密碼學應用實務策略

杜詩怡
高雄師範大學
資訊教育研究所
shihyi@icemail.nknu.edu.tw

楊中皇
高雄師範大學
資訊教育研究所
chyang@computer.org

鍾青萍
長庚大學
資訊管理研究所
p.pete.chong@gmail.com

摘要

在現今資訊科技發達的社會裡，無論是政府單位公文的傳送，或企業之間商業情報的資料通訊，均需仰賴網際網路以傳遞彼此的資料，因此網路安全逐漸成為眾人關切的重要議題。為了有效提升企業組織的競爭能力，並期望企業永續經營的理念下，許多業者注入大量的人力、物力、時間、金錢，以改善自身網路安全漏洞，如：BS7799/ISMS (Information Security Management System) [11] 資訊安全管理系統的導入。然而在追求網路安全的完美境界時，卻已不知不覺投入眾多的資源，造成許多企業的資源無法得到最完善的利用。

有鑑於此，協助企業組織耗費最低的成本，獲得最大的網路安全效益，乃刻不容緩之計。本研究主要收集目前廣為採用的密碼學技術，配合企業採用資訊安全管理技術所需耗費的成本，以推導出合理的 (Pragmatic) 策略，提供企業組織的管理者做出適切決策，讓企業資源得以做最有效的運用。

關鍵詞: 資訊安全管理、密碼學、決策支援、可行性研究 (feasibility)、合理化 (pragmatic)

1. 前言

密碼學最早的使用記錄可以追溯至距今約 4000 多年前，埃及人於墓碑上所雕刻的象形文字與不尋常符號的使用 [12]。換句話說，自人類開始懂得如何書寫時，密碼技術也隨之產生，而密碼技術的演進，可從眾多的歷史事件探查得知 [4]，眾人皆知的凱撒加密法 (Caesar Cipher) 便是應用於國家軍事而產生的加密方法。直到第二次世界大戰之後，由於電腦的出現 (1942 年, UNIVAC) [1] 及網際網路的興起，密碼技術也有了嶄新的風貌。

事實上，早期的加解密技術並不需要透過電腦進行運算，只是電腦發明後，舊有的密碼技術反而可透過電腦與暴力攻擊法 (Brute-Force Attack) 進行快速的破解，相形之下，舊有的密碼技術不再那麼安全。基於安全性的考量，有許多依賴電腦運算的加解密技術因而相繼興起。

現代的加解密技術主要可以分成兩大類，一種是私密金鑰加密系統，另一種則是公開金鑰加密系統。本研究根據這兩種密碼系統，僅列出較廣為使用的加解密演算法。每一個加密演算法，有其各自不同的困難度與成本，而這些困難度與實作成本對駭客而言實屬一大障礙。因此，我們將其實作成本與困難度加以定義後，再參照定義後的數據，列出幾項實用的策略，讓使用者可以根據這些策略，獲取最大的利益。並可在日後供決策管理人員，做為企業資源調整的參考依據。

2. 參考文獻

密碼技術在長達 4000 多年之久的洗禮下，除了使用目的改變之外，其設計理念與架構也大不相同。以往的密碼技術大部分做為軍事之用，但現代的密碼技術還可應用於商業行為及個人資料的保護 [2]；以往的密碼技術都是採用非電子運算的方式進行加解密，例如：密碼棒、密碼盤的使用 [4]，自從電腦的發明與網路的興盛之後，逐漸取代了加解密的運算方式，因此有許多依賴電腦運算的加解密技術逐漸被提出，例如：DES 加密演算法 [13]。

本研究僅列舉出自密碼學的演進以來，較常見的加解密技術，從這些加密技術深入探究破解這些

加解密技術時，可能需要耗費的成本，再根據這些耗費成本的數據，訂定出適合決策者進行企業網路安全控管之相關策略，以利於企業內部資源的調整。換句話說，讓決策人員可以耗費最少的成本，獲得最大的安全效益。

表 1 常見的加解密演算法

私密金鑰加/解密	公開金鑰加/解密
Transportation	Elliptic Curve
One-Time Pad	RSA
Hill Cipher	Diffie-Hellman
Caesar Cipher	ElGamal
Vegenere Cipher	Rabin
Stream	Monoalphabetic Cipher
DES	
3DES	
DESX	
IDEA	
AES	
RC2	
RC4	
RC5	
RC6	
Blowfish	
Skipjack	
CAST (128/256)	

2.1 私密金鑰加密系統

私密金鑰加密系統 (Private-Key Cryptosystems) 最主要的運作概念是指使用同一把金鑰進行加解密的動作，我們可以透過下圖了解私密金鑰加密系統的運作架構。由於加解密時都是採用同一把金鑰，因此金鑰千萬不能讓攻擊者知道，否則資料無法安全送到接收端。

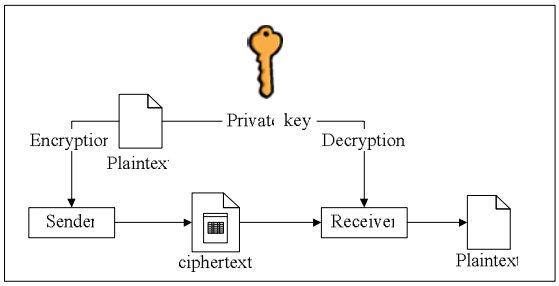

圖 1 私密金鑰加密系統

2.2 公開金鑰加密系統

公開金鑰加密系統 (Public-Key Cryptosystems) 的概念最早於 1975 年由 Diffie 等人所提出，然而第一個實作此構想的人卻是鼎鼎大名的 R. Rivest、A. Shamir and A. Adleman 三個人所提出的 RSA 演算法 [4]。目前 RSA 密碼廣泛應用於瀏覽器 [3] 中，以維護資料傳送時的安全。

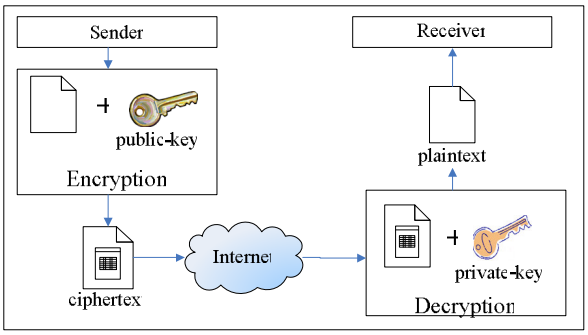

圖 2 公開金鑰加密系統

公開金鑰加密系統和私密金鑰加密系統最大的不同之處，在於加解密時是否使用不同的金鑰。首先傳送端使用接收端的公開金鑰將資料進行加密後，將密文傳送給接收端，而接收端收到密文後，再用自己本身的私密金鑰進行解密的動作，如此一來即可得知明文的內容，但必須特別注意的是接收端的私密金鑰千萬不能外流，否則也會遭受到有心人士的攻擊。

2.3 資訊安全管理系統 (Information Security Management System)

在前文中，我們曾提到密碼學的發展與人類的書信往來有著密不可分的關係，而現今社會的商業行為又必須倚賴網路進行資料的傳遞，因此確保資

料在網路通道中進行安全地傳輸，是極為重要的任務。

在電腦尚未發明之前，密碼技術通常做為軍事用途，若是能破解重要情報的密碼，那麼就有機會搶得先機，改變國家局勢；但隨著電腦的發明，密碼技術已不再僅做為軍事之用，其破解的目的可能是為了商業利益、個人利益，或是僅顯示個人的破解能力罷了。

目前大多數的密碼系統僅提供計算上的安全，換句話說系統其實是有可能被破解。只是攻擊者限於時間的因素，無法進行無止盡的破解 [3]。由此可知，資訊並沒有絕對的安全。因此，我們僅能盡力保護系統，使系統能在保護之下提供正常的服務 [1]。

根據 IDC 的近幾年的調查，有 72%的企業曾遇過網路安全入侵；更有 39%的企業對資訊安全性的威脅感到與日俱增 [5]。另外，在資策會關於網路安全的調查中也顯示，大部分的企業對於網路安全仍感到疑慮 [6]。在去年 IDC 的研究報告中亦指出，到 2005 年為止企業安全性相關支出項目逐漸攀升 [7]。由此可知，有越來越多的企業為了保護企業內部的資料，對於資訊安全的控管越來越予以重視，而資訊安全策略的制定更是深深影響企業的發展與未來。

2.3.1 攻擊的類型

加解密技術攻擊的類型主要可以分成三大類，分別是僅有密文攻擊 (ciphertext-only attack)、已知明文攻擊 (known-plaintext attack)、自選明文攻擊 (chosen-plaintext attack) [13]。本研究也針對此問題，整理出表 1 中所提出的 24 個加解密演算法之破解能力。

表 2 常見的密碼演算法之破解能力

加/解密演算法	破解演算法需具備能力
Stream	(1) Stream Encryption (2) Linear Feedback Shift Register, LFSR (3) Combination Function
Elliptic Curve Cryptosystem	(1) ECC Algorithm (2) ECC Concept (3) Public key Cryptosystem (4) Abstract Algebra (5) Discrete Mathematics (6) Discrete Logarithm Problem (7) Linear Algebra (8) Group, field, isomorphic
Diffie-Hellman	(1) Diffie-Hellman Algorithm (2) Public key Cryptosystem (3) Factorization method (4) Discrete Logarithm Problem (5) Modular Exponentiation
Rabin	(1) Rabin Algorithms (2) Public Key Cryptosystem (3) Prime Factorization
RSA	(1) RSA Algorithm (2) Public key Cryptosystem (3) Factorization method (4) Greatest Common Denominator (5) Method of Successive Division (6) Miller-Rabin (7) Extended Euclidean Algorithm (8) Fermat theorem (9) Discrete Logarithm Problem (10) Euler theorem (11) Modular Exponentiation
ElGamal	(1) ElGamal Algorithm (2) Public key Cryptosystem (3) Random Encryption (4) Factorization method (5) Discrete Logarithm Problem (6) Modular Exponentiation
AES	(1) AES Algorithms (2) Private Key Cryptosystem (3) S-Box (4) Key schedule (5) Replacement (6) Displacement (7) Boolean Algebra
RC5	(1) RC5 Algorithm (2) Private Key Cryptosystem (3) Fast block cipher (4) Key expansion (5) Integer addition (6) Boolean Algebra (7) Variable rotation
RC6	(1) RC6 Algorithms (2) RC5 Algorithms (3) Private Key Cryptosystem (4) Multiplication (5) variable rotation
Skipjack	(1) Skipjack Algorithms (2) Private Key Cryptosystem (3) Modes of Operation

IDEA	(1) IDEA Algorithms (2) Boolean Algebra Mod
Hill Cipher	(1) Private Key Cryptosystem (2) Matrix (3) Multiplication
CAST (128/256)	(1) CAST (128/256) Algorithms (2) Private Key Cryptosystem (3) Boolean Algebra (4) Rotation
Blowfish	(1) Blowfish Algorithms (2) Private Key Cryptosystem (3) Boolean Algebra
DESX	(1) DESX Algorithms (2) DES Algorithms (3) Private Key Cryptosystem (4) Boolean Algebra (5) key expansion (6) key schedule
Transportation	(1) Private Key Cryptosystem (2) Transportation (3) Addition (4) Subtraction
RC2	(1) RC2 Algorithms (2) Private Key Cryptosystem (3) Key expansion (4) S-Box (5) Block encryption algorithm
3DES	(1) 3DES Algorithms (2) Private Key Cryptosystem (3) Boolean Algebra (4) Key expansion (5) Key schedule
DES	(1) DES Algorithms (2) Private Key Cryptosystem (3) Boolean Algebra (4) Key expansion (5) Key schedule
RC4	(1) RC4 Algorithms (2) Private Key Cryptosystem (3) Secret-Key Sort (4) stream cipher designed
One-Time Pad	(1) Private Key Cryptosystem (2) Boolean Algebra (3) Random
Monoalphabetic Cipher	(1) Private Key Cryptosystem (2) Statistics Probability (3) Language Characteristic
Caesar Cipher	(1) Private Key Cryptosystem (2) Addition (3) Subtraction (4) Division
Vegenere Cipher	(1) Private Key Cryptosystem (2) Addition (3) Subtraction

每一種加密演算法，其設計的理論與概念並不相同。欲破解這些加密演算法，除了對該演算法的設計原理有相當程度的理解之外，更需要具備實作該加密演算法的先備知識，如此一來，才能以固有的先備知識為基礎，進行破解的行動。

為了有效防止這些有心人士的緒意攻擊，選取適合的安全機制以保護系統，達到防護功效，這些機制不外乎是保密性 (Confidentiality)、確認性 (Authentication)、資料完整性 (Data Integrity)、不可否認性 (Non-Repudiation)、存取控制 (Access Control)、可用性 (Availability) [3]。除了建立系統安全機制之外，還需建立相關的安全性策略，例如：評估風險並分配資源、定義明確的安全規範、安排使用者教育訓練、執行安全策略與稽核 [1]。

2.3.2 BS7799

BS7799 Code of Practice for Information Security 是由英國標準協會 (British Standards Institute，簡稱 BSI) 於 1999 年所提出定 [8]，其可分成 BS7799 Part1 and BS7799 Part 2 兩大部分。為目前國際上最著名的資訊安全規範，且已被 ISO (International Organization for Standardization) 接受成為國際標準 [9]。BS7799 Part1 又可稱為 ISO17799，其主要內容提供了施行資訊安全控制的措施，意即實施 ISMS 的實作說明；另於 2002 年修訂 BS7799 Part2，其內容主要說明實施 ISMS 與書面的具體要求 [11]，意即取得 BS7799 的驗證與證書方法 [8]。其詳細的規範內容，亦可到網路上下載相關的內容。

2.3.3 ISO 27001

ISO 27001 也是資訊安全管理系統的規範之一。 ISO 27001 標準規範由 ISO 於 2005 年 10 月所公佈，是目前最新的資訊安全管理系統驗證標準。ISO 27001 可以取代 BS7799-2 [14]。ISO 27001 預期未來能提供了一個建立(Establishing)、實作 (Implementing) 、運作 (operating) 、監視 (Monitoring) 、 檢閱 (Reviewing) 、維護 (Maintaining)、改善 (Improving) 資訊安全管理系統的模型 [15][16]。目前我國經濟部標準檢局，亦把 ISO 27001 列為資訊安全管理系統的檢驗標準之一

[10]。

目前市面上有許多業者開設 BS7799 或 ISO 27001 的相關課程，指導企業如何導入 ISMS 以確保企業內、外部的資訊安全。然而一個企業決定在導入該類的系統之前，必須先進行詳細的自我評估以及經過專人引導，再接受相關的教育訓練，才能真正了解該系統對於系統所帶來的影響與益處。導入 ISMS 只是一個開始，其後的風險評估才是最終的重點。這一連串的過程，必定要耗費企業不少的人力、物力、金錢與時間並進行相互配合，其成效也必須待上一段時間才能得知其成效為何。

有鑑於此，本研究所提出的幾個策略是從密碼演算法自身的困難程式，先行篩檢其實作成本占用預算的耗費程度，讓欲導入資訊安全管理系統的管理人員或決策者，能夠用最簡短的時間，進行初步的評估與考量。

3. 研究方法

根據表 2 對於各種演算法破解所需能力的評估，我們給予每一個加解密演算法破解時，所需具備的能力，為該項目設定某一參數，將參數值的結果予以加總，做為破解該密碼技術的困難度。

設計成本的訂定，主要參照演算法的程式設計所需耗費的時間或金錢。也就是說困難度高的演算法，並不代表其實作過程耗費成本或金錢；有些演算法除了軟體上的執行之外，還需配合硬體設備的裝置，才能發揮最大功效，但在此僅考慮軟體實作成本，尚不將硬體設備的配合使用納入考量。

4. 研究結果

根據本研究所提出的加解密演算法，經由實作該加解密演算法，所需具備的破解能力 (參見表 2)，在每一個破解能力的項目裡，均賦予一個破解指標，將這些破解指標加總之後，整理出表 3 的困難度。

表 3 常見的加解密演算法其困難度

加/解密演算法	難易度
Elliptic Curve Cryptosystem	53
RSA	50
ElGamal	37
Diffie-Hellman	33
AES	26
RC5	25
DESX	22
RC6	20
RC2	19
Stream	19
Rabin	18
3DES	17
DES	17
CAST (128/256)	17
RC4	16
Monoalphabetic Cipher	16
Skipjack	14
Blowfish	12
One-Time Pad	11
Transportation	11
Hill Cipher	10
IDEA	10
Caesar Cipher	9
Vegenere Cipher	9

從表 3 可知，Elliptic Curve Cryptosystem 和 RSA 與 EIGamal 和 Diffie-Hellman 兩大組加密演算法相較之下，其困難度的指標高出許多。因此，若我們假設使用 EIGamal 和 Diffie-Hellman 群組的加密演算法，即可濾除大部分來自於駭客所釋出的潛在威脅。越困難的加密演算法，其實作難度高。而這也是目前本研究所訂出的第一個實務策略。

經由查證演算法的實作過程，我們得到以下實作成本的資料 (見表 4) 。很顯然地，成本的增加速度一直都很緩慢，直到 Diffie-Hellman 演算法之後，成本的增加速度就變快了。這樣的情況與傳統的 80/20 規則相符；意思是說，我們可先藉由 20%的支出成本，得到 80%的利益。如為了獲取更多的利益，其成本的增加的速度會急速遞增 [17]。

表 4 常見的加解密演算法其實作成本

加/解密演算法	實作成本
Elliptic Curve Cryptosystem	9
RSA	8
ElGamal	7
Diffie-Hellman	6
AES	6
RC5	6
DESX	5
RC2	5
Stream	5
Rabin	5
3DES	5
DES	5
Skipjack	4
Blowfish	4
RC6	4
CAST (128/256)	3
RC4	3
IDEA	3
Transportation	2
Monoalphabetic Cipher	2
Hill Cipher	2
One-Time Pad	2
Caesar Cipher	2
Vegenere Cipher	2

透過表 4，我們可以清楚地看到，每一個加解密演算法實作成本的指標，這些指標可代表企業所需耗費的成本。在相同的耗費成本下，企業可以自由選擇不同的加解密演算法，以達到企業預防駭客的目標。例如：當企業僅願意花費 3 的成本時，其可選擇 CAST (128/256)、RC4、IDEA 這三種演算法。

因此，我們提出了第二個實務的策略：「查閱成本和利益的清單，利用最少的成本以過濾多數的駭客。」

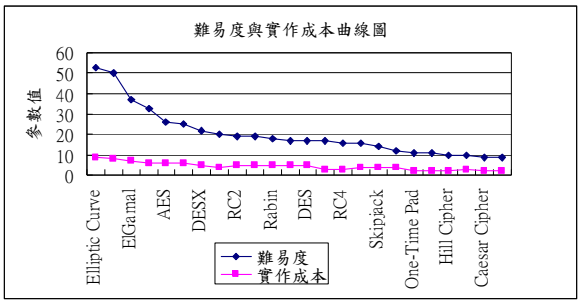

圖 3 難易度與實作成本曲線圖

圖 3 將所有的加解密演算法，依照其困難度與實作成本繪製成圖形，透過圖 3 可清楚地看到所有加解密演算法困難度與成本的差異程度，並且可以清楚地看出每一個加解密演算法，其困難易與實作成本之間的差異。

為了能更清楚的看到每一個加解密演算法其難易度與實作成本之間的關係，我們用實作成本除以難易度，以求出其比率。因為破解的難度實值上是公司的效益，我們改稱之為《成本/效益》比，參見表 5。

表 5 實作《成本/效益》比

加/解密演算法	難易度	實作成本（效益）	比率
Elliptic Curve Cryptosystem	53	9	0.17
RSA	50	8	0.16
ElGamal	37	7	0.19
Diffie-Hellman	33	6	0.18
AES	26	6	0.23
RC5	25	6	0.24
DESX	22	5	0.23
RC6	20	4	0.20
RC2	19	5	0.26
Stream	19	5	0.26
Rabin	18	5	0.28
3DES	17	5	0.29
DES	17	5	0.29
CAST (128/256)	17	3	0.18

RC4	16	3	0.19
Skipjack	14	4	0.29
Blowfish	12	4	0.33
One-Time Pad	11	2	0.18
Transportation	11	2	0.18
Monoalphabetic Cipher	16	4	0.25
Hill Cipher	10	2	0.20
IDEA	10	3	0.30
Caesar Cipher	9	2	0.22

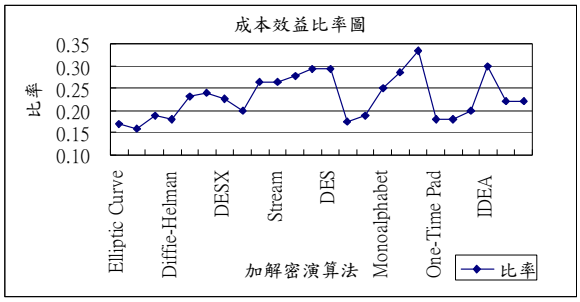

圖 4 《成本效益》比率圖

根據表 5，我們將這些比率視為《成本/效益》比率，並且將每一個加解密演算法的《成本/效益》比率，轉繪成圖 4。在表 5 中，可以明顯得知，若是願意耗費成本 8，可以得到像 RSA 的安全層級（50），且其《成本/效益》比 0.16 是個 Maximization（最低）的解。但是如果資源僅有 6 的話，Diffie-Hellman 會是最優化 (Optimal) 的選擇 [17]。

5. 結論與未來研究

本研究採用實務的策略，徹底分析每一個加解密演算法的破解能力，以定義出駭客進行入侵所遇到的困難程度。因此，我們做了一個合理的假設：「越困難的加解密演算法，越少的駭客可以破解。」我們也更進一步地針對每一個加解密演算法，評估該演算法實作時的成本。透過這些評估出來的實作成本，可協助使用者防範駭客。根據我們的結果，可以找出一種實作時最理想的策略。

然而在這過程中，若駭客企圖破解加解密演算法時，其必須要擁有相當的能力且態度積極。換句話說，當駭客企圖嚐試破解之前，必須有足夠的誘因且耗費一定的時間，才能破解成功。在未來的研究方向，我們預期針對駭客的心理學研究，以定義出大部分駭客最有興趣的領域；若這些領域不吸引極有能力的駭客時，我們就可以採用較低強度的加解密演算法以進行防護，而我們也認為此類的研究是相當有貢獻的。

參考文獻

[1] 陳錦輝，計算機概論，金禾資訊，2004 7 月

[2] 黃心嘉，"密碼之過去、現在與未來"，資通安全分析專論 T94007，2005 年 11 月，p.1-2

[3] 楊中皇，網路安全理論與實務，金禾資訊，2006 年 3 月

[4] Simon Singh，劉燕芬譯, 碼書, 2005, p008、p23-91、p145、p159、pp.309-321

[5] Cisco Systems, "從整合性資訊安全談企業競爭力 ",http://www.cisco.com/global/TW/about/news/news_20030804_2.shtml, 1992-2006 年

[6] 資策會，"企業對網路安全感到疑慮" http://sna.csie.ndhu.edu.tw/~cnyang/CO/sld009.htm 。

[7] IDC Taiwan, "2005 年止企業安全性相關支出日漸攀升", http://www.idc.com.tw/report/Column/ column_050314.htm# 。

[8] 劉智勇,"淺談 BS7799 及其導入步驟" http://www.ringline.com.tw/epaper/BS7799.htm, 2004 年 3 月。

[9] 中華龍網股份有限公司, "系統安全概念" http://www.dragonsoft.com/doc/bs7799-intro.php

[10] 經濟部標準檢驗局，"指定資訊安全管理系統驗證之標準為 ISO/IEC 27001" 。 http://www.bsmi.gov.tw/page/pagetype8_sub.jsp?no=38&pageno=886&type_no=1&groupid=5

[11] ISO 1799 CENTRAL, "*Introducing ISO 17799*" http://www.17799central.com/

[12] Cairns, Australia, "*History of Cryptography*", http://www.cypher.com.au/crypto_history.htm, January 2006

[13] A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, October 1996 http://www.cisco.com/global/TW/about/news/news_20030804_2.shtml 。

[14] ISO 27001, "*ISO 27001 Security*," http://www.27001-online.com/

[15] IsecT Ltd, "*ISO/IEC 27001:2005, the latest international standard Specification for an Information Security Management System*" http://www.iso27001security.com/html/iso27001.html, 2006.

[16] ISO 27000 and ISO 27001, "*Introducing ISO 27001*," http://www.17799central.com/iso-27001.htm, 2005.

[17] Chong, P.P., T.T. Chuang, M. Chang, and J.C.H. Chen, "Optimal Purchase Decision Criteria for Information Technology," a chapter in *Managing IT in Government, Business & Communities*, IGP/INFOSCI/IRM, 2003.