

# 國立交通大學

資訊科學系

碩士論文

一個具層級性的動態群體簽章系統

A Hierarchical Dynamic Group Signature Scheme

研究生：張兆儀

指導教授：曾文貴 教授

中華民國九十三年六月

一個具層級性的動態群體簽章系統  
A Hierarchical Dynamic Group Signature Scheme

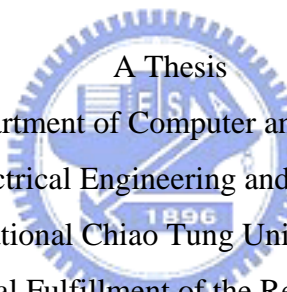
研究生：張兆儀

Student : Chao-Yi Chang

指導教授：曾文貴

Advisor : Wen-Guey Tzeng

國立交通大學  
資訊科學系  
碩士論文



A Thesis  
Submitted to Department of Computer and Information Science  
College of Electrical Engineering and Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in

Computer and Information Science

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

# 一個具層級性的動態群體簽章系統

學生：張兆儀

指導教授：曾文貴 博士

國立交通大學資訊科學系

## 摘要

群體簽章的概念，是指在一個群體中，任何一個成員皆可匿名地代表此群體作簽章的動作。當遇到有爭議之時，才由系統指定的群體管理者來解開(open)簽章，公佈簽章者的身份。對群體管理者之外的人而言，此種簽章除了滿足原本簽章該有的安全性外，還須達到匿名性的要求。

本篇論文是基於[15]的群體簽章架構，加上層級性的構想：即處在較高層級的群體管理者，除了可打開該群體的群體簽章外，還可打開此群體之下所轄群體的群體簽章。另外，也加入了群體成員加入時可給定合法時限的作法，以減輕[15]中憑證廢止列表的負擔。

關鍵詞：群體簽章、層級性、時限

# A Hierarchical Dynamic Group Signature Scheme

Student : Chao-Yi Chang

Advisor : Dr. Wen-Guey Tzeng

Department of Computer and Information Science  
National Chiao-Tung University

## Abstract

The concept of “group signature” means that, in a group, any member can sign on behalf of the group anonymously. While there is a dispute, a group manager assigned by the system will “open” the signature and make the signer public. For any one except the group manager, this kind of signatures satisfies not only the security properties but also “anonymity”.

This thesis is based on the construction of group signature in [15], and adds the idea of “Hierarchy”. A manager of a higher group, can open not only the group signatures of this group, but also the group signatures of lower affiliated groups. Additionally, we also add the method of assigning valid time-bound when a user joins the group, in order to reduce the loads of the certificate revocation lists in [15].

**Keywords** : Group signature, Hierarchy, Time-bounded

# 誌謝

在此感謝我的指導老師曾文貴教授，在我碩士班兩年的學習過程中，不只讓我在學業上受益良多，更在生活上以及言行上給我許多教導。此外，我要感謝口試委員，交大資工系蔡錫鈞教授和清大資工系孫宏民教授，在論文上給予我許多良好的建議和指導，讓我的論文更加完善。除此之外我要感謝實驗室同學，尚宸、坤杉、振魁和佩琳的幫忙，實驗室學長成康、惠龍、學姊季穎的指導，以及最可愛的女朋友和實驗室學弟妹們在精神方面的鼓勵。

最後，我要感謝我的家人，不論在精神或物質上都給予我極大的支持，讓我在無後顧之憂的情況下可以順利完成學業。在此，謹以此文獻給我所有我想要感謝的人。



# 目錄

摘要 .....	i
Abstract .....	ii
誌謝 .....	iii
目錄 .....	iv
第一章 引言 .....	1
第一節 研究動機 .....	1
第二節 研究重點 .....	3
第三節 各章節簡介 .....	3
第二章 基本技術與原理 .....	4
第一節 相關研究 .....	4
第二節 群體簽章的系統模型與特性 .....	6
第三節 相關的定義與假設 .....	10
第四節 零知識互動式證明系統及知識簽章 .....	11
第三章 一個具層級性的動態群體簽章系統 .....	18
第一節 參數定義 .....	18
第二節 一個具層級性的動態群體簽章系統 .....	19
第四章 安全性證明 .....	30
第一節 基礎架構的安全性 .....	30

第二節 其他安全性證明 .....	41
第五章 結論 .....	45
第一節 解章者 .....	45
第二節 結論與未來研究方向 .....	47
參考文獻 .....	49



# 第一章 引言

在現今的科技時代中，數位化資訊的傳送非常地多元且普遍，尤其是在網際網路普及的情形下，傳送資料除了是否能傳達之外，也衍生出許多不同的需要。無可避免地，並非所有資訊皆可不加處理而直接讓人擷取並利用，因此許多相關的保護機制便被人們提出來：例如加密 (Encryption)、數位簽章 (Digital signature)、身份認證 (Authentication)……等等。

數位簽章的基本觀念，便是要提供某方與自己所發文件之間的關係與責任。例如對自己所發出之文件不可否認，或讓其他人相信此文件為某方所簽且未被竄改的正確文件。其基本精神為簽章者在簽章中藏入一些與文件和個人有關的秘密資訊，而驗證者可透過一些與上述秘密資訊有關的公開資訊來驗證。實現的方法需架構在公開金鑰密碼系統 (Public key cryptography) 中，簽名者 (Signer) 需有自己的公開金鑰 (Public key) 和私密金鑰 (Secret key)，簽名時利用後者透過簽章演算法 (Signing algorithm) 來對文件造出簽章 (Signature)，前者需公開以供驗證者 (Verifier) 索取並透過驗證演算法 (Verifying algorithm) 驗證之用。

群體簽章 (Group signature) 為數位簽章的一種運用，目的即是在於讓一個群體中的任一成員，可代替群體作簽章動作而不洩漏身份，當有爭執或疑義時，才由群體中的管理者出面從簽章中擷取出使用者身份。

## 第一節 研究動機

群體簽章的應用狀況，我們可以想像如下情況：有一金融機構可發行電子錢，方式為若有一份紀錄金額的文件，則此機構中的部門可對此文件進行簽章，經簽章後即成為合法之電子錢。在外流通時，收費者可透過公開資訊驗證此電子錢為該金融機構發行且為合法之電子錢，但並不需知道此電子錢



由該金融機構何部門所發出。只有當發生弊案或不合理狀況時，才需由此金融機構之管理者打開電子錢內部資訊，查明此金額是由何人發出。

還有一種情況：某公司欲對其發行之產品加上版權宣告，於是各部門所賣出的產品都會先經過簽章步驟。這樣發行的產品，可讓一般人輕易驗證此產品為此公司所製造，而不會知道是由何部門售出或使用者由何行銷管道得來，當有需要調查時，才由管理者對產品之簽章進行擷取身份的動作。

如上的情形皆可透過群體簽章的機制來實現。因為群體簽章的目的即是讓群體成員可代替群體作簽章且不洩漏身份，簽出的章具有其安全性。遇有爭議時，系統之管理者可公開簽章之簽章者以解決爭議。

目前群體簽章可適用的對象，除了以上基本的功能外，還可以滿足成員剔除的功能。也就是說可以讓某一個成員喪失代替群體簽章的能力。較方便的作法是加入時間與廢止憑證列表的機制：在每個時期，成員簽章金鑰中的憑證都不同，當成員被剔除時，系統會將此成員自現在以後的憑證公開，以讓驗證者能對照廢止憑證列表，確定此簽章非由遭剔除的成員所簽出。

我們想要滿足的是以下額外的狀況，

- (1) 在之前群體簽章應用的兩個狀況中，可以讓公司或銀行有等級高低之分，即等級較高的銀行或公司可以在特殊的狀況下，解開其子公司或子銀行的簽章，公開簽章者身份。而較高等級的機構也可以是政府的行政單位或治安單位，以供法律用途。
- (2) 在一般的廢止機制中，驗證過程常需對照廢止憑證列表一一檢驗，若能增加成員加入群體即指定其合法時限的功能，如此盡可能減少廢止憑證列表的資料量，與驗證時的運算量，系統效率也會有所提升。

## 第二節 研究重點

本篇論文的研究方向主要有兩點，第一點是結合階層 (Hierarchy)[1] 和群體簽章系統[15]的方法，發展出一個可具有階層式架構的群體簽章系統；第二點是在簽章系統中加入使用者合法時限的概念，以分擔廢止憑證列表所需儲存的資料量及驗證時對照廢止憑證列表的計算量。

成果方面，我們如同上述將[15]中提出的群體簽章系統以[1]的階層架構技巧連結在一起，於是較高階層的群體管理者可以在需要時解開較低階層群體的簽章，此部分功能甚至可視需要，加以授權給某個專職解開簽章的角色。而在使用者加入群體即給定合法時限的作法，也證明是安全的。整個架構也仍然滿足[15]中群體簽章所有的安全性。

## 第三節 各章節簡介



本文各章節配置如下：第二章介紹基本知識，包括群體簽章架構、各項安全性、證明安全性時需用到的數論假設、與群體簽章常用到的元件-知識簽章。第三章正式介紹我們的整個系統。第四章為安全性證明，包括原本底層的群體簽章架構安全性與改為層級性架構後需考慮的安全狀況。第五章則介紹一個可獨立的功能-解章 (open)，與未來研究方向及結論。

## 第二章 基本技術與原理

群體簽章還是不脫簽章的基本需求，也就是有人簽章，有人驗證。只是改成一個團體中的成員都能簽出代表此團體的章。由這個觀念出發，衍生出許多安全性的需求和改良，也可能與其他簽章系統的某些觀念相結合而發展出新的群體簽章。本章將首先將介紹群體簽章的相關研究發展、模型與安全性。再介紹有關我們系統安全所需要用到的數論假設，最後介紹近年來群體簽章最常需要用到的元件—知識簽章。

### 第一節 相關研究

群體簽章的概念最早是由 D. Chaum 和 E. van Heyst 在 1991 年首先提出來的[12]。架構中包含了群體成員、成員管理者 (Membership manager) 和撤銷管理者 (Revocation manager)。群體的成員被允許能匿名地代替群體來作簽章，簽章可由群體的公開金鑰驗證而不洩漏簽章者的身份。成員管理者負責系統設定與成員加入；撤銷管理者的能力是撤除簽章的匿名性 (Anonymity)，也就是解開簽章求出簽章者的身份。其中兩管理者通常為同一單位，簡稱為群體管理者 (Group manager)。此架構中最基本的安全要求就是簽章的成員身份不會由所簽出的章洩漏(匿名性)，除非能夠擁有一把用來解開簽章的私密金鑰(可追蹤性 Traceability)。隨著時間過去，有越來越多的安全性被考慮。例如：不可連結性 (Unlinkability)、不可偽造性 (Unforgeability)、聯合抵抗性 (Collusion Resistance)[3]、脫罪性 (Exculpability)[3]、抵抗誣陷性 (Framing resistance)[13]。

L. Chen 和 T. P. Petersen 將 [12] 作了改進 [13]，裡面使用了 Schoenmaker 的協定來隱藏簽章者的身份，也允許新成員動態 (dynamically)

加入，並提出了將群體簽章運用在電子拍賣的構想。後來 H. Petersen 提出了一個的方法能將一般數位簽章轉換成群體簽章的架構[21]，所使用的方法是結合 Stadler 的離散對數可驗證加密法和 Schoenmaker 的協定，證明知道離散對數且不洩漏其值。在[6][12][13][21]這些研究中，群體的公開金鑰和簽章的長度，都會和群體的大小，也就是成員個數，呈線性 (linear) 成長的關係，因此便很不適合用於成員個數較多的大群體。到了 1997 年，J. Camenisch 和 M. Stadler 聯合提出了第一篇公開金鑰和簽章的長度可以不需和成員個數成線性關係的群體簽章架構[10]，所用到的主要技巧就是知識簽章 (Signature of knowledge)。這樣的構想對後來群體簽章的發展影響很大，有很多的研究都是以知識簽章的基礎加以改進。Camenisch 和 M. Michels 在強 RSA 假設[4][14]的基礎上，提出了一個有效率的架構[7][8]，隨後 H. J. Kim 等人改進他們的架構並提供了有效率的成員撤銷機制[17]。G. Ateniese 和 G. Tsudik 在[3]中討論了群體簽章在現實生活中的應用可能需要考慮的問題，例如聯合攻擊 (Coalition attacks) 和成員刪除 (Member deletion)。Ateniese 等人後來發表了一個可證安全 (provably secure) 的群體簽章架構[2]。D. X. Song 在[23]中提出了兩個問題：(1)考慮成員簽章金鑰洩漏和(2)如何有效率地撤銷群體成員。事實上，一個成員的群體的簽章金鑰中應該包括了成員的秘密值 (Membership secret) 和成員憑證 (Membership secret)[2][10]。於是 Song 根據前進式安全 (Forward secure)[5][16]的觀念，發展出第一個前進式安全的群體簽章架構。在他的架構中，將系統的生存時間 (Life-time) 整個分成  $T$  個時段，在時段  $j+1$  開始時，群體成員會利用一個公開的單向函數 (One-way function) 將自己前一時期的簽章金鑰  $sk_j$  更新為此時期的簽章金鑰  $sk_{j+1}$ ，再將  $sk_j$  之值由系統中刪去，這樣一來，若時段  $j+1$  時系統被攻破了，攻擊者也無法由  $sk_{j+1}$  往回推出時段  $j+1$  之前的簽章金鑰，達到了前進式安全的特性。Song

也將自己的架構延伸出支援成員刪除的功能，在他的群體簽章架構裡，群體的公開金鑰並不會受成員加入、撤銷，或是更新金鑰所影響。2003 年的 J. Zhang、Q. Wu、Y. Wang 等人曾提出一個有效率的新群體簽章架構[26]，因為並非使用知識簽章，因此在計算和通訊上比起一般的架構都更有效率，同時並宣稱滿足所有需要的安全性。但在[24]中卻證明了 Zhang-Wu-Wang 的群體簽章架構是可連結 (linkable)、不可追蹤 (untraceable) 且可偽造 (forgeable) 的。

在[18]中曾經介紹了一個群體簽章的系統，各群體之間有層級性的關係。其層級性主要是著重在身處比較高階群體的成員，可以代替比較低階的群體作簽章。這篇論文中的系統是以許多數論中的性質來實作的，因此成員簽出的章與群體的公開金鑰皆會與成員所加入群體中的成員數呈線性關係。此篇論文的研究方向也提過可以嘗試在此系統中用知識簽章來改進這些缺點。而我們的論文則是直接用了知識簽章的作法，因此不會有[18]中所提的缺點，但我們的方向是著重在群體管理者解開簽章的部分，所以並沒有討論到[18]中高階群體成員可代替低階群體簽章的功能。

群體簽章的特性之一-無連結性 (Unlinkability) 可應用在某些需要隱藏身份的情況，例如電子投票 (Electronic voting)、電子拍賣 (Electronic Auction)等。也可被用在電子錢 (Electronic cash) 系統中來隱藏發錢的機構[9][11][19][20][22]。

## 第二節 群體簽章的系統模型與特性

組成群體簽章系統的基本元件一共有五個[2]。除此之外，[15]中再加入了金鑰更新程序 (Key update procedure) [16]與廢止程序 (Revoke procedure)。



本文所提的群體簽章系統還加入了層級初始程序 (Hierarchy initialization)，所以我們將完整架構定義如下。

**定義1.** 一個有層級性的動態群體簽章系統可分為下列八大部分：

■ 層級初始：決定所有群體相互間的層級關係，並設定各群體的部分私密金鑰。程序結束後，各群體  $G_i$  皆會得到一把不同的私密金鑰  $K_i$ 。

■ 群體設定：輸入一個安全參數  $l$ ，一個金鑰值  $K$ ，此機率程序 (Probabilistic procedure) 會輸出系統參數 (System parameter)、群體公開金鑰和群體管理者的私密金鑰。

■ 參與程序：此程序由群體管理者和使用者進行。執行結果是會讓使用者成為群體的合法成員，並得到使用者最初合法時段的金鑰更新元件。

■ 更新程序：輸入時段  $k$  的金鑰更新元件，此程序會輸出時段  $k$  的簽名金鑰與時段  $k+1$  的金鑰更新元件，並刪除時段  $k-1$  的群體簽章金鑰和時段  $k$  的金鑰更新元件。

■ 簽章程序：輸入所屬群體  $i$ 、群體簽章金鑰、訊息  $m$ ，和時間  $k$ ，簽章程序會輸出群體  $i$  相對於時間  $k$  對此訊息  $m$  上的群體簽章。

■ 驗證程序：輸入群體  $i$  相對於時間  $k$  在訊息  $m$  上的群體簽章和群體  $i$  的公開金鑰，此驗證程序可驗證出此簽章是否為群體  $i$  相對於時間  $k$  在訊息  $m$  上的群體簽章。在我們的驗證步驟中還必須檢查憑證撤銷列表 (CRL, Certificate revocation lists)，以確認此簽章是否由已被廢止的成員身份憑證簽出來的，若是由被廢止的成員身份憑證簽出來的，則不能通過此驗證程序。

■ 廢止程序：在時段  $k$  時，管理者對欲廢止的成員身份憑證，利用金

鑰更新程序算出此成員在時段  $k+1$ 、 $k+2$ ..... $d$  時的身份憑證並公佈在 CRL，其中  $d$  為此遭廢止的使用者在參與程序時，群體管理者所指定時限的最末合法時段，若無設定則  $d =$  系統生存時間  $\tau$ 。再發出訊息指示系統剩餘成員進入下一時段，剩餘成員也在進入下一時段時更新自己持有的簽名金鑰和金鑰更新元件。

■ 解章程序：給定一個群體簽章、相對於此簽章的訊息、此簽章所屬群體的公開金鑰、解章者的私密金鑰。若此解章者具有合法之解章權限，則此程序可輸出簽名者的身份。

群體簽章的安全性，都是由它所需要符合的特性，加上實際狀況發展而來的。一般的群體簽章需要滿足下列七項特性[2][15]，包括正確性 (Correctness)、不可偽造性 (Unforgeability)、匿名性 (Anonymity)、無連結性 (Unlinkability)、脫罪性 (Exculpability)、可追蹤性 (Traceability)、聯合抵抗性 (Coalition-Resistance)。另外還有在[15][23]中所定義的有追溯效力的公開廢止性 (Retroactive public revocability) 和向後式無連結性 (Backward unlinkability)。接下來我們定義一個安全的動態群體簽章系統。

**定義2.** 一個安全的有時限性的群體簽章系統，必須符合下列九個特性：

■ 正確性：一個由合法的群體成員經過簽章程序所造出來的簽章，必須能通過驗證程序。

■ 不可偽造性：只有合法的群體成員能夠以群體的身份代表群體簽章。

■ 匿名性：給定針對某訊息的合法簽章，除了群體管理者外，任何人要在多項式時間內算出此簽章真正的簽名者身份是很困難的。

■ 無連結性：除了群體管理者外，任何人要在多項式時間內判別兩份

正確的簽章是否由相同的簽章者所簽出是非常困難的。

■ 脫罪性：任何群體中的合法成員，包括群體管理者，都不能以其他合法成員的身份去作簽名。有一相近的性質為不可誣陷性 (Non-framing)，其意義為一個群體成員不會被迫對自己未簽過的簽章負責，即一個非本身所簽的章不會被解章程序判定為自己所簽。

■ 聯合抵抗性：可分為兩種，弱的聯合抵抗性為在群體中，若一群有惡意的人，其中成員所持有最早的金鑰更新元件為在時段  $k_{\min}$  的金鑰更新元件，則他們無法造出一個時間早於  $k_{\min}$  的合法簽章，使得群體管理者無法透過解章程序追蹤到這群人中任何一人。因為在本篇論文中，加上了使用者有合法期限的設定，所以在弱的聯合抵抗性方面，除了滿足以上的敘述外，還希望這些惡意成員也無法造出一個時間晚於  $d_{\max}$  的合法簽章能滿足上述的條件，其中  $d_{\max}$  為這些惡意成員所被指定的合法時限中最大者。強的聯合抵抗性為在群體中，任何一群有惡意的人，皆無法合作簽出一份任何時間的合法群體簽章，使得群體管理者無法透過解章程序追蹤到這群人中任何一人。

■ 可追蹤性：可根據上方聯合抵抗性的定義分為兩種：弱的可追蹤性為在上方定義弱的聯合抵抗性之下，群體管理者一定可以解開一個合法的群體簽章，得到真正簽章者的身份。強的可追蹤性為在上方定義強的聯合抵抗性之下，群體管理者一定可以解開一個合法的群體簽章，得到真正簽章者的身份。

■ 有追溯效力的公開廢止性：在時段  $k$  時，群體管理者可以將成員在時段  $l$  時，成員用來簽章的簽章金鑰中的成員憑證公佈，使得任何在時段  $l$  之後(包括  $l$ )，此成員用廢止後的成員憑證簽出來的章都是不合法的。



■ 向後式無連結性：若廢止時間在  $l$ ，則在時段  $l$  之前的簽章仍然保持匿名性和無連結性。

### 第三節 相關的定義與假設

強 RSA 假設是一個比 RSA 更強的假設，RSA 假設是說在一個模數 (Modulus)  $n$  之下，要找某數的  $e$  次方根是很難的，其中  $e \in \mathbb{Z}_{>1}$  是一個固定的公開值；而強 RSA 假設則是說要找出任意  $e$  次方根都是困難的。此假設最早是由[4][14]提出來的。

**定義3.** 強 RSA 問題( Strong-RSA problem )：令  $n = pq$  其中  $n$  為一般 RSA 問題中的模數  $n$ ，即  $p$ 、 $q$  為強質數，且令  $G$  為  $\mathbb{Z}_n^*$  的一個含  $\#G$  個元素的子循環群，令  $l_G = \lceil \log_2(\#G) \rceil$ 。給定  $n$  和某一個  $z \in G$ ，強 RSA 問題就是要找到一組  $(u, e)$  其中  $u \in G$  且  $e \in \mathbb{Z}_{>1}$  滿足  $z \equiv u^e \pmod{n}$ 。

**假設1.** 強 RSA 假設(Strong-RSA Assumption)：令  $l_G$  為安全參數。任意一個滿足定義 3.要求的  $n$ ，對任何機率的多項式時間演算法  $A$ ， $A$  能解強 RSA 問題的機率皆可忽略。正規一點的寫法如下：對所有機率的多項式時間演算法  $A$ ，所有的  $c > 0$ ，和所有足夠大的安全參數  $l_G$ ，都會滿足下方這樣的式子

$$\Pr \left[ z \equiv u^e \pmod{n} \wedge \gcd(e, \phi(n)) = 1 \wedge 1 < e < n : z \in_R G; (u, e) \leftarrow A(n, z) \right] \leq l_G^{-c}$$

## 第四節 零知識互動式證明系統及知識簽章

所謂的零知識互動證明系統 (Zero-knowledge interactive prove system) 是一種由兩方溝通的互動式證明系統 (Interactive proof system)：一方為證明者 (Prover)，另一方為驗證者 (Verifier)；系統中有一些公開值，而系統的活動就是證明者需向驗證者證明其知道有關公開值的一些秘密值。零知識則是指在這個證明過程中，證明者除了證明了它知道這些秘密值之外，不會透露任何其他其它資訊。

互動式的證明系統可分為兩類，一類是語言成員的證明 (Proof of membership of language)，用來讓證明者向驗證者證明某些資訊是否屬於某個特定的語言；另一類為判斷式的證明 (Proof of knowledge of predicates)，主要是證明者用來向驗證者證明他知道某些秘密的實際值能滿足判斷式。而我們在這裡會用到的是後者，我們引用[15]的定義來介紹。使用到的名詞定義如下：



PPTM 為機率的多項式時間杜林機器 (Probabilistic polynomial-time Turing machine)。

$Q$  為某個二元的判斷式(predicate)，且可輸入  $x$ ，若  $p$  對  $x$  而言合於此判斷式，則  $Q(x, p)=1$ ，否則  $Q(x, p)=0$ 。而任何合於正確形式的  $x$  都會存在秘密值  $\rho$  使得  $Q(x, \rho)=1$ 。

定義  $\langle P(\rho), V \rangle(x)$  為  $P$  和  $V$  的一個互動式證明系統，使得  $x$  是此系統中公開的輸入值，而  $\rho$  是  $P$  的私有輸入值。

**定義4.** 令證明者  $P$  和驗證者  $V$  都是 PPTM，則對於  $Q$  的一個零知識互動式證明系統需滿足以下三個條件：

1. 完整性 (Completeness) : 對任何滿足  $Q(x, \rho) = 1$  的  $x$  和  $\rho$ ,

$$\Pr[\langle P(\rho), V \rangle(x) = 1] = 1 \text{ 恆成立。}$$

2. 完美性 (Soundness) : 存在一個機率的多項式時間的知識擷取者 (knowledge extractor)  $E$ , 使得任何  $x$  屬於  $Dom(Q)$  ( $Q$  的定義域, 即  $x$  合於正確形式) 與任何證明者  $P^*$ , 若存在著以下的關係

$$\Pr[\langle P^*(\rho), V \rangle(x) = 1] \geq \frac{1}{p(|x|)},$$

我們就完全可以推得這樣的結果  $\Pr[E(P^*, V, x) = \rho^*, Q(x, \rho^*) = 1] = 1 - \varepsilon(|x|)$ , 其中  $p(\cdot)$  為一個多項式, 而  $\varepsilon(\cdot)$  為一個可忽略的值。

3. 零知識 (Zero-knowledge) : 對每個驗證者  $V^*$ , 都會存在一個模擬器 (Simulator)  $M_{V^*}$ , 使得如下兩個分佈為多項式時間之不可分辨

(Polynomially indistinguishable) :

$$\left\{ \langle P(\rho), V^* \rangle(x) \right\}_{x \in Dom(Q), Q(x, \rho) = 1}$$

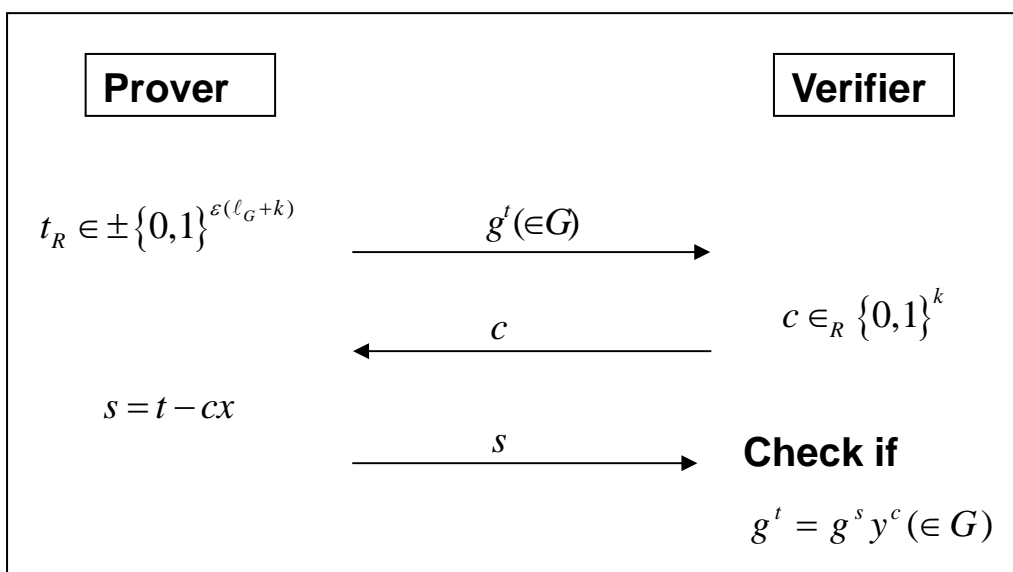
$$\left\{ M_{V^*}(x) \right\}_{x \in Dom(Q)}$$

在完整性中, 主要是證明若  $x$  和  $\rho$  符合  $Q(x, \rho) = 1$ , 那麼擁有私有輸入值的證明者  $P$  一定能說服 (Convince) 驗證者  $V$ , 使  $V$  相信  $P$  知道  $\rho$  這個秘密值。在完美性中, 主要是說若有某個證明者  $P^*$  有很大(不可忽略)的機率可說服驗證者  $V$  的話, 那麼會存在一個知識擷取者  $E$ , 可利用這個  $P^*$  來與此  $V$  互動, 將此  $P^*$  當子程式呼叫使用, 產生出相對應於  $x$ , 而能滿足  $Q(x, \rho^*) = 1$  的秘密值  $\rho^*$ 。在零知識部分, 是希望對任何驗證者  $V^*$  而言, 都能存在一個模擬器  $M_{V^*}$ 。當  $V^*$  與一其對應公開值為  $x$  的證明者  $P$  作互動式證明時, 所能得到的資訊若與  $M_{V^*}$  只得到公開值  $x$  而

能造出來的資訊分佈為不可分辨的話，則代表  $V^*$  與  $P$  的互動並不會造成  $P$  有多餘資訊的外洩，此即為零知識的含意。

而這裡我們所需要用到的元件-知識簽章 (Signature of knowledge)[10]就是由以上的零知識互動式證明系統轉化而來，需要加上一個雜湊函數 (Hash function) 來扮演一個誠實的驗證者 (Honest-verifier) 提供挑戰值 (Challenge)，讓整個證明由互動式成為可由證明者單方提供的非互動式證明。我們介紹這裡所需要的四種知識簽章，包括(i)離散對數(ii)兩離散對數的相等(iii)落在一定區間的離散對數(iv)表示方式。這些都是建構在一個循環群  $G = \langle g \rangle$  之上，其元素個數  $\#G$  是未知的，然而其長度  $\ell_G$  (即一整數  $\ell_G$  使得  $2^{\ell_G-1} \leq \#G \leq 2^{\ell_G}$ ) 是公開的。參考[15]，我們定義  $G$  中一元素  $y$  的離散對數為任何一個整數  $x$  能使得在  $G$  中  $y = g^x$ ，記為  $x = \log_g y$ 。另有一抗碰撞(collision resistant)的雜湊函數  $H: \{0,1\}^* \rightarrow \{0,1\}^k$  能將任何長度的二文字串映射到一個長度為  $k$  的雜湊值。與定義一安全參數  $\epsilon > 1$ 。

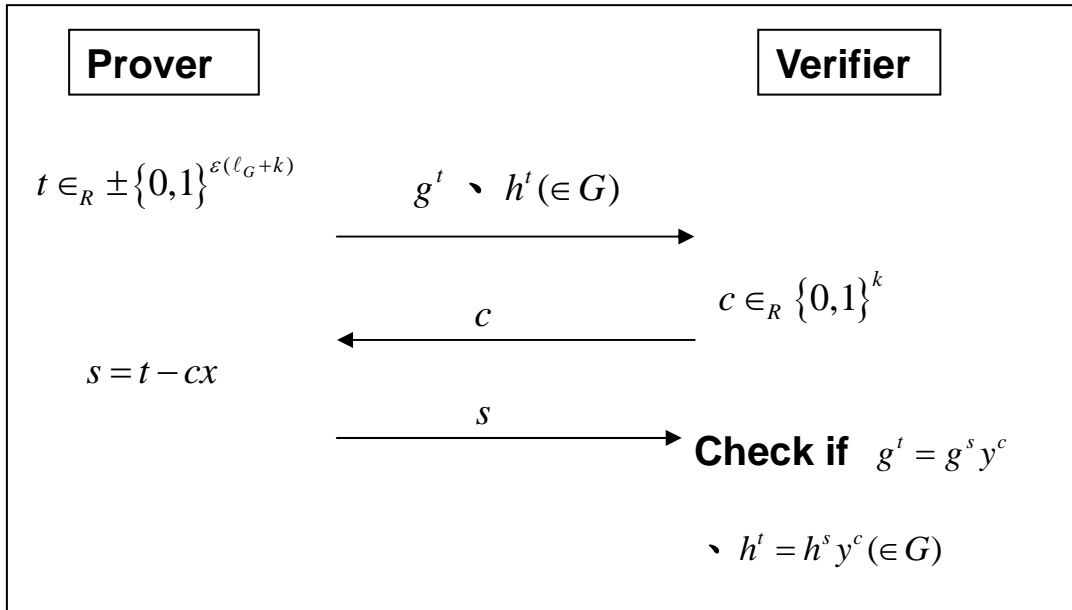
**定義5. 離散對數值的知識簽章 (Signature of knowledge of discrete logarithm)[10]**：令  $y$ 、 $g$  為  $G$  中的元素。一個數對  $(c, s) \in \{0,1\}^k \times \pm\{0,1\}^{\epsilon(\ell_G+k)+1}$  是一個證明者知道  $y = g^x$  對基底  $g$  的離散對數值，對一訊息  $m \in \{0,1\}^*$  造出的知識簽章。驗證者可由是否  $c = H(y \| g \| g^s y^c \| m)$  來驗證。此簽章造法為由一知道  $x$  者，任選一隨機的  $t \in \pm\{0,1\}^{\epsilon(\ell_G+k)}$  再算出  $c = H(y \| g \| g^t \| m)$  與  $s = t - cx$ 。下圖即為此證明系統的原始的互動式版本，在轉為知識簽章時，我們用了一個抗碰撞的雜湊函數來作為一個誠實的驗證者。



圖表 1. 離散對數值的知識簽章

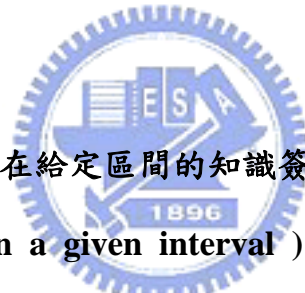
由定義 5 稍加變化可造出如下兩離散對數相等的知識簽章，我們令  $x$  同時為  $y_1$  對基底  $g$  與  $y_2$  對基底  $h$  的離散對數。

**定義 6. 兩離散對數相等性的知識簽章 (Signature of knowledge of equality of two discrete logarithms)[10]**：令  $y_1$ 、 $y_2$ 、 $g$ 、 $h$  為  $G$  中的元素。一數對  $(c, s) \in \{0,1\}^k \times \pm\{0,1\}^{\varepsilon(\ell_G+k)+1}$  是一個證明者知道  $y_1$  對基底  $g$  與  $y_2$  對基底  $h$  的離散對數且兩者相等，對一訊息  $m \in \{0,1\}^*$  造出的知識簽章。驗證者可由是否  $c = H(y_1 \| y_2 \| g \| h \| g^s y_1^c \| h^s y_2^c \| m)$  來驗證。此簽章造法為由一知道  $x$  者，任選一隨機的  $t \in \pm\{0,1\}^{\varepsilon(\ell_G+k)}$  再算出  $c = H(y_1 \| y_2 \| g \| h \| g^t \| h^t \| m)$  與  $s = t - cx$ 。下圖即為此證明系統的原始的互動式版本。同樣用一個抗碰撞的雜湊函數來作為誠實的驗證者。

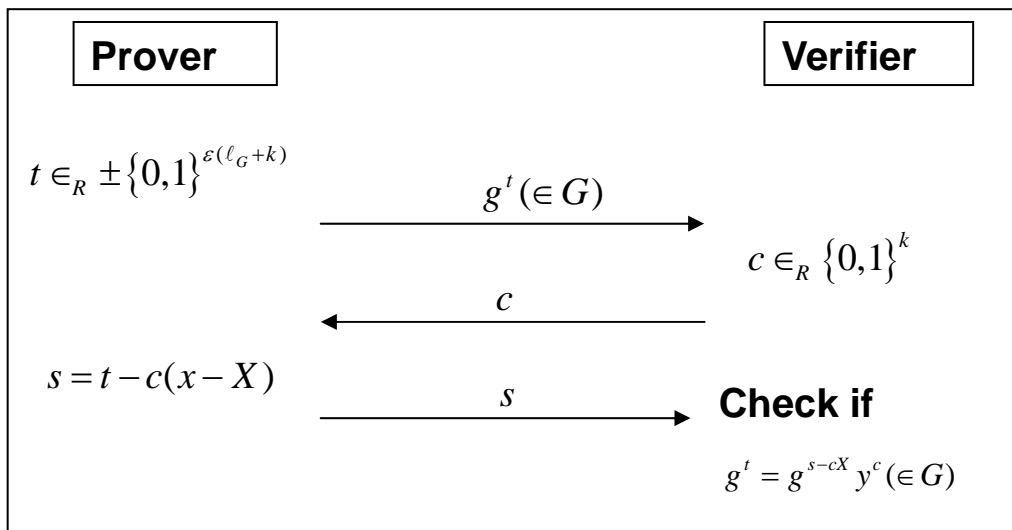


圖表 2. 兩離散對數值相等性的知識簽章

接下來是一個與定義 5 類似的版本，證明的是知道一個落在給定區間的離散對數值。



**定義 7. 離散對數值落在給定區間的知識簽章 (Signature of knowledge of discrete logarithm lying in a given interval)** [10]: 令  $y, g$  為  $G$  中的元素。一個數對  $(c, s) \in \{0,1\}^k \times \pm \{0,1\}^{\varepsilon(\ell_G+k)+1}$  是一個證明者知道  $y = g^x$  對基底  $g$  且落在區間  $]X - 2^{\varepsilon(\ell_G+k)}, X + 2^{\varepsilon(\ell_G+k)}[$  的離散對數值，對一訊息  $m \in \{0,1\}^*$  造出的知識簽章。驗證者可由是否  $c = H(y \| g \| g^{s-cx} y^c \| m)$  來驗證。此簽章造法為由一知道  $x$  者，任選一隨機的  $t \in \pm \{0,1\}^{\varepsilon(\ell_G+k)}$  再算出  $c = H(y \| g \| g^t \| m)$  與  $s = t - c(x - X)$ 。下圖即為此證明系統的原始的互動式版本。(附註：即使證明者知道的  $x$  介於區間  $]X - 2^{\ell_G}, X + 2^{\ell_G}[$  裡，此知識簽章仍只能證明證明者所知之秘密值介於  $]X - 2^{\varepsilon(\ell_G+k)}, X + 2^{\varepsilon(\ell_G+k)}[$  )。

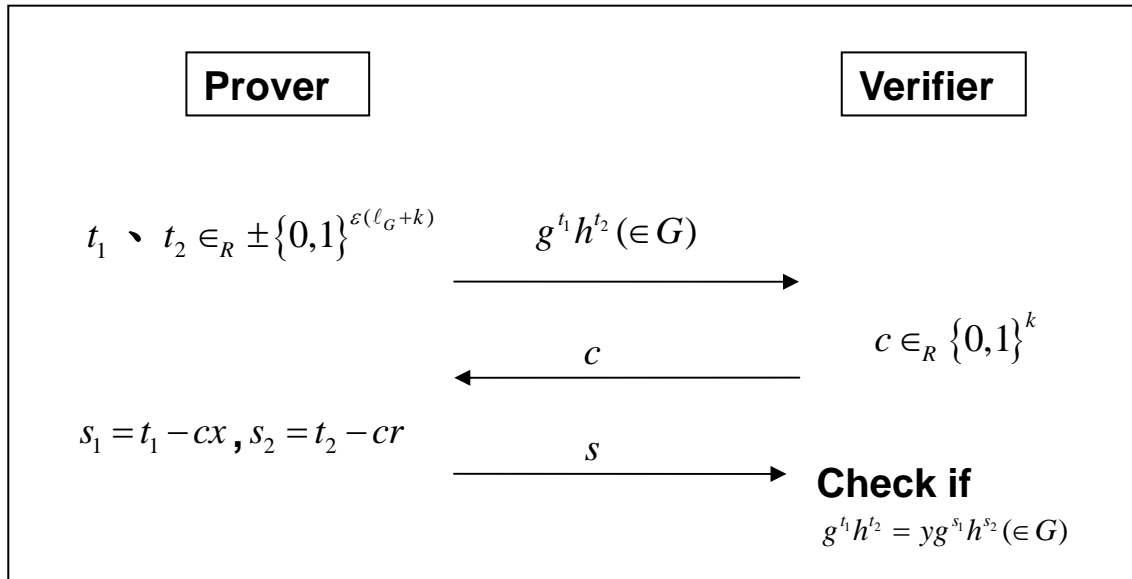


圖表 3. 離散對數值落在給定區間的知識簽章

另一類是有關表示方式的知識簽章。其實定義 5 的離散對數也可以看成是某數對單一基底的表示方式。我們舉較單純的兩基底模式來作介紹，更多基底的模式可類推。

**定義 8. 表示方式的知識簽章 (Signature of knowledge of a representation)** 令  $y$ 、 $g$ 、 $h$  為  $G$  中的元素。一值組  $(c, s_1, s_2) \in \{0,1\}^k \times \pm\{0,1\}^{\varepsilon(\ell_G+k)+1} \times \pm\{0,1\}^{\varepsilon(\ell_G+k)+1}$  是一個證明者知道  $y = g^x h^r$  相對於基底  $g$ 、 $h$  的表示方式，對一訊息  $m \in \{0,1\}^*$  造出的知識簽章。驗證者可由是否  $c = H(y \| g \| h \| y^c g^{s_1} h^{s_2} \| m)$  來驗證。此簽章造法為由一知道  $x$  者，任選一兩個隨機的  $t_1$ 、 $t_2 \in \pm\{0,1\}^{\varepsilon(\ell_G+k)}$  再算出  $c = H(y \| g \| h \| g^{t_1} h^{t_2} \| m)$  與  $s_1 = t_1 - cx$ 、 $s_2 = t_2 - cr$ 。下圖即為此證明系統的原始的互動式版本。





圖表 4. 表示方式的知識簽章





## 第三章 一個具層級性的動態群體簽章系統

本文中底層所採用的群體簽章系統和[2]中曾提到，應用兩種不同的簽章系統，來結合出群體簽章系統的作法是一樣的。第一個簽章系統的目的在於管理者需簽署一個憑證給要加入群體的使用者，而第二個簽章系統才真正用來簽出群體簽章。在第二個簽章系統中運用了知識簽章的技巧，以使群體簽章的大小不用再與群體成員個數相關[10]。廢止成員的觀念，採用的是[15]中的新作法，在系統中加入時間的觀念，讓成員可以自行更新群體簽章金鑰，來達成一個廢止機制。大致來說：我們的系統是以[15]中所提的群體簽章系統為基礎，導入層級的觀念並作相關修改。新增了一開始的層級初始步驟以決定層級間的關係。並在參與程序中增加可限定使用者合法時限的功能。

### 第一節 參數定義

本系統主要是以[15]中的群體簽章系統加以演變，大部分參數將繼續沿用，但仍有為了避免混淆而作的修改與新增部分，因此還是將所有系統所需參數介紹如下。

a.  $\varepsilon > 1$ 、 $h$ 、 $l_p$  為系統的安全參數， $\lambda_1$ 、 $\lambda_2$ 、 $\gamma_1$ 、 $\gamma_2$  的取法需滿足

以下的關係：

$$(1) \lambda_1 > \varepsilon(\lambda_2 + h) + 2$$

$$(2) \lambda_2 > 4l_p$$

$$(3) \gamma_1 > \varepsilon(\gamma_2 + h) + 2$$

$$(4) \gamma_2 > \lambda_1 + 2$$

b. 定義如下的區間  $\Lambda = [2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}]$ 、 $\Gamma = [2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}]$ ，

$$\Gamma_i = \left[ 2^{\gamma_1} - \left( 1 - \frac{2(i-1)}{\tau} \right) \cdot 2^{\gamma_2}, 2^{\gamma_1} - \left( 1 - \frac{2i}{\tau} \right) \cdot 2^{\gamma_2} \right], \tau \text{ 為時間區間分割的總}$$

數，其中  $1 \leq i \leq \tau$ 。

c.  $H: \{0,1\}^* \rightarrow \{0,1\}^h$ 、 $H': (\{0,1\}^*, n) \rightarrow \mathbb{Z}_n$  為兩抗碰撞之雜湊函數。 $H$  為原本在簽章程序中所需用到的。 $H'$  為新定義的雜湊函數，目的在於隨機在  $\mathbb{Z}_n$  中選出一個數來當作群體管理者的私密金鑰。

d.  $T$  與  $T$  與  $\tau$  皆不同。

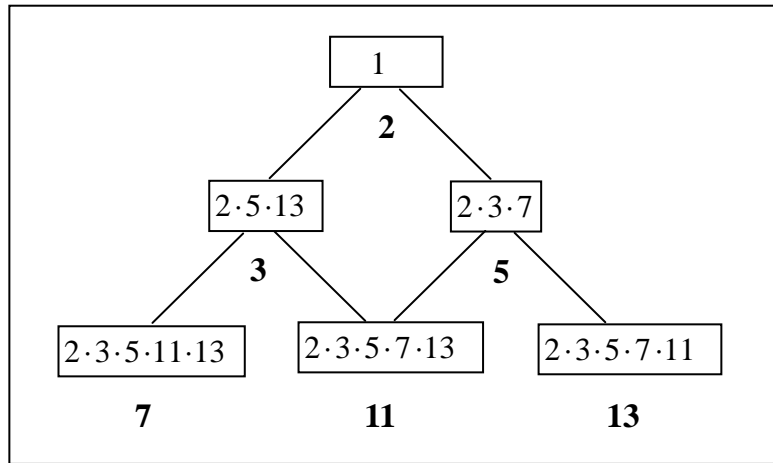
## 第二節 一個具層級性的動態群體簽章系統

### 1. 層級初始 (Hierarchy initialization)

1.1. 決定各群體的階級高低及權力領域之關係，繪出類似如下之似樹狀圖，每個節點代表一個群體，各節點間線段相連部份表示上層群體管理者有權限可解開下層群體之簽章。

1.2. 對每個群體  $G_i$  各指定一個不同的質數  $P_i$ ，如附圖各節點下方之值所示。

1.3. 計算各群體所擁有的  $T_i = \prod_{G_j \preceq G_i} P_j$ ，其中  $\preceq$  為 partial order。我們標示在下圖各節點內。



圖表 5. 層級初始程序步驟 a、b、c 圖例

1.4. 由一公正之機構(系統管理者)隨機選出  $K_0$  和兩個大質數  $P$ 、 $Q$ 。算出

$M = P \cdot Q$  且將  $M$  公開。使得各群體管理者  $GM_i$  的私密值為

$$K_i = f_{T_i}(K_0) = (K_0)^{T_i} \pmod{M}。$$

我們定義若  $G_i$  的層級比  $G_j$  高且可管轄  $G_j$ ，在圖上的表示方式即

$G_j$  為  $G_i$  的子孫節點，記為  $G_j \leq G_i$ ，若無此關係，則可記為  $G_j \not\leq G_i$ 。步驟

1.3 的意思即是將所有群體  $G_i$  所不能管轄之群體的  $P$  值連乘起來，即為  $T_i$  的值。這樣的設定方式主要是採用了由[1]中建議的層級式的金鑰推導法來的，關於其安全性也有簡單的證明，我們後面會再詳細的說明其安全。

## 2. 群體設定 (Group setup)

2.1. 對每個群體  $G_i$ ，管理者  $GM_i$  選擇兩個強質數  $p_i = 2p_i' + 1$ 、

$$q_i = 2q_i' + 1，長度各為 \frac{h}{2} 位元長，再算出  $n_i = p_i \cdot q_i$ 。$$

2.2. 管理者  $GM_i$  算出  $x_i = H'(K_i, n_i)$  其中  $x_i \in \mathbb{Z}_{n_i}$ 。

2.3. 管理者  $GM_i$  任意由  $QR_{n_i}$  中選擇  $a_i$ 、 $a_{0i}$ 、 $g_i$ 、 $h_i$ ，並將

$y_i = g_i^{x_i} \pmod{n_i}$  公佈。如此我們就有  $(n_i, a_i, a_{0i}, y_i, g_i, h_i)$  為群體  $G_i$  的公開值，以及各群體管理者  $GM_i$  會擁有一把私密金鑰  $(p_i, q_i, x_i, K_i)$ 。

**2.4.** 任選滿足  $e_{i,k} \in \Gamma_k$  的質數  $e_{i,1}, e_{i,2}, \dots, e_{i,r}$ ，同時將它們公開。

在此群體設定過程中，步驟 2.1 為各群體自行選擇  $p_i, q_i$  值，可不受限制，即各自初始一個 [15] 中的群體。在步驟 2.2 中所用的  $H': (\{0,1\}^*, n) \rightarrow \mathbb{Z}_n$  為一個抗碰撞的雜湊函數，其實作方式[25]可以用一個標準的雜湊函數  $h: \{0,1\}^* \rightarrow \{0,1\}^\ell$ ，其中  $\ell = \lceil \log_2 n \rceil$  即為  $n$  的長度。若  $h(x) < n$ ，則輸出  $H'(x, n) = h(x)$ 。否則就輸出  $H'(x, n) = h(x) - \lfloor \frac{n}{2} \rfloor$ 。如此即可讓  $\mathbb{Z}_n$  中的任意一個元素  $b$ ，滿足  $\left| \Pr[H'(x, n) = b] - \frac{1}{n} \right| < 2^{-\ell}$ ，則  $H'$  可接近均勻分布(Uniform distribution)。步驟 2.3、2.4 仍依[15]中設定。



### 3. 參與程序 (Join)

**3.1.** 當一個使用者  $j$  要加入群體  $i$  時，記為  $U_{i,j}$ ，假設給定其可合法代替群體簽章的時間為  $c$  到  $d$ 。

**3.2.**  $U_{i,j}$  產生兩私密值  $\tilde{x}_{i,j} \in_R [0, 2^{2\lambda}]$ 、 $\tilde{r} \in_R [0, n_i^2]$ 。送出  $C_1 = g^{\tilde{x}_{i,j}} \cdot h^{\tilde{r}} \pmod{n_i}$  給管理者  $GM_i$  並且證明自己知道  $C_1$  以  $g$  和  $h$  為底的表示方法，證明方式可參考定義 8。

**3.3.** 管理者  $GM_i$  確認是否  $C_1 \in QR_{n_i}$ 。若為真，則選取  $\alpha_{i,j}, \beta_{i,j} \in_R [0, 2^{2\lambda}]$  送給  $U_{i,j}$ 。

**3.4.**  $U_{i,j}$  算出  $x_{i,j} = 2^{\lambda_1} + (\alpha_{i,j} \cdot \tilde{x}_{i,j} + \beta_{i,j} \pmod{2^{2\lambda_2}})$ ，和  $C_2 = a_i^{x_{i,j}} \pmod{n_i}$ 。將  $C_2$

送給  $GM_i$  且證明下列事實：

i.  $\log_{a_i} C_2 \in \Lambda$

ii. 知道一組  $u, v, w$  使得

$$(1) \log_{a_i} \frac{C_2}{a_i^{2^{\lambda_1}}} \in [-2^{\lambda_2}, 2^{\lambda_2}]$$

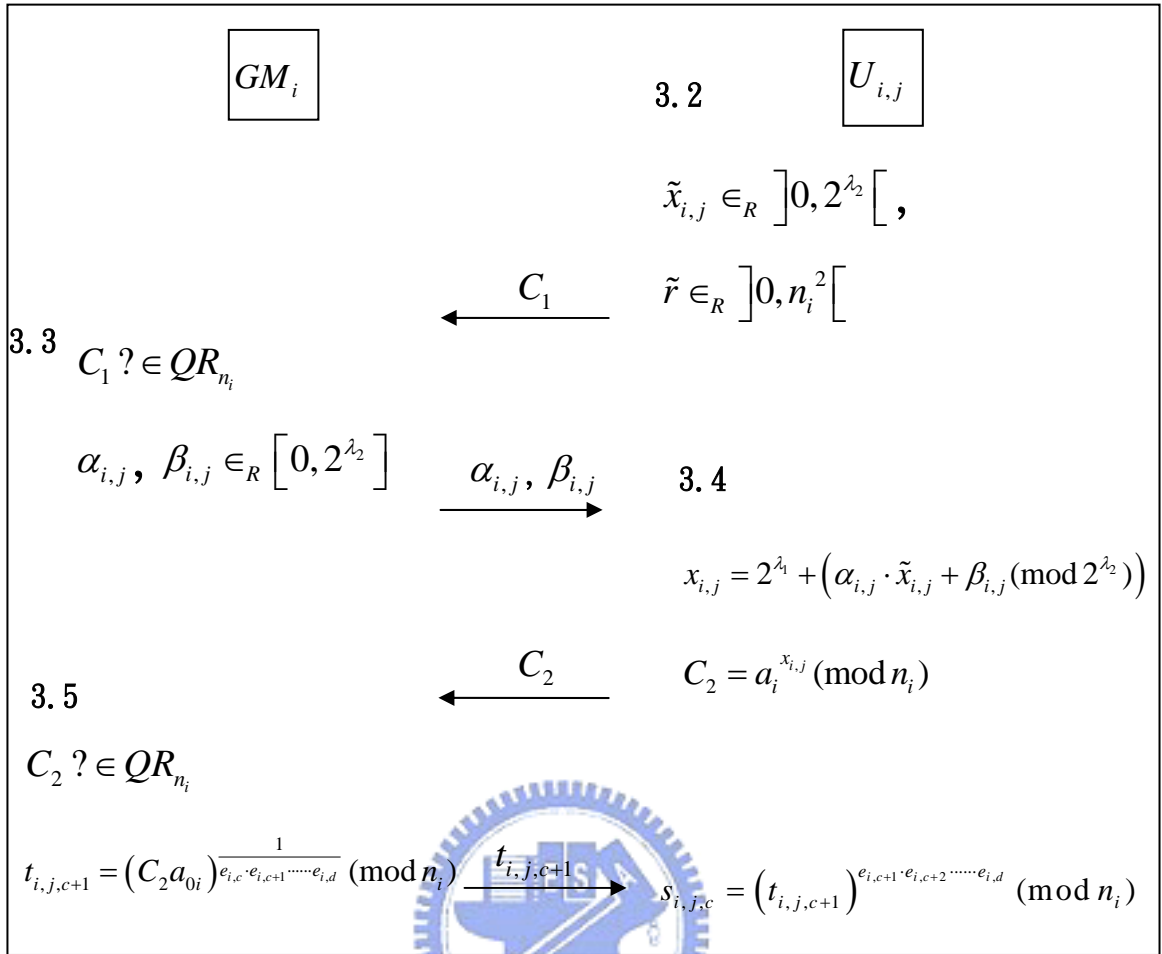
$$(2) C_1^{\alpha_{i,j}} g_i^{\beta_{i,j}} = g_i^u \left( g_i^{2^{\lambda_2}} \right)^v h^w \pmod{n_i}$$

同樣可用定義提供的方法來提出知識簽章。

**3.5.** 管理者  $GM_i$  確認是否  $C_2 \in QR_{n_i}$ 。若是的話，管理者  $GM_i$  算出

$t_{i,j,c} = (C_2 a_{0i})^{\frac{1}{e_{i,c} \cdot e_{i,c+1} \cdots e_{i,d}}} \pmod{n_i}$  且將值送給  $U_{i,j}$ 。此時，使用者  $U_{i,j}$  在時間  $c$  的私密金鑰即為  $sk_{i,j,c} = (x_{i,j}, s_{i,j,c}, t_{i,j,c+1})$  其中  $s_{i,j,c} = (t_{i,j,c})^{e_{i,c+1} \cdot e_{i,c+2} \cdots e_{i,d}} \pmod{n_i}$ ， $t_{i,j,c+1} = (t_{i,j,c})^{e_c} \pmod{n_i}$ 。





圖表 6. 參與程序

參與程序中的步驟 3.2、3.3、3.4 與[15]中的作法大致相同。為了方便指定成員的合法區間，我們的架構中已經把  $e_{i,1}$ 、 $e_{i,2}$ ..等值視為群體中的公開值了，所以並不需有[15]中的 seed 部分。在步驟 3.5 中的作法，也就是為了要使成員加入時，能夠先指定好其合法區間，過期就不能再代表群體簽章。當使用者拿到  $t_{i,j,c}$  之後，就可以算出他在最初合法時間 c 時的憑證，也是簽章金鑰中的  $s_{i,j,c} = (t_{i,j,c})^{e_{i,c+1} \cdot e_{i,c+2} \cdots e_{i,d}} \pmod{n_i}$  與更新元件  $t_{i,j,c+1} = (t_{i,j,c})^{e_{i,c}} \pmod{n_i}$ 。這個群體簽章原理主要就是把成員憑證當作簽章金鑰使用，簽章時用系統的公開金鑰來加密這個憑證，以便讓群體的管理者可以用此群體的私密金鑰來解密出這個憑證，達成解開簽章的部分。

#### 4. 更新 (Update) (使用者 $U_{i,j}$ 由時間 $k$ 至 時間 $k+1$ )

在時間  $k$  時，使用者  $U_{i,j}$  所擁有的私密金鑰為  $sk_{i,j,k} = (x_{i,j}, s_{i,j,k}, t_{i,j,k+1})$ ，在進入時間  $k+1$  後，將所持金鑰更新為  $sk_{i,j,k+1} = (x_{i,j}, s_{i,j,k+1}, t_{i,j,k+2})$  其中  $s_{i,j,k+1} = (t_{i,j,k+1})^{e_{i,k+2} \cdot e_{i,k+3} \cdots e_{i,d}} \pmod{n_i}$ ， $t_{i,j,k+2} = (t_{i,j,k+1})^{e_{i,k+1}} \pmod{n_i}$ 。

這是一個要達成前進式安全 (Forward secure) 必需的作法，讓每個時段使用者的成員憑證不同，可以往後更新但無法往前回溯出之前的憑證，所以簽章金鑰推導的過程必須是一個不可逆的程序，在更新過後即將舊的金鑰和更新元件丟棄，如此便能保證在時段  $k$  入侵的攻擊者無法得到時段  $k$  之前的簽章金鑰來偽造時段  $k$  之前的簽章，這樣才能達到前進式安全的特性。這種觀念和更新方法主要是由[16]中來的。而我們又在參與程序根據欲給定使用者的時限有技巧的給予憑證，可以讓一個在時段  $c \sim d$  間為合法的成員，到達時段  $d$  之後，即無法繼續更新出合法的簽章金鑰，因此可達到使用者時限 (Time-bounded) 的設定。

#### 5. 廢止 (Revoke) (在時段 $k$ 時廢止使用者 $U_{i,j}$ ，其合法時段為 $c \sim d$ )

##### 5.1. 管理者 $GM_i$ 利用更新的演算法算出 $s_{i,j,k}$ 、 $s_{i,j,k+1}$ 、 $\dots$ 、 $s_{i,j,d}$ (即 $U_{i,j}$

在時段  $k$  之後到時段  $d$  之間所有時期之憑證，亦為簽章之私密金鑰)，並依其對應時間將其分別公佈在廢止憑證列表  $RL_{i,k}$ 、 $RL_{i,k+1}$ 、 $\dots$ 、 $RL_{i,d}$  中。

##### 5.2. 管理者發出訊號指示系統行進到下一個時間。

在時段  $k$  時，若需對一成員，撤除其可合法代替群體簽章的權力，也就是在給定某使用者合法時限後，時限未到即需將某使用者當時與之後的憑證

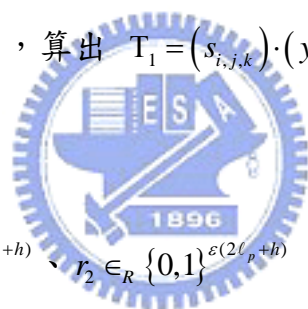


廢止的話，就必須執行此廢止程序。實際作法是系統需維護一組憑證廢止列表： $\{RL_{i,k} \mid 1 \leq k \leq \tau, i \text{ 為群組}\}$ 。當欲在時段  $k$  廢止一憑證時，即是利用更新程序算出該成員在時間  $k+1$ 、 $k+2$ 、...、 $d$  的憑證，其中  $d$  為此使用者的合法時限，並分別存入  $RL_{i,k}$ 、 $RL_{i,k+1}$ 、...、 $RL_{i,d}$  中。而這些存在廢止清單的憑證，以後在有需要驗證簽章時，我們會一個一個拿來作一些運算，以確定我們正在驗證的簽章是否由這些憑證所簽出來的。雖然這些廢止的憑證是公開的，每個人也都應該能作驗證簽章的動作，但向後式無連結性會保持將廢止憑證公開後的安全。

## 6. 簽章 (Sign) (使用者 $U_{i,j}$ 在時間 $k$ )

6.1. 隨機選取  $\omega \in_R \{0,1\}^{2\ell_p}$ ，算出  $T_1 = (s_{i,j,k}) \cdot (y_i)^\omega \pmod{n_i}$  以及

$$T_2 = (g_i)^\omega \pmod{n_i}。$$



6.2. 隨機選取  $r_1 \in_R \{0,1\}^{\varepsilon(\lambda_2+h)}$ 、 $r_2 \in_R \{0,1\}^{\varepsilon(2\ell_p+h)}$  並算出下列各值：

$$\text{i. } d_1 = \frac{(T_1)^{e_{i,k}}}{(a_i)^{r_1} \cdot (y_i)^{e_{i,k} \cdot r_2}} \pmod{n_i}$$

$$\text{ii. } d_2 = (g_i)^{r_2} \pmod{n_i}$$

$$\text{iii. } c = H(g_i \parallel y_i \parallel a_i \parallel a_{0i} \parallel T_1 \parallel T_2 \parallel d_1 \parallel d_2 \parallel k \parallel e_{i,k} \parallel m)$$

$$\text{iv. } s_1 = r_1 - c(x_{i,j} - 2^{\lambda_1})$$

$$\text{v. } s_2 = r_2 - c\omega$$

6.3. 輸出簽章為  $\sigma = (c, i, k, e_{i,k}, s_1, s_2, T_1, T_2)$ 。



簽名程序中，主要是以 ElGamal 的加密方法來加密簽章者所持的憑證，再以知識簽章的技巧來證明簽章者知道某些秘密值。加密的部分用群體(管理者)的公開金鑰值中的  $(y_i, g_i)$  來加密成員  $U_{i,j}$  在時間  $k$  的憑證(簽章金鑰)  $s_{i,j,k}$ 。這樣一來，在需要解開簽章的時候，群體管理者只要用自己的私密金鑰就可以解開加密值以獲得憑證。

為了要確定簽章者的確是利用當時段的憑證  $s_{i,j,k}$  來簽章，所以用了步驟 6.2 這樣的證明過程。 $d_1$  是為了證明所加密的憑證值  $s_{i,j,k}$  確實是成員  $U_{i,j}$  在時間  $k$  的憑證，也就是滿足  $s_{i,j,k} = (a^{x_{i,j}} a_0)^{\frac{1}{e_{i,k}}} \pmod{n_i}$ 。 $d_2$  的取法則是為了要證明簽章者知道私密值  $x_{i,j}$ 。只有在知道憑證  $s_{i,j,k}$  和私密值  $x_{i,j}$  的情況下，才能夠不管步驟 6.2.iii 中挑戰值  $c$  的值為何，皆能通過一個零知識的證明系統之驗證。



## 7. 驗證 (Verify) (針對某一簽章 $\sigma = (c, i, k, e, s_1, s_2, T_1, T_2)$ )

7.1. 確認是否  $e \in \Gamma_k$ 。若否，則此簽章有誤。

7.2. 算出  $D_1 = \frac{(a_{0i})^c T_1^{e(1-c)}}{(a_i)^{s_1 - c \cdot 2^{\lambda_i}} \cdot (y_i)^{e \cdot s_2}} \pmod{n_i}$ 、 $D_2 = T_2^c \cdot (g_i)^{s_2} \pmod{n_i}$ ，並確認是否

$c = (g_i \parallel y_i \parallel a_i \parallel a_{0i} \parallel T_1 \parallel T_2 \parallel D_1 \parallel D_2 \parallel k \parallel e \parallel m)$ 。若否，則輸出驗證失敗。

7.3. 確認是否  $s_1 \in \pm\{0,1\}^{\varepsilon(\lambda_2+h)+1}$ ，以及  $s_2 \in \pm\{0,1\}^{\varepsilon(2\ell_p+h)+1}$ 。若否，則輸出驗證失敗。

7.4. 接著重複以下步驟查詢廢除憑證列表：

- i. 驗證者將  $(s_2, T_2, e, c)$  送給管理者  $GM_i$ 。

- ii.  $GM_i$  算出  $C = \left( (y_i)^{s_2} \cdot (T_2)^{c \cdot x_i} \right)^e \pmod{n_i}$ ，並隨機選取  $g_0 \in_R QR_{n_i}$ ，再將  $(g_0, g_0^C)$  傳回給驗證者。
- iii. 驗證者由廢止憑證列表中取出一憑證  $A$ ，算出  $D' = (y_i)^{e \cdot s_2} (T_1)^{e \cdot c} (A)^{-e \cdot c} \pmod{n_i}$ 。
- iv. 確認是否  $g_0^C = g_0^{D'}$ 。若是，則此簽章由此廢止憑證所簽，輸出此簽章驗證結果“失敗”；否則重複步驟 7.4 驗證廢止憑證列表  $RL_{i,k}$  上所有憑證。

驗證的方法仍然和[15]類似，前半部為驗證簽章是否為正確格式。在步驟 7.1 檢查  $e$  值範圍是否落在  $\Gamma_k$  中，避免以特定的  $e$  值捏造簽章中的證明過程之慮。步驟 7.2 則是利用比較簽章中的雜湊值  $c$  是否等於我們所算出的  $c'$ ，用來達成原本零知識互動式證明系統改成非互動式之後的驗證工作，只有真正知道正確的憑證和秘密值才能通過此驗證步驟。步驟 7.4 則是檢查證明系統中回應 (Response) 值的範圍是否正確，同樣可將捏造證明過程的可能性降至最低。若使用者超過其合法簽章時限，依照我們在參與程序中的設定方法，使用者沒辦法利用更新程序來造出簽章當時所需的私密金鑰中的憑證，因此這樣的過期使用者無法造出一個成功的零知識非互動式證明系統，就沒辦法通過這部分的驗證。

驗證程序的後半部，則是要檢查此憑證是否已在給定合法期限內提前被廢止。驗證的方法是要將正在驗的簽章內部所隱藏的憑證與此時期被廢止的憑證一個一個比對，若有相同，此簽章即為無效。由於驗證程序是要讓每個人都能驗證，但卻不能洩漏簽出此簽章的憑證，因此這部分的驗證必須與系統中擁有能解開簽章中憑證能力的群體管理者作溝通。方法則如步驟 7.4 所

示，傳送此簽章中的  $(s_2, T_2, e, c)$  部分，讓群體管理者用解開簽章用的私密金鑰  $x_i$  算出  $C = \left( (y_i)^{s_2} \cdot (T_2)^{c \cdot x_i} \right)^e \pmod{n_i}$ 。因為在簽章程序的知識簽章中有  $s_2 = r_2 - c\omega$  的關係，所以  $(y_i^e)^{r_2} = (y_i^e)^{s_2} (y_i^{e\omega})^c$ ，此等式左方即為  $C$  值，其中不含目前所驗簽章之憑證資料，但含有群體私密金鑰  $x_i$  加入運算；令右方為  $D$ ，則因為  $C = D$ ，所以  $g_0^C = g_0^D$  應該成立。群體管理者將  $C$  值以  $(g_0, g_0^C)$  的形式隱藏在指數並回傳給驗證者。而  $D$  可改寫成如下形式  $D = (y_i)^{e \cdot s_2} (T_1)^{e \cdot c} (S)^{-e \cdot c} \pmod{n_i}$ ，其中  $S$  為此簽章的憑證。 $T_1$  含此簽章的憑證資料，但若  $S$  確為簽出此簽章的憑證，則  $(S)^{-e \cdot c}$  會將憑證的資料抵銷。因此我們比對憑證撤銷列表的步驟即是將列表上的憑證一一代入  $D' = (y_i)^{e \cdot s_2} (T_1)^{e \cdot c} (A)^{-e \cdot c} \pmod{n_i}$  中的  $A$ ，求出  $D'$ 。若  $g_0^C = g_0^{D'}$  則此群體簽章是由在憑證廢止列表中的憑證所簽出，故為無效的群體簽章。

8. 解章 (Open) (當管理者  $GM_i$  需打開簽章  $\sigma = (c, i, k, e, s_1, s_2, T_1, T_2)$ )

8.1. 算出  $x_i = H \left( \left( \begin{array}{c} T_1 \\ K_i^{T_1}, n_i \end{array} \right) \right)$ 。

8.2. 由驗證演算法驗證簽章  $\sigma$ 。

8.3. 算出此簽章所用的憑證  $s = \frac{T_1}{T_2^{x_i}} \pmod{n_i}$ 。

8.4. 管理者需提出證明知道  $\log_{g_i} y_i = \log_{T_2} \left( \frac{T_1}{S} \pmod{n_i} \right)$  之值，並輸出  $s$ 。

解開簽章的步驟首先是要能夠取得該簽章所屬群體的私密金鑰，用來解開在簽章時所作的 ElGamal 加密。按照我們對  $x_i$  這個值的設定方法，若群

體管理者  $GM_i$  有權限能解開群體  $G_i$  的簽章的話，那麼  $T_i$  就會被  $T_i$  所整除，因此步驟 8.1 中  $(K_i)^{\frac{T_i}{T_i}} = (K_0)^{\frac{T_i}{T_i}} = (K_0)^{T_i} = K_i \pmod{M}$  可求得  $K_i$ ，再用相同於群體設定時的抗碰撞雜湊函數  $H'$  將  $K_i$  對應至一個  $\mathbb{Z}_{n_i}$  中的值，此值即為群體  $G_i$  的私密金鑰  $x_i$ 。然後經過驗證程序確定此簽章為正確的形式且非被廢止的憑證所簽出，因為在驗證程序的後半部需用到  $x_i$ ，因此步驟 8.1 必須先作。步驟 8.3 則是用  $x_i$  來作 ElGamal 解密，算出的  $s$  即為此簽章的憑證。最後在步驟 8.4 再附上一個知識簽章，證明前一步驟中用來解密的  $x_i$  確實為群體  $G_i$  的私密金鑰  $x_i$ 。



## 第四章 安全性證明

要證明我們系統的安全性，大約可分為幾部分：(1)各群體的群體簽章架構是安全的。(2)層級架構中用來解開群體簽章的私密金鑰，確實只能由上層的管理者往下層推導。(3)若下層的成員兼為上層的管理者，本文架構之安全也不會受到影響。(4)對[15]群體簽章架構中的改變，使用者加入群體時可決定其合法時限之設計，可以證明是安全的。第一節介紹的是第(1)部分，由於我們的層級架構是以[15]為基礎，因此我們會介紹此架構中滿足的安全性與相關證明步驟，並加以適當的修改，以符合本文中之架構，原始版本詳細步驟可參閱[15]。第二節介紹(2)、(3)、(4)，是將基礎架構運用在我們的系統中所需考慮的其他相關證明。

### 第一節 基礎架構的安全性

有關[15]中群體簽章架構的安全性，在該篇文中已有詳細證明，本節會簡介其證明過程並重新考慮在本系統中之安全性。

**定理1.** 在強 RSA 的假設條件之下，[15]中的群體簽章系統是建立在對於成員憑證和成員私密值的統計性零知識互動式證明。

**證明：**根據之前關於零知識互動式證明系統的介紹，一個零知識的互動式證明系統，必須滿足完整性、完美性、和零知識。

1. **完整性：**若證明者確實知道秘密值，則一定能通過證明。這部分可顯而易見地透過簽章與驗證程序來確定：若簽章者確實知道當時間的憑證和成員的私密值，則必可簽出能驗證通過的正確簽章。
2. **完美性：**若有一證明者  $P^*$  有很大的機率能說服驗證者  $V$ ，則須存

在一個知識擷取者  $E$ ， $E$  可利用  $P^*$  與回轉 (rewind) 的技巧來擷取出證明者欲證明之秘密值。因為此簽章系統要證明的是知道憑證值與成員的私密值，我們這裡以憑證值為例，敘述知識擷取者  $E$  的作法，成員私密值的擷取法亦類似。

(a) 假設存在證明  $P^*$  有很大的機率能說服驗證者  $V$ ，那表示在多项式時間內可以經由回轉的技巧，針對相同的 committed 值，對不同的挑戰值 (Challenge)，得到不同的回應值 (Response)。因此我們可得到兩組合理的值組 (Tuple)： $(c, i, k, e_k, s_1, s_2, d_1, d_2, T_1, T_2)$  與  $(\tilde{c}, i, k, e_k, \tilde{s}_1, \tilde{s}_2, d_1, d_2, T_1, T_2)$ ，其中兩值組的 committed 值  $d_1$ 、 $d_2$  都一樣。

(b) 我們可由兩值組間  $d_2$  的相等關係來找出  $\omega$ ，再利用  $T_1 / y_i^\omega$  來取得憑證值。由  $d_2 \equiv (g_i)^{s_2} \equiv (g_i)^{c\omega + s_2} \equiv T_2^c g_i^{s_2} \equiv T_2^{\tilde{c}} g_i^{\tilde{s}_2} \pmod{n}$ ，得到  $g_i^{s_2 - \tilde{s}_2} \equiv T_2^{\tilde{c} - c} \pmod{n}$ 。令  $\delta = \gcd(s_2 - \tilde{s}_2, \tilde{c} - c)$ ，則可用延伸的歐基里德演算法 (Extended Euclidean algorithm) 來得到  $\alpha, \beta \in \mathbb{Z}$  滿足  $\delta = \alpha(s_2 - \tilde{s}_2) + \beta(\tilde{c} - c)$ 。故  $g_i \equiv g_i^{\frac{\alpha(s_2 - \tilde{s}_2) + \beta(\tilde{c} - c)}{\delta}} \equiv (T_2^\alpha g_i^\beta)^{\frac{\tilde{c} - c}{\delta}} \pmod{n}$ 。從  $\tilde{c} - c$  與  $\delta$  的大小關係分三種情況討論：

i.  $\tilde{c} - c < \delta$ ：不可能，因為  $\delta$  為  $\tilde{c} - c$  與  $s_2 - \tilde{s}_2$  的最大公因數。

ii.  $\tilde{c} - c > \delta$ ：不可能。若為真則上方等式

$$g_i \equiv (T_2^\alpha g_i^\beta)^{\frac{\tilde{c} - c}{\delta}} \pmod{n} \text{ 中的 } \frac{\tilde{c} - c}{\delta} > 1, \text{ 已解決了強 RSA 的問}$$

題，違背其假設。



iii. 故可知  $\tilde{c} - c = \delta$ ，且  $\frac{s_2 - \tilde{s}_2}{\tilde{c} - c}$  之值為某整數  $\pi$ 。由於

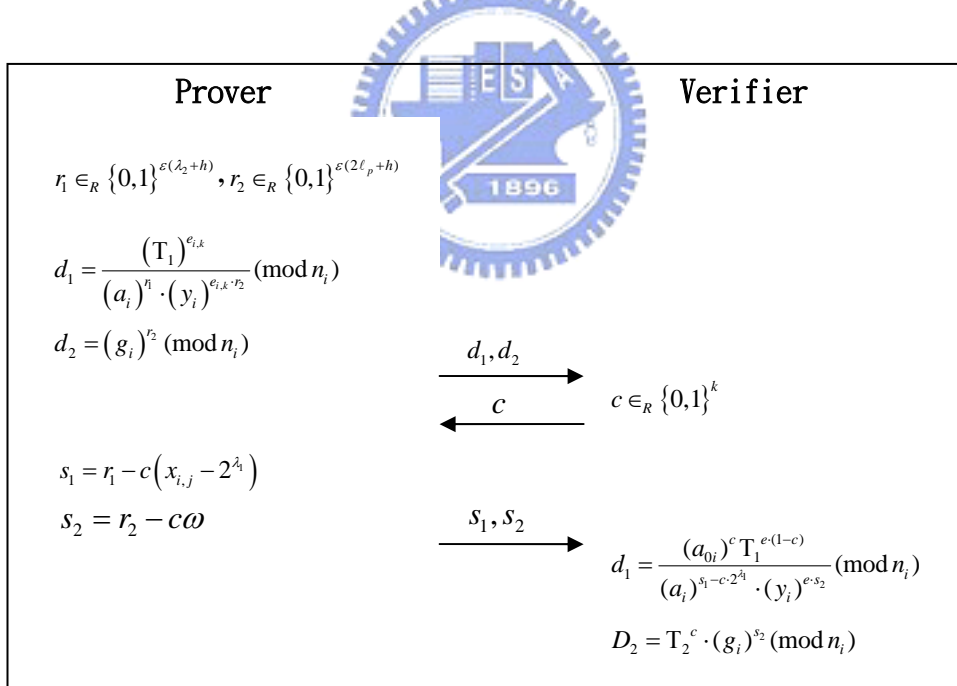
committed 值  $d_2$  相等，因此  $s_2 = r_2 - c\omega$  且  $\tilde{s}_2 = r_2 - \tilde{c}\omega$ ，

可得到  $\omega = \frac{s_2 - \tilde{s}_2}{\tilde{c} - c} = \pi$ 。最後得到憑證  $s_{i,j,k} = \frac{T_1}{y_i^\pi} \pmod{n}$ 。

(c) 成員的私密值  $x_{i,j}$  亦可用類似技巧擷取出來，主要是利用兩值組

間  $d_1$  的相等性得出關係式，其餘步驟與擷取  $\omega$  值類似。

3. **零知識**：這裡需要證明的是我們可以造出一個模擬器  $M$ ，讓證明者和驗證者真正經由協定跑出來的 view 可以和這個模擬器造出來的 view，成統計上的不可分辨 (Statistical indistinguishable)。我們的簽章程序所作的零知識互動式證明系統可以下圖表示。



圖表 7. 簽名程序的互動式證明系統

依下列步驟[15]造出一模擬器：

(a) 依均勻分布 (Uniform distribution) 隨機選取  $c' \in_R \{0,1\}^k$ 、

$$s'_1 \in_R \{0,1\}^{\varepsilon(\lambda_2+k)} \text{、} s'_2 \in_R \{0,1\}^{\varepsilon(2\ell_p+k)} \text{。}$$

$$(b) \text{ 計算 } d'_1 = \frac{a_0^{c'} T_1^{e_{i,k}(1-c')}}{a^{s'_1-c'2^{\lambda_1}} y^{e_{i,k}s'_2}} (\text{mod } n) \text{、} d'_2 = T_2^{c'} g^{s'_2} (\text{mod } n) \text{。}$$

(c) 輸出值組  $(c', s'_1, s'_2, d'_1, d'_2)$ 。

因為由上圖中，真正由證明者和驗證者經協定跑出來的正確值

$$\text{組 } (c, s_1, s_2, d_1, d_2) \text{ 會滿足 } d_1 = \frac{a_0^c T_1^{e_{i,k}(1-c)}}{a^{s_1-c2^{\lambda_1}} y^{e_{i,k}s_2}} (\text{mod } n) \text{ 與}$$

$d_2 = T_2^c g^{s_2} (\text{mod } n)$ ，因此只要證明模擬器跑出來的結果

$\Pr[C'=c, S'_1=s_1, S'_2=s_2, D'_1=d_1, D'_2=d_2]$  和真正協定跑出的結果

$\Pr[C=c, S_1=s_1, S_2=s_2, D_1=d_1, D_2=d_2]$  是統計上不可分辨的，即表示

驗證者經與證明者間的互動式證明系統(即驗證簽章)所會得到的資訊，也可由驗證者自己透過這個模擬器得到與其為統計上不可分辨的資訊，所以代表這個簽章程序是零知識的，詳細機率分析可參考

[15]。

**定理2.** 對某一群體  $G_i$ ，若給予  $K$  個任意時段的合法成員私密值和更新值  $(x_{i,j}, t_{i,j,k})^1$ ，其中  $1 \leq j \leq K$ ， $k$  為時段，可重複。而令  $k$  值中最小者為  $k_{\min}$ 。每個值組皆有一截止期限  $d_j$ ，其中  $1 \leq j \leq K$ ，令最大的  $d$  值為  $d_{\max}$ 。在強RSA假設下，給予  $K$  組任意時段的合法成員私密值和更新值  $(x_{i,j}, t_{i,j,k})$ ，也無法造出一個新的合法值組  $(\hat{x}, t_{i,h,m})$  其中  $m < k_{\min}$  或

<sup>1</sup>為簡化索引符號，在此將時段 $k$ 時的更新值記為 $t_{i,j,k}$ 。而在架構中，時段 $k$ 時的更新值則為 $t_{i,j,k+1}$ 。



$$m > d_{\max}。$$

**證明：**因為各個時期的憑證值  $s$ ，皆是由更新值  $t$  得來，因此我們考慮簽章時所需的金鑰時，可考慮私密值  $x$  和更新值  $t$  [15]。我們要將強 RSA 的問題轉化到產生新合理值組  $(x_{i,j}, t_{i,j,k})$  的問題。方式是若有一個攻擊者能由  $K$  個  $(x, t)$  值組得到一個私密值  $\hat{x}$  與對應此值而在某個大於時段  $d_{\max}$  或小於時段  $k_{\min}$  的合法  $t$  值，則我們能利用此攻擊者來解強 RSA 問題，違背強 RSA 假設，所以此攻擊者不存在。

假設  $M$  是一個攻擊者，它被允許參加  $K$  次參與程序來獲得這  $K$  個合理值組  $(x_{i,j}, t_{i,j,k})$ ，然後會輸出一組合法值組  $(\hat{x}, t_{i,h,m})$ 。則我們可用一個演算法  $A$ ， $A$  可呼叫  $M$ ，其輸入為強 RSA 問題的輸入  $(n, z)$ ，輸出為滿足  $z \equiv u^e \pmod{n}$  的值組  $(u, e)$ 。 $A$  的步驟如下：

*Algorithm1*( $n, z$ )

1. 隨機選取(1)  $x_{i,1}, x_{i,2}, \dots, x_{i,K} \in_R \Lambda$ ，當作  $K$  組合法的私密值，與(2)  $K$  個時段  $k_1, k_2, \dots, k_K$ ，當作  $K$  組  $t$  值的時段，與(3)  $K$  個時段  $d_1, d_2, \dots, d_K$ ，當作此  $K$  組  $(x, t)$  值組的最末合法時限，即截止時段，其中  $k_j \leq d_j, \forall j \in \{1, \dots, K\}$ 。並亂數選取所有時段滿足  $e_{i,k} \in \Gamma_k$  的  $e$  值。
2. 令  $a \equiv z^{e_{i,k_{\min}} \cdot e_{i,k_{\min}+1} \cdots e_{i,d_{\max}}} \pmod{n}$ 。

3. 隨機選取  $r \in_R \Lambda$ ，再令  $a_{0i} \equiv a_i^r \pmod{n}$ 。
4. 依下列規則，對所有  $j \in \{1, \dots, K\}$ ，給定  $t_{i,j,k}$ ：
  - (1) 若  $k_j \neq k_{\min} \wedge d_j \neq d_{\max}$  則算  $t_{i,j,k} \equiv z^{(x_{i,j}+r)e_{i,k_{\min}} \dots e_{i,k_j-1} e_{i,d_j+1} \dots e_{i,d_{\max}}} \pmod{n}$
  - (2) 若  $k_j = k_{\min} \wedge d_j \neq d_{\max}$  則算  $t_{i,j,k} \equiv z^{(x_{i,j}+r)e_{i,d_j+1} \dots e_{i,d_{\max}}} \pmod{n}$
  - (3) 若  $k_j \neq k_{\min} \wedge d_j = d_{\max}$  則算  $t_{i,j,k} \equiv z^{(x_{i,j}+r)e_{i,k_{\min}} \dots e_{i,k_j-1}} \pmod{n}$
  - (4) 若  $k_j = k_{\min} \wedge d_j = d_{\max}$  則算  $t_{i,j,k} \equiv z^{(x_{i,j}+r)} \pmod{n}$
5. 任選  $g_i, h_i \in_R QR_n$ ， $x \in_R \{1, n^2\}$ ，並令  $n_i = n$ 、 $y_i = g_i^{x_i} \pmod{n_i}$ 。
6. 讓  $M$  跑  $K$  次參與程序，其中輸入的公開參數為  $(n_i, a_i, a_{0i}, y_i, g_i, h_i)$ 。假設我們在跑第  $j$  次，在步驟 3.2 收到 committed 值  $C_1$  後，我們用將  $M$  回轉 (Rewind) 的技巧來擷取出其中能使得  $C_1 = g_i^{\tilde{x}_{i,j}} h_i^{\tilde{r}_{i,j}} \pmod{n_i}$  的  $\tilde{x}_{i,j}$  值和  $\tilde{r}_{i,j}$  值。並算出能滿足  $x_{i,j} = 2^{\lambda_1} + (\alpha_{i,j} \cdot \tilde{x}_{i,j} + \beta_{i,j} \pmod{2^{\lambda_2}})$  的  $\alpha_{i,j}$  與  $\beta_{i,j}$ ，在下一步驟 3.3 送給  $M$ 。接著跑到最後一步驟 3.5 再將相對應於  $x_{i,j}$ ，在第 4 步驟算出的  $t_{i,j,k}$  送給  $M$ 。
7. 當  $M$  跑完全部  $K$  個回合後，輸出一組值組  $(\hat{x}, t_{i,h,m})$ ，對所有  $d$  滿足  $m \leq d \leq \tau$ ，由小到大依序測試何者為值組  $(\hat{x}, t_{i,h,m})$  的最大合法期限  $d$ 。考慮  $m$  的大小：

(1) 若  $m < k_{\min}$ ，則

a. 若  $d < k_{\min}$ ，則輸出  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,d})}{\delta} \right)$ 。

b. 若  $k_{\min} \leq d < k_{\max}$ ，則輸出  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,k_{\min}-1})}{\delta} \right)$ 。

c. 若  $k_{\max} = d$ ，則輸出  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,k_{\min}-1})}{\delta} \right)$ 。

d. 若  $k_{\max} < d$ ，則輸出  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,k_{\max}+1} \cdots e_{i,d})}{\delta} \right)$

(2) 若  $m > d_{\max}$ ，令  $\delta = \gcd(e_{i,m} \cdots e_{i,d}, (\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})$  利用延伸的

歐基里德演算法算出  $\alpha$ 、 $\beta$  使得

$\delta = \alpha(e_{i,m} \cdots e_{i,d}) + \beta((\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})$ ，再輸出強 RSA 問題的答

案  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,d})}{\delta} \right)$ 。

在第 7 步驟的(1)中， $M$  輸出的值組  $(\hat{x}, t_{i,h,m})$  若是在某個時段  $m < k_{\min}$

合法，假設此合法值組的合法期限可到時段  $d$ ，那麼依  $d$  的範圍可分為四種情況：

a. 若  $d < k_{\min}$ ：則值組  $(\hat{x}, t_{i,h,m})$  會有  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,d}} \equiv a_i^{\hat{x}} a_{0i} \equiv z^{(\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}}} \pmod{n_i}$

的關係，所以令  $\delta = \gcd(e_{i,m} \cdots e_{i,d}, (\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})$ ，再用延伸的歐基里德演

算法求出  $\alpha$ 、 $\beta$  滿足  $\delta = \alpha(e_{i,m} \cdots e_{i,d}) + \beta((\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})$ 。於是

$$z^\delta \equiv z^{\alpha(e_{i,m} \cdots e_{i,d})} \cdot z^{\beta((\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})} \equiv z^{\alpha(e_{i,m} \cdots e_{i,d})} \cdot (t_{i,h,m})^{\beta(e_{i,m} \cdots e_{i,d})} \equiv (z^\alpha (t_{i,h,m})^\beta)^{(e_{i,m} \cdots e_{i,d})} \pmod{n_i} ,$$

→  $z \equiv \left( z^\alpha (t_{i,h,m})^\beta \right)^{\frac{(e_{i,m} \cdots e_{i,d})}{\delta}} \pmod{n_i}$ ，所以值組  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,d})}{\delta} \right)$  為強

RSA 問題的解。

**b.** 若  $k_{\min} \leq d < d_{\max}$ ，值組  $(\hat{x}, t_{i,h,m})$  會有  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,d}} \equiv a_i^{\hat{x}} a_{0i} \equiv z^{(\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}}} \pmod{n_i}$

的關係，因此  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,k_{\min}-1}} \equiv z^{(\hat{x}+r)e_{i,d+1} \cdots e_{i,d_{\max}}} \pmod{n_i}$ 。所以令

$\delta = \gcd(e_{i,m} \cdots e_{i,k_{\min}-1}, (\hat{x}+r)e_{i,d+1} \cdots e_{i,d_{\max}})$ ，再用延伸的歐基里德演算法求出  $\alpha$ 、

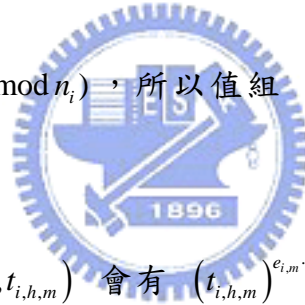
$\beta$  滿足  $\delta = \alpha(e_{i,m} \cdots e_{i,k_{\min}-1}) + \beta((\hat{x}+r)e_{i,d+1} \cdots e_{i,d_{\max}})$ 。於是

$$z^\delta \equiv z^{\alpha(e_{i,m} \cdots e_{i,k_{\min}-1})} \cdot z^{\beta((\hat{x}+r)e_{i,d+1} \cdots e_{i,d_{\max}})} \equiv z^{\alpha(e_{i,m} \cdots e_{i,k_{\min}-1})} \cdot (t_{i,h,m})^{\beta(e_{i,m} \cdots e_{i,k_{\min}-1})}$$

$$\equiv \left( z^\alpha (t_{i,h,m})^\beta \right)^{(e_{i,m} \cdots e_{i,k_{\min}-1})} \pmod{n_i}，$$

→  $z \equiv \left( z^\alpha (t_{i,h,m})^\beta \right)^{\frac{(e_{i,m} \cdots e_{i,k_{\min}-1})}{\delta}} \pmod{n_i}$ ，所以值組  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,k_{\min}-1})}{\delta} \right)$

為強 RSA 的解。



**c.** 若  $d_{\max} = d$ ，則值組  $(\hat{x}, t_{i,h,m})$  會有  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,d}} \equiv a_i^{\hat{x}} a_{0i} \equiv z^{(\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}}} \pmod{n_i}$

的關係，則  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,k_{\min}-1}} \equiv z^{(\hat{x}+r)} \pmod{n_i}$ 。所以令  $\delta = \gcd(e_{i,m} \cdots e_{i,k_{\min}-1}, (\hat{x}+r))$ ，

再用延伸的歐基里德演算法求出  $\alpha$ 、 $\beta$  滿足

$\delta = \alpha(e_{i,m} \cdots e_{i,k_{\min}-1}) + \beta((\hat{x}+r))$ 。於是

$$z^\delta \equiv z^{\alpha(e_{i,m} \cdots e_{i,k_{\min}-1})} \cdot z^{\beta(\hat{x}+r)} \equiv z^{\alpha(e_{i,m} \cdots e_{i,k_{\min}-1})} \cdot (t_{i,h,m})^{\beta(e_{i,m} \cdots e_{i,k_{\min}-1})} \equiv \left( z^\alpha (t_{i,h,m})^\beta \right)^{(e_{i,m} \cdots e_{i,k_{\min}-1})} \pmod{n_i}$$

→  $z \equiv \left( z^\alpha (t_{i,h,m})^\beta \right)^{\frac{(e_{i,m} \cdots e_{i,k_{\min}-1})}{\delta}} \pmod{n_i}$ ，所以值組  $(u, e) = \left( z^\alpha (t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,k_{\min}-1})}{\delta} \right)$

為強 RSA 的解。

**d.** 若  $d_{\max} < d$ ，則值組  $(\hat{x}, t_{i,h,m})$  會有  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,d}} \equiv a_i^{\hat{x}} a_{0i} \equiv z^{(\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}}} \pmod{n_i}$

的關係，則  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d}} \equiv z^{(\hat{x}+r)} \pmod{n_i}$ 。所以令

$\delta = \gcd(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d}, (\hat{x}+r))$ ，再用延伸的歐基里德演算法求出  $\alpha$ 、

$\beta$  滿足  $\delta = \alpha(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d}) + \beta((\hat{x}+r))$ 。於是

$$z^\delta \equiv z^{\alpha(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d})} \cdot z^{\beta(\hat{x}+r)} \equiv z^{\alpha(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d})} \cdot (t_{i,h,m})^{\beta(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d})}$$

$$\equiv (z^\alpha(t_{i,h,m})^\beta)^{(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d})} \pmod{n_i}$$

$$\rightarrow z \equiv (z^\alpha(t_{i,h,m})^\beta)^{\frac{(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d})}{\delta}} \pmod{n_i} \quad , \quad \text{所以值組}$$

$$(u, e) = \left( z^\alpha(t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,k_{\min}-1} \cdot e_{i,d_{\max}+1} \cdots e_{i,d})}{\delta} \right) \text{ 為強 RSA 的解。}$$

而在第 7 步驟的(2)中，則是若  $M$  輸出的值組  $(\hat{x}, t_{i,h,m})$  在時段  $m > d_{\max}$

為合法，則值組  $(\hat{x}, t_{i,h,m})$  會有  $(t_{i,h,m})^{e_{i,m} \cdots e_{i,d}} \equiv a_i^{\hat{x}} a_{0i} \equiv z^{(\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}}} \pmod{n_i}$  的關

係，所以令  $\delta = \gcd(e_{i,m} \cdots e_{i,d}, (\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})$ ，再用延伸的歐基里德演算法

求出  $\alpha$ 、 $\beta$  滿足  $\delta = \alpha(e_{i,m} \cdots e_{i,d}) + \beta((\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})$ 。於是

$$z^\delta \equiv z^{\alpha(e_{i,m} \cdots e_{i,d})} \cdot z^{\beta((\hat{x}+r)e_{i,k_{\min}} \cdots e_{i,d_{\max}})} \equiv z^{\alpha(e_{i,m} \cdots e_{i,d})} \cdot (t_{i,h,m})^{\beta(e_{i,m} \cdots e_{i,d})} \equiv (z^\alpha(t_{i,h,m})^\beta)^{(e_{i,m} \cdots e_{i,d})} \pmod{n_i} \quad ,$$

$$\rightarrow z \equiv (z^\alpha(t_{i,h,m})^\beta)^{\frac{(e_{i,m} \cdots e_{i,d})}{\delta}} \pmod{n_i} \quad , \quad \text{所以值組 } (u, e) = \left( z^\alpha(t_{i,h,m})^\beta, \frac{(e_{i,m} \cdots e_{i,d})}{\delta} \right) \text{ 為強}$$

RSA 的解。

由於強 RSA 問題是很難的，所以我們知道不存在一個  $M$  可以造出一

組合理值組  $(\hat{x}, t_{i,h,m})$ ，其中  $m < k_{\min}$  或  $m > d_{\max}$ 。

**定理3.** 我們在各群體中採用的群體簽章系統，在 Random oracle model、強 RSA 假設和延伸的 Diffie-Hellman[15]判決假設皆成立的情況下，滿足正確性、不可偽造性、匿名性、無連結性、脫罪性、可追蹤性、聯合抵

抗性、有追溯效力的公開廢止性與向後式無連結性。

**證明：**我們會引用[15]中的證明來依序證明改變為層級性架構後，仍然能滿足正確性、不可偽造性、匿名性、無連結性、脫罪性、可追蹤性、聯合抵抗性、有追溯效力的公開廢止性與向後式無連結性。

1. 正確性：若成員  $U_{i,j}$  在時段  $k$  為合法成員，則能利用參與和更新程序來得到時段  $k$  的憑證  $s_{i,j,k}$ ，再利用簽章程序來簽章。由定理 1 的完整性得知，這樣的簽章必能通過驗證程序。
2. 不可偽造性：由定理 1 可知我們的簽章具有統計性零知識的特性，再加上我們是在 Random oracle model 下，所以  $H$  這個抗碰撞雜湊函數的輸出的確是一個隨機的函數。所以無法由簽章中得到任何資訊，當然也包括群體簽章金鑰的資訊，因此群體簽章不會被偽造。
3. 匿名性：同樣可由此簽章系統的零知識特性來說明，除了群體管理者以外，皆無法由簽章中得到任何有關憑證和私密值的資訊，且是在 Random oracle model 下，不會洩漏任何資訊，因此可保證其匿名性。
4. 無連結性：同一群體的群體簽章需無法判斷是否為同一人所簽。假設有兩份同屬於群體  $G_i$  的正確簽章： $\sigma_1 = (c, i, j, e_j, s_1, s_2, T_1, T_2)$  與  $\sigma_2 = (\tilde{c}, i, k, e_k, \tilde{s}_1, \tilde{s}_2, \tilde{T}_1, \tilde{T}_2)$ ，若有一演算法  $A$  能判斷  $\sigma_1$  和  $\sigma_2$  是否由同一位成員所簽，則若 (1) 在  $j=k$  時， $A$  可以判斷  $\log_{y_i} \frac{T_1}{\tilde{T}_1} \pmod{n_i}$  和  $\log_{g_i} \frac{T_2}{\tilde{T}_2} \pmod{n_i}$  是否相等，若 (2) 在  $j < k$  時， $A$  可以判斷  $\log_{y_i} \frac{T_1^{e_{i,j}}}{\tilde{T}_1^{e_{i,k}}} \pmod{n_i}$  和  $\log_{g_i} \frac{T_2^{e_{i,j}}}{\tilde{T}_2^{e_{i,k}}} \pmod{n_i}$  是否相等。因此我

們可以設計一演算法利用  $A$  來解決延伸的 Diffie-Hellman 判決假設 [15]，此演算法詳細步驟可參閱[15]。

5. 脫罪性：可分為兩點討論：(1)在參與程序中，使用者與管理者的互動，都是以零知識互動式證明來實作，因此即使群體管理者也無法知道某成員  $U_{i,j}$  的私密值  $x_{i,j}$  [7]。(2)在簽章程序先用  $T_1$ 、 $T_2$  以 ElGamal 的方式加密憑證，再用知識簽章的方式證明知道  $x_{i,j}$  和時段  $k$  的憑證  $s_{i,j,k}$ ，因此由定理 1 可知只有同時知道  $x_{i,j}$  和  $s_{i,j,k}$  的人才能簽出這樣的知識簽章，所以群體管理者和其他成員皆無法以使用者  $U_{i,j}$  的身份簽出合法的知識簽章。
6. 聯合抵抗性：由定理 2 可知，此簽章系統滿足弱的聯合抵抗性，不管是根據定義的往前，或是經修改過的往後部分，皆可滿足。
7. 可追蹤性：由公開程序可知，由合法成員簽出的正確簽章，可被群體管理者經解章程序將憑證擷取出來，且由定理 2 得知，這樣的群體簽章系統可以保證一群惡意的合法成員，無法合作造出不具身份的成員用時段  $k_{\min}$  之前或  $d_{\max}$  之後的群體簽名金鑰所簽出的群體簽章。
8. 有追溯效力的公開廢止性：若使用者  $U_{i,j}$  在其合法時段  $k$  時，用其憑證  $s_{i,j,k}$  簽出簽章  $\sigma = (c, i, k, e_k, s_1, s_2, T_1, T_2)$ 。因為使用者  $U_{i,j}$  擁有私密值  $x_{i,j}$  與憑證  $s_{i,j,k} = (a_i^{x_{i,j}} a_{0i})^{\frac{1}{e_{i,k}}} \pmod{n_i}$ ，所以能通過驗證程序的 1~4 步驟，但因為此憑證若被廢止後，會被公佈在列表  $RL_{i,k}$  中，驗證程序剩下的步驟，事實上就是在比對  $s_{i,j,k}$  與列表上的憑證，若相同即會被檢查出來，此憑證即為不合法的憑證，驗證失敗。



9. 向後式無連結性：此群體架構滿足無連結性，這裡要說的是即使因憑證廢止列表的關係洩漏了一些憑證，仍然無法判斷某一簽章  $\sigma = (c, i, k, e_k, s_1, s_2, T_1, T_2)$  是否由某一遭廢止的憑證  $s_{i,j',k'}$  所簽，其中  $k < k'$ 。假設有一演算法可判別  $\sigma$  是否由某一被廢止的成員  $U_{i,j'}$  所簽出，則可以判別  $\log_{y_i} \frac{T_1^{e_{i,k}}}{(s_{i,j',k'})^{e_{i,k}}} \pmod{n_i}$  與  $\log_{g_i} T_2^{e_{i,k}} \pmod{n_i}$  是否相等，則可設計一演算法利用 A 來解決延伸的 Diffie-Hellman 判決假設 [15]，此演算法詳細步驟可參閱[15]。

## 第二節 其他安全性證明

**定理4.** 根據我們指定給各群體管理員 X 值的方式，一個群體中的 X 值，只有它的群體管理者和其祖先節點的群體管理者能推導出來。

**證明：** 我們先證明若非某群體的群體管理者和其祖先節點的群體管理者則無法聯合推導出此群體的 K 值。令  $G_i$  為一個群體，群體的集合  $C = \{G_j | G_j \ni G_i\}$  成員為一群按設定不該能推導出  $G_i$  中秘密值  $K_i$  的管理員之群體集合。將  $G_i$  所被指定的質數值記為  $P_i$ ，運算所得的 T 值記為  $T_i$ ，而  $G_j$  所被指定的質數記為  $P_j$ ，所持 T 值為  $T_j$ ，則我們證明的方式要先證明： $C$  這個群體的串通集合能推出  $G_i$  中的 K 值，若且唯若  $\text{g.c.d}\{T_j\}$  整除  $T_i$  (即  $C$  之中含有  $G_i$  的管理者或其祖先節點的群體管理者，因為若



不含，則  $\text{g.c.d}\{T_j\}$  不整除  $T_i$  )<sup>2</sup>。

( $\Leftarrow$ )若  $\text{g.c.d}\{T_j\}$  整除  $T_i$ ，令  $g = \text{g.c.d}\{T_j\}$  則我們可用歐幾里德演算法 (Euclid's algorithm) 來求出一些  $\alpha_j$  使得  $g = \sum_C \alpha_j T_j$ ，若  $T = gr$ ，則可以算出  $K = K_0^T = K_0^{gr} = \prod_C K_0^{\alpha_j T_j r} = \prod_C K_j^{\alpha_j r} \pmod{M}$ 。

( $\Rightarrow$ )我們用轉化 (Reduction) 的方式證明。若有一可計算的函式  $F$  能讓集合  $C$  中成員算出  $K_i$ ，則我們可利用  $F$  來解出目前被公認很難的  $\mathbb{Z}_n^*$  中的  $r$  次方根問題。假設  $F$  為一個可計算的函式使得  $K^T = F(K^{T_1}, K^{T_2}, \dots, K^{T_n}) \pmod{M}$  其中  $\text{g.c.d}\{T_j\}$  不整除  $T_i$ ，令

$r = \frac{\text{g.c.d}\{T_j\}}{\text{g.c.d}\{\{T_j\}, T_i\}}$ ，那麼對  $\mathbb{Z}_M^*$  中的任意元素  $H$ ，我們可用下方的演算法來

解出  $H$  的  $r$  次方根。

*Algorithm2*( $H, M, T_1, T_2, \dots, T_n$ )

1. 令  $d = \text{g.c.d}\{T_j\}$

2. 令  $e = \text{g.c.d}\{d, T_i\}$

3. 令  $r = \frac{d}{e}$

4. 對  $j=1$  到  $n$

    令  $r_j = \frac{T_j}{d}$

5. 使用歐基里德演算法來算出  $a$ 、 $b$  使得  $e = aT_i + bd$

---

<sup>2</sup>註：若  $C$  中的成員皆非  $G_i$  或  $G_i$  的祖先節點，對於所有  $T_j$ ， $P_i$  都會整除  $T_j$ ，因此  $P_i$  會整除所有  $T_j$  的最大公因數。另外  $P_i$  不整除  $T_i$ 。所以我們可以得到  $\text{g.c.d}\{T_j\}$  不整除  $T_i$ 。

6. 輸出  $H^b \cdot F(H^{r_1}, H^{r_2}, \dots, H^{r_n})^a$

這是因為若令  $K = H^{\frac{1}{d}}$ ，那麼  $H^{\frac{1}{r}} \equiv K^{\frac{d}{r}} \equiv K^e \equiv K^{aT_i+bd} \equiv F(K^{T_1}, K^{T_2}, \dots, K^{T_n})^a \cdot H^b \equiv H^b \cdot F(H^{r_1}, H^{r_2}, \dots, H^{r_n}) \pmod{M}$ 。考慮  $r$  的值，若  $r=1$ ，即  $d=e$ ，則  $\text{g.c.d}\{T_j\}$  整除  $T_i$  ( $\rightarrow \leftarrow$  與假設不符)。若  $r \neq 1$ ，則我們用此演算法解出了  $H$  的  $r$  次方根。但因為目前密碼學上一般相信解  $\mathbb{Z}_n^*$  中的  $r$  ( $r > 2$ ) 次方根(在模數  $M$  之下)的問題是很難的，其難度與分解  $M$  相當。因此這樣的函式  $F$  並不存在。

由以上雙向的證明可以得知：若非某群體的群體管理者和其祖先節點的群體管理者則無法串通聯合推導出此群體的  $K$  值。而在我們的架構中， $X$  值的選取是由  $K$  值透過一個抗碰撞雜湊函數來選取，不同的  $K$  值將對應至不同的  $X$  值，因此  $K$  值的安全性即可保障  $X$  值的安全性。

**定理5.** 若下層的成員兼為上層的管理者，本文架構之安全也不會受到影響。

**證明：**根據定理 2 的證明，擁有群體  $G_i$  中一些合理值組  $(x_{i,j}, t_{i,j,k})$  的集合並不能再造出新的合理值組來簽章。而若此群體  $G_i$  中的成員兼為上層的管理者，他對此群體增加的權限僅止於知道群體私密金鑰  $x_i$ ，使他能利用 ElGamal 的解密法，解開此群體的簽章，得到群體簽章中的憑證值。但無法得到該憑證的使用者私密值  $x_{i,j}$  值，所以無法由解開簽章所得的憑證值造出該使用者的簽章，其餘安全性依然滿足。

**定理6.** 一個加入群體時被指定在時間  $c$  到  $d$  之間為合法的成員，無法造出一個在時間  $[c, d]$  之外的合法簽章。

**證明：**因為此群體簽章具有完美性，所以若能在時間  $[c,d]$  之外造出合法簽章，即代表造出了在時間  $[c,d]$  之外的合法憑證  $S$  值。我們的證明採用轉化的方式：若存在一個多項式時間的入侵者  $A$  被允許與群體管理者  $GM_i$  進行參與程序加入群體  $G_i$ ，且完成參與程序後，被指定為在時間  $[c,d]$  間為合法的群體成員，並得到值組  $(x,t)$  為其私密金鑰與更新元件。若  $A$  能造出在時間  $f \notin [c,d]$  之外的合法簽章金鑰  $s$ ，則我們能利用  $A$  根據下面的演算法步驟來解強 RSA 的問題。

*Algorithm3*( $n,z$ )

1. 令  $a=z$ 、 $a_0=z$ ，選取質數  $e_{i,1}$ 、 $e_{i,2}$ 、 $\dots$ 、 $e_{i,T}$  滿足  $e_{i,k} \in \Gamma_k$ 。
2. 讓  $A$  跑參與程序獲得  $(x,t)$ ， $A$  輸出在時段  $f$  有效的  $s$  值，其中  $f \notin [c,d]$ 。
3. 令  $\delta = g.c.d(e_f, x+1)$ ，使用歐基里德演算法算出  $\alpha$ 、 $\beta$  使得  $\delta = \alpha \cdot e_f + \beta \cdot (x+1)$ 。
4. 輸出  $(u, e) = \left( z^\alpha s^\beta, \frac{e_f}{\delta} \right)$ 。

這是因為若  $s$  為在時段  $f$  的正確憑證值，則會滿足  $s^{e_f} = a^x a_0 = z^{x+1}$ ，所以我們可驗證上述的演算法輸出結果，
$$u^e \equiv \left( z^\alpha s^\beta \right)^{\frac{e_f}{\delta}} \equiv z^{\frac{\alpha e_f}{\delta}} \cdot s^{\frac{\beta e_f}{\delta}} \equiv z^{\frac{\alpha e_f}{\delta}} \cdot z^{\frac{\beta(x+1)}{\delta}} \equiv z^{\frac{\alpha e_f + \beta(x+1)}{\delta}} \equiv z \pmod{n}$$

因為強 RSA 問題是很難的，故可得知，這樣的  $A$  是不存在的。所以新增使用者時限的功能，的確不會讓使用者在合法時限後還能代替群體作一個合法的簽章。

# 第五章 結論

本章將討論本架構中可獨立出來之一個角色，即解章者 (Opener)，其功能只在於解開簽章，取出簽出此章的憑證。最後則是結論與未來研究方向。

## 第一節 解章者

我們所使用的階層性 (Hierarchy)，主要是著重在解開簽章的功能。因為在此種階層性架構中，通常層級的設定與成員的加入較易控制，層級為較大的架構，可在一開始即設定好。而成員的加入可指定需與欲加入群體之群體管理者執行參與程序。所以在後來的使用上，反而解開簽章會變成一個可以有階級觀念的常用功能。所以本篇論文中真正的階層性是指在解開簽章的功能中，擁有較上層解章功能的管理者，除了可解開所屬群體的簽章外，還可以解開其下方所轄群體的簽章。這樣的解章 (Open) 功能，我們可以把它獨立出來，成為一個解章者的身份。

其實我們群體簽章的匿名性，主要是因為我們用了 ElGamal 加密法將憑證給加密起來，因此若需要解開簽章的話，需要的金鑰便是 ElGamal 的解密金鑰，即我們架構中的各群體  $G_i$  的  $x_i$  值。所以如果有一個使用者，他擁有某一群體  $G_i$  的  $x_i$  值，就能擁有該群體解章者的功能。

在我們的架構中，如果我們要让解章者跟群體管理者一樣能解開其下方所轄階層的簽章的話，那就要再額外讓解章者知道自己所屬群體的  $K_i$  值。有了  $K_i$  值後，假設群體  $G_i$  是群體  $G_j$  可管轄的一個群體，那群體  $G_i$  的解章者要開啟群體  $G_j$  的簽章的話，因為  $T_i$  會整除  $T_j$ ，所以只需利用

$K_{i'} = K_i^{\frac{T_i}{T_i'}} \pmod{M}$ ，再用雜湊函數  $H'$  算出  $x_{i'} = H'(K_{i'}, n_{i'})$ ，即可得到用來解開群體  $G_{i'}$  簽章的私密金鑰  $x_{i'}$ 。

如果我們想要讓這樣的角色有時效性的話，像是能在某一段時間代理解開簽章的功能時，有兩個簡單的方法可能辦到：

(1) 例如在每個時段開始時，由系統管理者隨機選出一個  $r$  值，將原本的  $K_0$  值設為  $K_0^r$ 。再依層級初始程序的步驟 1.4 與群體設定程序的步驟 2.2、2.3 重新算出各群體中新的  $x_i$  值與  $y_i$  值。而有時效性的解章者在每個時段開始時須向群體管理者或系統管理者取得  $r$  值來更新所持的私密金鑰。所謂的時效性由群體管理者或系統管理者決定是否給此解章者  $r$  值，這是一個簡單的想法。稍加改進還可讓各群體管理者有[15]中的 seed 可推得以後的  $r$  值，只有解章者需要在每個時段跟管理者要  $r$  值。但用在層級性構造時，每次更新金鑰所需成本太大，且解章者每時段皆須與管理者連線，故較不可行。

(2) 此想法來自於[16]中的觀念，事實上也是本篇中用在有時限使用者的觀念。為免複雜化架構，故只略敘而不正規寫出，且不考慮中途撤銷的情形。因為我們主要是要將 ElGamal 加密中的金鑰對  $(x, y)$  滿足  $g^x = y$  中的  $x$  加上時限。所以為各時段皆取一  $\phi$  值，共有  $\phi_1, \dots, \phi_r$  並公開，每群體各自取一  $\alpha, \beta$ ，當經由群體設定的步驟 2.2 得到  $x_i$  後，先算出  $A = (\alpha^{x_i} \beta)^{\frac{1}{\phi_1 \cdots \phi_r}} \pmod{n}$  的值後，用  $x_{i,1} = A^{\phi_1 \cdots \phi_r} \pmod{n}$  與  $y_{i,1} = g^A \pmod{n}$  成為原本的  $x_i$  與  $y_i$  在時段 1 時的值，即第一個時段群體中 ElGamal 加密的金

鑰對 (key pair)。之後每次由時段  $k$  進入時段  $k+1$ ，則算出  $x_{i,k+1} = (x_{i,1})^{\phi_1 \cdots \phi_k \cdot \phi_{k+2} \cdots \phi_r} \pmod{n}$ 、 $y_{i,k+1} = g^{(x_{i,k+1})} \pmod{n}$ ，可如我們架構中的群體成員多記錄一個更新值  $t$ ，就可以省下許多計算。若有解章者須有時段  $c$  到  $d$  的解章權限，則給予一  $x = (x_{i,1})^{\phi_1 \cdots \phi_{c-1} \cdot \phi_{d+1} \cdots \phi_r} \pmod{n}$ ，這樣一來，雖然每時段還是得更新各群體中 ElGamal 加密的金鑰對，但解章者就不必每時段與群體管理者連線，只要作與管理者類似的更新，即可確保其時效性。

## 第二節 結論與未來研究方向

我們以[15]中擁有撤銷機制的前進式群體簽章為基礎，結合階層性的觀念，使得各群體有高低與主從之分，身處於比較高階級的管理者，除了可解開該群體的簽章外，還可解開該群體所轄之下方群體的簽章。這樣的方式是利用[1]中一個安全的階層性金鑰推導法來設定群體簽章中用 ElGamal 加密法來加密成員憑證時，ElGamal 的解密金鑰。因此可以達成上述的功能。另外對於原本[15]中的架構，因為使用了憑證廢止列表的方法，隨著廢止憑證數增加，各時間廢止列表之資料量也將增加，且越後面的時段，廢止列表的資料量將越多，因此我們更改了成員參與的部分，將其改變為可在成員加入時即設定其合法時限的作法，可在適用的時候使用，幫助減輕憑證廢止列表的資料量負擔。

未來的研究方向可以有幾個部分：(1)在本篇架構的基礎上，設法將簽章的部分導入層級性的觀念。例如身處較高階層的群體成員，可代替下方的群體簽章，但必須考慮效率。例如至少希望保持群體的公開金鑰和簽章長度皆與群體成員個數無關。(2)在本文這種類似多群體 (mtltigroups) 的架構下，一



個成員若同時屬於多個群體，因為並未考慮其推導方法，所以必須同時持有  
多組簽章金鑰，這是一個應該避免的結果，至少希望所持有的簽章金鑰不應  
該與所加入的群體個數呈線性 (linear) 成長關係。(3)在改進的架構中，解章  
者是否能更有效率的指定其合法時限，成為一個可方便指定代理期間的功  
能。這些雖然不是本文所要探討的重點，但都可以作為日後改善的方向。



## 參考文獻

- [1] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," In *ACM Transactions on Computer Systems*, vol. 1, no.3, pp. 239-248 , 1983.
- [2] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," In *Proceedings of Advances in Cryptology - Crypto 2000, LNCS 1880*, pp. 255-270, Springer-Verlag, 2000.
- [3] G. Ateniese and G. Tsudik, "Some open issues and new directions in group signature," In *Proceedings of Financial Cryptography 1999, LNCS 1648*, pp. 196-211, Springer-Verlag, 1999.
- [4] N. Baric and B. Pfitzman, "Collision-free accumulators and fail-stop signature schemes without trees," In *Proceedings of Advances in Cryptology - Eurocrypt 1997, LNCS 1233*, pp. 480-494, Springer-Verlag, 1997.
- [5] M. Bellare and S. K. Miner, "A forward-secure digital signature scheme," In *Proceedings of Advances in Cryptology - Crypto 1999, LNCS 1666*, pp. 431-448, Springer-Verlag, 1999.
- [6] J. Camenisch, "Efficient and generalized group signatures," In *Proceedings of Advances in Cryptology - Eurocrypt 1997, LNCS 1233*, pp. 465-479, Springer-Verlag, 1997.
- [7] J. Camenisch and M. Michels, "A group signature scheme based on an RSA-variant," Technical Report RS-98-27, BRICS, Departement of Computer Science, University of Aarhus, 1998. Preliminary version : [8]



- [8] J. Camenisch and M. Michels, "A group signature scheme with improved efficiency," In *Proceedings of Advances in Cryptology – Asiacrypt 1998, LNCS 1514*, pp. 160-174, Springer-Verlag, 1998.
- [9] J. Camenisch and M. Michels, "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation," In *Proceedings of Advances in Cryptology - Eurocrypt 2001, LNCS 2045*, pp. 93-118, Springer-Verlag, 2001.
- [10] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," In *Proceedings of Advances in Cryptology - Crypto 1997, LNCS 1296*, pp. 410-424, Springer-Verlag, 1997.
- [11] S. Canard and J. Traoré, "On Fair E-cash Systems Based on Group Signature Schemes," In *Proceedings of Information Security and Privacy (ACISP'03), LNCS 2727*, pp. 237-248, Springer-Verlag, 2003.
- [12] D. Chaum and E. van Heyst, "Group signatures," In *Proceedings of Advances in Cryptology - Eurocrypt 1991, LNCS 547*, pp. 257-265, Springer-Verlag, 1991.
- [13] L. Chen and T. Pedersen, "New group signature schemes," In *Proceedings of Advances in Cryptology - Eurocrypt 1994, LNCS 950*, pp. 171-181, Springer-Verlag, 1995.
- [14] E. Fujisaki and T. Okamoto, "Statistical zero-knowledge protocols to prove modular polynomial relations," In *Proceedings of Advances Cryptology - Crypto 1997, LNCS 1294*, pp. 16-30, Springer-Verlag, 1997.
- [15] C. H. Huang and W. G. Tzeng, "A Forward Group Signature Scheme with

- Revocation Mechanism,” National Chiao-Tung University, Master Thesis, 2002.
- [16]G. Itkis and L. Reyzin, “Forward-secure signatures with optimal signing and verifying,” In *Proceedings of Advances in Cryptology - Crypto 2001*, LNCS 2139, pp. 332-354, Springer-Verlag, 2001.
- [17]H. J. Kim, J. I. Lim and D. H. Lee, “Efficient and secure member deletion in group signature schemes,” In *Proceedings of the Third International Conference on Information Security and Cryptology*, LNCS 2015, pp. 150-161, Springer-Verlag, 2001.
- [18]S. Kim, S. Park and D. Won, “Group signatures for Hierarchical Multigroups,” In *Proceedings of the Information Security Workshop (ISW'97)*, LNCS 1396, pp. 273-281, Springer-Verlag, 1998.
- [19]A. Lysyanskaya and Z. Ramzan, “Group blind digital signatures: A scalable solution to electronic cash,” In *Proceedings of Financial Cryptography 1998*, LNCS 1465, pp. 184-197, Springer-Verlag, 1998.
- [20]G. Maitland and C. Boyd, “Fair Electronic Cash Based on a Group Signature Scheme,” In *Proceedings of Information and Communications Security (ICICS'01)*, LNCS 2229, pp.461-465, Springer-Verlag, 2001.
- [21]H. Petersen, “How to convert any digital signature scheme into a group signature scheme,” In *Proceedings of the 5th International Workshop on Security Protocols*, LNCS 1361, pp.177-190, Springer-Verlag, 1998.
- [22]K. Sakurai and S. Miyazaki, “An anonymous electronic bidding protocol based on a new convertible group signature scheme,” In *Proceedings of*

*Information Security and Privacy (ACISP'00)*, LNCS 1841, pp.385-399, Springer-Verlag, 2000.

[23]D. Song, “Practical forward-secure group signature schemes,” In *Proceedings of the Eighth ACM Symposium on Computer and Communication Security (CCS2001)*, pp.225-234, 2001.

[24]G. Wang, “On the Security of a Group Signature Scheme with Forward Security,” In *Cryptology ePrint Archive*, <http://eprint.iacr.org/2003/226/>

[25]M. Zhang, “New Approaches to Password Authenticated Key Exchange based on RSA,” In *Cryptology ePrint Archive*, <http://eprint.iacr.org/2004/033/>

[26]J. Zhang, Q. Wu and Y. Wang “A novel efficient group signature scheme with forward security,” In *Proceedings of Information and Communications Security (ICICS'03)*, LNCS 2836, pp.292-300, Springer-Verlag, 2003.

