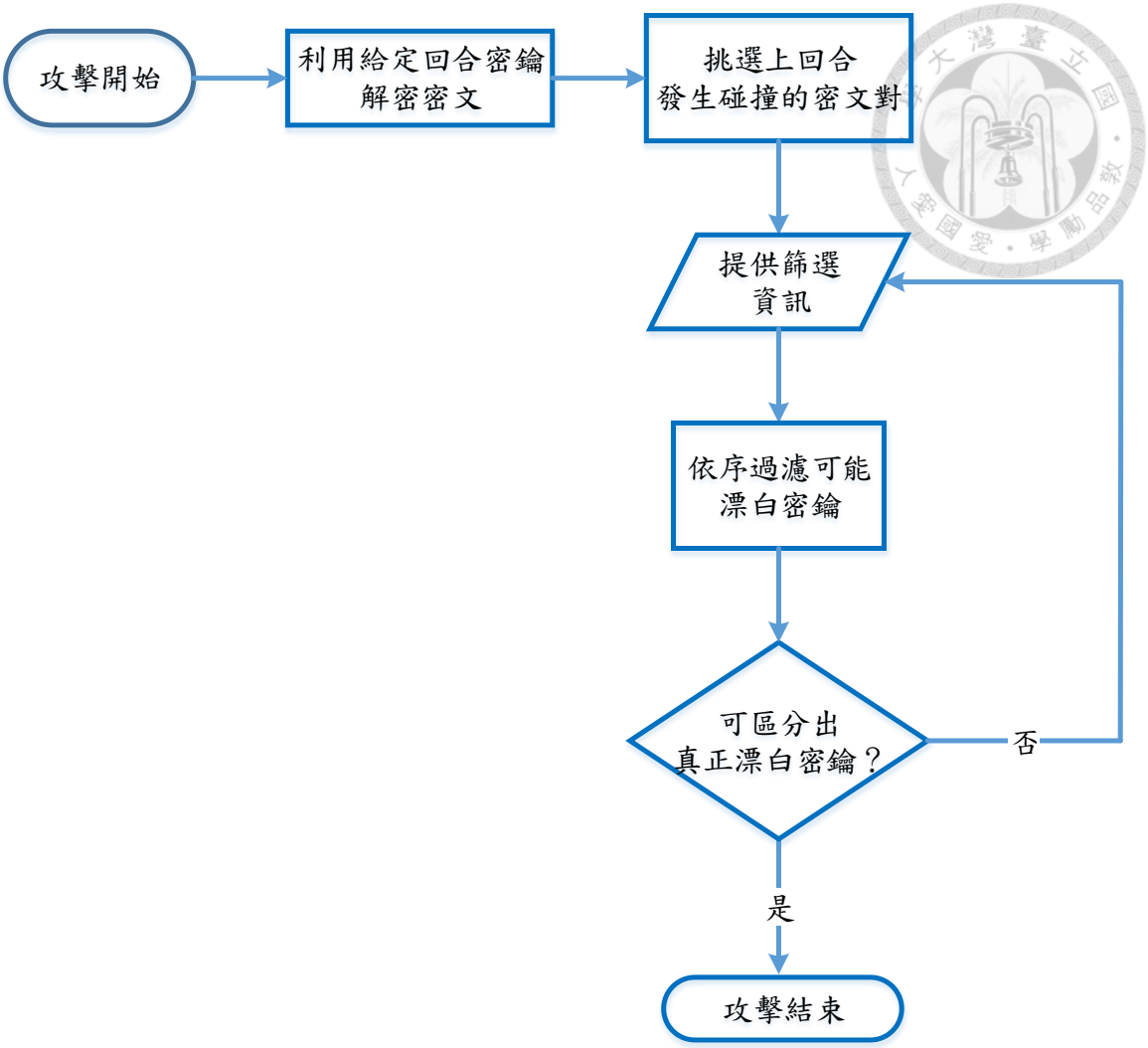
第六節、 攻擊步驟



【圖 6-1-1】 Oracle 1 攻擊流程



【圖 6-1-2】 Oracle 2 攻擊流程



【圖 6-1-3】 漂白密鑰還原流程

第伍章、 複雜度分析

已知還原漂白密鑰之計算複雜度為$O(2^{32})$且可重複使用既有資料，故不另行討論。此外，為統一標準以利分析比較，以下各假設皆考慮 *offset* 為奇數之情形。

- **Case 1：無 *whitening* 作用之情形**

【表 5-1-1】各種 Oracle 前提下出現正解或錯誤解的期望次數（無漂白）

	$Exp(\#x_{true})$	$Exp(\#x_{wrong})$	區別正解/錯誤解 所需資料量估計	Oracle 成立機率
Oracle 1	n	$< \frac{n}{2^{32}}$	$n = 2^6$	2^{-32}
Oracle 2	$\frac{n}{2^{27}}$	$\leq \frac{n}{2^{31}}$	$n = 2^{37}$	$\sum_{i=0}^{32} \frac{(C_i^{32})^2}{2^{64}} \approx 0.0993$
Oracle 3			$\to \infty$	1

根據上表，可估算末回合攻擊時，各假設所需資料量與計算複雜度如下：

【表 5-1-2】各種 Oracle 前提下末回合攻擊所需資料量及計算複雜度（無漂白）

	攻擊所需資料量	計算複雜度
Oracle 1	$2^6 \cdot \frac{1}{2^{-32}} = 2^{38}$	$O(2^6 \cdot 2^{32}) = O(2^{38})$
Oracle 2	$2^{37} \cdot \frac{1}{0.0993} \approx 2^{40.33}$	$O(2^{37} \cdot 2^{32}) = O(2^{69})$
Oracle 3	$\to \infty$	$\to \infty$

已知 RC6 內部共加密 19 回合，則可估算其資料量與計算複雜度如下：

【表 5-1-3】各種 Oracle 前提下攻擊所需總資料量及計算複雜度（無漂白）

	攻擊所需資料量	計算複雜度
Oracle 1	$2^{38}\times 19$	$O(2^{38}\times 19)$
Oracle 2	$2^{40.33}\times 19$	$O(2^{69}\times 19)$
Oracle 3	$\to\infty$	$\to\infty$

- **Case 2：受 *whitening* 作用之情形**

【表 5-1-4】各種 Oracle 前提下出現正解或錯誤解的期望次數（經漂白）

	$Exp(\#x_{true})$	$Exp(\#x_{wrong})$	區別正解/錯誤解 所需資料量估計	Oracle 成立機率
Oracle 1	$\geq n$	$\leq \frac{n}{2^{27}}$	$n=2^6$	2^{-32}
Oracle 2	$\frac{n}{2^{28}}$	$\frac{n}{2^{29}}$	$n=2^{42}$	$\sum_{i=0}^{32}\frac{(C_i^{32})^2}{2^{64}}\approx 0.0993$
Oracle 3			$\to\infty$	1

根據上表，可估算末回合攻擊時，各假設所需資料量與計算複雜度如下：

【表 5-1-5】各種 Oracle 前提下末回合攻擊所需資料量及計算複雜度（經漂白）

	攻擊所需資料量	計算複雜度
Oracle 1	$2^6\cdot\frac{1}{2^{-32}}=2^{38}$	$O(2^6\cdot 2^{32}\cdot 2^4)=O(2^{42})$
Oracle 2	$2^{42}\cdot\frac{1}{0.0993}\approx 2^{45.33}$	$O(2^{42}\cdot 2^{32}\cdot 2^4)=O(2^{78})$
Oracle 3	$\to\infty$	$\to\infty$

根據 Case 1 與 Case 2 之分析，可估算整體所需資料量與計算複雜度如下：

【表 5-1-6】各種 Oracle 前提下攻擊所需總資料量及計算複雜度

	攻擊所需資料量	計算複雜度
Oracle 1	$O(2^{43})$	$O(2^{43})$
Oracle 2	$O(2^{46})$	$O(2^{78})$
Oracle 3	$\to\infty$	$\to\infty$

第陸章、 總結

根據本論文提出之代數攻擊法可知，旁道攻擊法所提供的「內部碰撞」訊息，確可有效破解 RC6 最末回合之加密密鑰 $S[2r+1]$（或 $S[2r]$），依其假設強度，所需資料量分別為 $O(2^{43})$ 與 $O(2^{46})$，計算複雜度則為 $O(2^{43})$ 與 $O(2^{78})$；而在最末回合密鑰已知之前提下，利用本論文提出之篩濾法亦可有效還原對應之漂白密鑰 $S[2r+2]$（或 $S[2r+3]$），其計算複雜度約為 $O(2^{32})$。雖然在最弱假設（無任何旁道資訊）下，本論文方法尚無法在合理時間內精準篩濾出真正回合密鑰，然以機率統計角度而言，仍足以證明其密鑰可能值之分布並非完全均勻隨機。

第柒章、 參考文獻

[1] *B. S. Kaliski Jr., Y. L. Yin, On the Security of the RC5 Encryption Algorithm", RSA Laboratories Technical Report TR-602, Version 1.0 - September 1998.*

[2] *E. Biham and A. Shamir. Differential Cryptanalysis of the Data Encryption Standard. Springer-Verlag, New York, 1993.*

[3] *R.L. Rivest, M.J.B. Robshaw, R. Sidney and Y.L. Yin, The RC6 Block Cipher", v1.1, August 20, 1998.*

[4] *S. Contini, R.L. Rivest, M.J.B. Robshaw and Y.L. Yin. The Security of the RC6 Block Cipher. v.1.0, August 20, 1998. Available at www.rsa.com/rsalabs/aes/.*

[5] *S. Contini, R.L. Rivest, M.J.B. Robshaw, and Y.L. Yin. Some Comments on the First Round AES Evaluation of RC6. Available at http://csrc.nist.gov/encryption/aes/round1/pubcmnts.htm.*

[6] *S. Contini, R.L. Rivest, M.J.B. Robshaw, and Y.L. Yin. Improved analysis of some simplified variants of RC6. In L. Knudsen, editor, Fast Software Encryption, Sixth International Workshop, Rome, Italy, March 1999, LNCS 1636, pages 1–15. Springer Verlag, 1999.*

[7] *S. Contini and Y.L. Yin. On differential properties of data dependent rotations and their use in Mars and RC6. Presented at the 2nd AES conference, see www.nist.gov/aes.*

[8] *J. Daemen, L. Knudsen, and V. Rijmen. The block cipher Square. In E. Biham, editor, Fast Software Encryption, Fourth International Workshop, Haifa, Israel, January 1997, LNCS 1267, pages 149–165. Springer Verlag, 1997.H. Gilbert, H. Handschuh, A. Joux, and S. Vaudenay. A Statistical Attack on RC6. In B. Schneier,*

editor, Fast Software Encryption, Seventh International Workshop. Springer Verlag, 2001. To appear.

[9] *A. Biryukov and E. Kushilevitz. Improved cryptanalysis of RC5. In K. Nyberg, editor, Advances in Cryptology Eurocrypt '98, volume 1403 Lecture Notes in Computer Science, pages 85-99, 1998. Springer Verlag.*

[10] *M. Matsui, The first experimental cryptanalysis of the Data Encryption Standard". In Advances in Cryptology - Crypto'94, pp 1-11, Springer Verlag, New York, 1994.*

[11] *B.S. Kaliski and Y.L. Yin. On differential and linear cryptanalysis of the RC5 encryption algorithm. In D. Coppersmith, editor, Advances in Cryptology Crypto '95, volume 963 of Lecture Notes in Computer Science, pages* 171-184, 1995. *Springer Verlag.*

[12] *M.H. Heys. Linearly weak keys of RC5. IEE Electronic Letters, Vol. 33, pages 836-838,* 1997.

[13] *L.R. Knudsen and W. Meier. Improved differential attacks on RC5. In N. Koblitz, editor, Advances in Cryptology - Crypto '96, volume 1109 of Lecture Notes in Computer Science, pages 216-228, 1996. Springer Verlag.*

[14] *A. A. Selcuk. New results in linear cryptanalysis of RC5. In S. Vaudenay, editor, Fast Software Encryption, volume 1372 of Lecture Notes in Computer Science, pages 1-16, 1998, Springer-Verlag.*

第捌章、 附錄

diff \offset	1	2	3	4
0	2	4	2	16
1	1	0	1	0
2	1	0	1	0
3	1	3	1	0
4	1	0	1	0
5	1	0	1	0
6	1	2	1	0
7	1	1	1	0
8	0	0	0	0
9	1	1	1	0
10	1	2	1	0
11	1	0	1	0
12	1	0	1	0
13	1	3	1	0
14	1	0	1	0
15	1	0	1	0

diff \offset	1	2	3	4	5	6	7	8
0	2	4	2	16	2	4	2	256
1	1	0	1	0	1	0	1	0
2	1	0	1	0	1	0	1	0
3	1	3	1	0	1	3	1	0
4	1	0	1	0	1	0	1	0
5	1	0	1	0	1	0	1	0
6	1	3	1	0	1	3	1	0
7	1	0	1	0	1	0	1	0
8	1	0	1	0	1	0	1	0
9	1	3	1	0	1	3	1	0
10	1	0	1	0	1	0	1	0
11	1	0	1	0	1	0	1	0
12	1	3	1	0	1	3	1	0
13	1	0	1	0	1	0	1	0
14	1	0	1	0	1	0	1	0
15	1	3	1	15	1	3	1	0
16	1	0	1	0	1	0	1	0
17	1	0	1	0	1	0	1	0
18	1	3	1	0	1	3	1	0
19	1	0	1	0	1	0	1	0
20	1	0	1	0	1	0	1	0
21	1	3	1	0	1	3	1	0
22	1	0	1	0	1	0	1	0
23	1	0	1	0	1	0	1	0
24	1	3	1	0	1	3	1	0
25	1	0	1	0	1	0	1	0
26	1	0	1	0	1	0	1	0
27	1	3	1	0	1	3	1	0
28	1	0	1	0	1	0	1	0
29	1	0	1	0	1	0	1	0
30	1	3	1	14	1	3	1	0
31	1	0	1	1	1	0	1	0
32	1	0	1	0	1	0	1	0
33	1	3	1	0	1	3	1	0
34	1	0	1	0	1	0	1	0
35	1	0	1	0	1	0	1	0
36	1	3	1	0	1	3	1	0

37	1	0	1	0	1	0	1	0
38	1	0	0	0	0	0	1	0
39	1	3	2	0	2	3	1	0
40	1	0	1	0	1	0	1	0
41	1	0	1	0	1	0	1	0
42	1	3	1	0	1	3	1	0
43	1	0	1	0	1	0	1	0
44	1	0	1	0	1	0	1	0
45	1	3	0	13	0	3	1	0
46	1	0	2	2	2	0	1	0
47	1	0	1	0	1	0	1	0
48	1	3	1	0	1	3	1	0
49	1	0	1	0	1	0	1	0
50	1	0	1	0	1	0	1	0
51	1	3	1	0	1	3	1	0
52	1	0	0	0	0	0	1	0
53	1	0	2	0	2	0	1	0
54	1	3	1	0	1	3	1	0
55	1	0	1	0	1	0	1	0
56	1	0	1	0	1	0	1	0
57	1	3	1	0	1	3	1	0
58	1	0	1	0	1	0	1	0
59	1	0	0	0	0	0	1	0
60	1	3	2	12	2	3	1	0
61	1	0	1	3	1	0	1	0
62	1	0	1	0	1	0	1	0
63	1	3	1	0	1	3	1	0
64	1	0	1	0	1	0	1	0
65	1	0	1	0	1	0	1	0
66	1	2	0	0	0	2	1	0
67	1	1	2	0	2	1	1	0
68	1	0	1	0	1	0	1	0
69	1	2	0	0	0	2	1	0
70	1	1	2	0	2	1	1	0
71	1	0	1	0	1	0	1	0
72	1	2	1	0	1	2	1	0
73	1	1	0	0	0	1	1	0
74	1	0	2	0	2	0	1	0

75	1	2	1	11	1	2	1	0
76	1	1	0	4	0	1	1	0
77	1	0	2	0	2	0	1	0
78	1	2	1	0	1	2	1	0
79	1	1	1	0	1	1	1	0
80	1	0	0	0	0	0	1	0
81	1	2	2	0	2	2	1	0
82	1	1	1	0	1	1	1	0
83	1	0	0	0	0	0	1	0
84	1	2	2	0	2	2	1	0
85	1	1	1	0	1	1	1	0
86	1	0	1	0	1	0	1	0
87	1	2	0	0	0	2	1	0
88	1	1	2	0	2	1	1	0
89	1	0	1	0	1	0	1	0
90	1	2	0	10	0	2	1	0
91	1	1	2	5	2	1	1	0
92	1	0	1	0	1	0	1	0
93	1	2	1	0	1	2	1	0
94	1	1	0	0	0	1	1	0
95	1	0	2	0	2	0	1	0
96	1	2	1	0	1	2	1	0
97	1	1	0	0	0	1	1	0
98	1	0	2	0	2	0	1	0
99	1	2	1	0	1	2	1	0
100	1	1	0	0	0	1	1	0
101	1	0	1	0	1	0	1	0
102	1	2	2	0	2	2	1	0
103	1	1	1	0	1	1	1	0
104	1	0	0	0	0	0	1	0
105	1	2	2	9	2	2	1	0
106	1	1	1	6	1	1	1	0
107	1	0	0	0	0	0	1	0
108	1	2	1	0	1	2	1	0
109	1	1	2	0	2	1	1	0
110	1	0	1	0	1	0	1	0
111	1	2	0	0	0	2	1	0
112	1	1	2	0	2	1	1	0

113	1	0	1	0	1	0	1	0
114	1	2	0	0	0	2	1	0
115	1	1	1	0	1	1	1	0
116	1	0	2	0	2	0	1	0
117	1	2	1	0	1	2	1	0
118	1	1	0	0	0	1	1	0
119	1	0	2	0	2	0	1	0
120	1	2	1	8	1	2	1	0
121	1	1	0	7	0	1	1	0
122	1	0	1	0	1	0	1	0
123	1	2	2	0	2	2	1	0
124	1	1	1	0	1	1	1	0
125	1	0	0	0	0	0	1	0
126	1	2	2	0	2	2	1	0
127	1	1	1	0	1	1	1	0
128	0	0	0	0	0	0	0	0
129	1	1	1	0	1	1	1	0
130	1	2	2	0	2	2	1	0
131	1	0	0	0	0	0	1	0
132	1	1	1	0	1	1	1	0
133	1	2	2	0	2	2	1	0
134	1	0	1	0	1	0	1	0
135	1	1	0	7	0	1	1	0
136	1	2	1	8	1	2	1	0
137	1	0	2	0	2	0	1	0
138	1	1	0	0	0	1	1	0
139	1	2	1	0	1	2	1	0
140	1	0	2	0	2	0	1	0
141	1	1	1	0	1	1	1	0
142	1	2	0	0	0	2	1	0
143	1	0	1	0	1	0	1	0
144	1	1	2	0	2	1	1	0
145	1	2	0	0	0	2	1	0
146	1	0	1	0	1	0	1	0
147	1	1	2	0	2	1	1	0
148	1	2	1	0	1	2	1	0
149	1	0	0	0	0	0	1	0
150	1	1	1	6	1	1	1	0

151	1	2	2	9	2	2	1	0
152	1	0	0	0	0	0	1	0
153	1	1	1	0	1	1	1	0
154	1	2	2	0	2	2	1	0
155	1	0	1	0	1	0	1	0
156	1	1	0	0	0	1	1	0
157	1	2	1	0	1	2	1	0
158	1	0	2	0	2	0	1	0
159	1	1	0	0	0	1	1	0
160	1	2	1	0	1	2	1	0
161	1	0	2	0	2	0	1	0
162	1	1	0	0	0	1	1	0
163	1	2	1	0	1	2	1	0
164	1	0	1	0	1	0	1	0
165	1	1	2	5	2	1	1	0
166	1	2	0	10	0	2	1	0
167	1	0	1	0	1	0	1	0
168	1	1	2	0	2	1	1	0
169	1	2	0	0	0	2	1	0
170	1	0	1	0	1	0	1	0
171	1	1	1	0	1	1	1	0
172	1	2	2	0	2	2	1	0
173	1	0	0	0	0	0	1	0
174	1	1	1	0	1	1	1	0
175	1	2	2	0	2	2	1	0
176	1	0	0	0	0	0	1	0
177	1	1	1	0	1	1	1	0
178	1	2	1	0	1	2	1	0
179	1	0	2	0	2	0	1	0
180	1	1	0	4	0	1	1	0
181	1	2	1	11	1	2	1	0
182	1	0	2	0	2	0	1	0
183	1	1	0	0	0	1	1	0
184	1	2	1	0	1	2	1	0
185	1	0	1	0	1	0	1	0
186	1	1	2	0	2	1	1	0
187	1	2	0	0	0	2	1	0
188	1	0	1	0	1	0	1	0

189	1	1	2	0	2	1	1	0
190	1	2	0	0	0	2	1	0
191	1	0	1	0	1	0	1	0
192	1	0	1	0	1	0	1	0
193	1	3	1	0	1	3	1	0
194	1	0	1	0	1	0	1	0
195	1	0	1	3	1	0	1	0
196	1	3	2	12	2	3	1	0
197	1	0	0	0	0	0	1	0
198	1	0	1	0	1	0	1	0
199	1	3	1	0	1	3	1	0
200	1	0	1	0	1	0	1	0
201	1	0	1	0	1	0	1	0
202	1	3	1	0	1	3	1	0
203	1	0	2	0	2	0	1	0
204	1	0	0	0	0	0	1	0
205	1	3	1	0	1	3	1	0
206	1	0	1	0	1	0	1	0
207	1	0	1	0	1	0	1	0
208	1	3	1	0	1	3	1	0
209	1	0	1	0	1	0	1	0
210	1	0	2	2	2	0	1	0
211	1	3	0	13	0	3	1	0
212	1	0	1	0	1	0	1	0
213	1	0	1	0	1	0	1	0
214	1	3	1	0	1	3	1	0
215	1	0	1	0	1	0	1	0
216	1	0	1	0	1	0	1	0
217	1	3	2	0	2	3	1	0
218	1	0	0	0	0	0	1	0
219	1	0	1	0	1	0	1	0
220	1	3	1	0	1	3	1	0
221	1	0	1	0	1	0	1	0
222	1	0	1	0	1	0	1	0
223	1	3	1	0	1	3	1	0
224	1	0	1	0	1	0	1	0
225	1	0	1	1	1	0	1	0
226	1	3	1	14	1	3	1	0

227	1	0	1	0	1	0	1	0
228	1	0	1	0	1	0	1	0
229	1	3	1	0	1	3	1	0
230	1	0	1	0	1	0	1	0
231	1	0	1	0	1	0	1	0
232	1	3	1	0	1	3	1	0
233	1	0	1	0	1	0	1	0
234	1	0	1	0	1	0	1	0
235	1	3	1	0	1	3	1	0
236	1	0	1	0	1	0	1	0
237	1	0	1	0	1	0	1	0
238	1	3	1	0	1	3	1	0
239	1	0	1	0	1	0	1	0
240	1	0	1	0	1	0	1	0
241	1	3	1	15	1	3	1	0
242	1	0	1	0	1	0	1	0
243	1	0	1	0	1	0	1	0
244	1	3	1	0	1	3	1	0
245	1	0	1	0	1	0	1	0
246	1	0	1	0	1	0	1	0
247	1	3	1	0	1	3	1	0
248	1	0	1	0	1	0	1	0
249	1	0	1	0	1	0	1	0
250	1	3	1	0	1	3	1	0
251	1	0	1	0	1	0	1	0
252	1	0	1	0	1	0	1	0
253	1	3	1	0	1	3	1	0
254	1	0	1	0	1	0	1	0
255	1	0	1	0	1	0	1	0