簡單快速雲端階層式組織授權之應用

李鴻璋
淡江大學資訊管理學系
johnez.lee@gmail.com

蔡佳勳
淡江大學資訊管理學系
coolwcolcc@gmail.com

摘要

本論文提出一套應用在雲端運算中，使階層式結構中群組間進行資料授權能夠簡單快速方法。本方法中，結構中群組有一把公開金鑰 PK 及私密金鑰 SK，並且將群組私密金鑰 SK，用直接上屬群組公開金鑰加密產生公開參數 R。利用直屬結構表公開各群組公開金鑰，相對公開參數 R 及直接上屬群組等資訊。

群組會將資料用群組私密金鑰 SK 所導出的加密金鑰，加密該文件，並將其上傳至雲端中。當被授權的群組(即上屬群組們)，則透過對直屬結構表中公開數值，遞迴路徑運算出該群組私密金鑰 SK 進而解密該資料。本論文所提機制亦與 AKL、Lo-Hwang-Liu、Chu-Hsing Lin 三位所提機制在多個面象(運作成員、效率、新成員加入及離開等)做比較，本論文具不用 CA(Certificate Authority，憑證管理中心)、運算簡單、具當階層式結構擴大較少公開參數改變的優點。

關鍵詞：階層式、公開金鑰、雲端運算

A simple and fast cloud authorization for hierarchical organization

Hung-Chang Lee
Department of Information Management, Tamkang University
johnez.lee@gmail.com

Chia-Hsun Tsai
Department of Information Management, Tamkang University
coolwcolcc@gmail.com

Abstract

This study proposes a simple and fast data authorization in hierarchical structure between groups for cloud computing. Within this method, each group in hierarchical organization gets a pair key, naming the public key ’PK’and the private key ‘SK’. The group use its direct ancestor groups’ public keys ’PK’ to encrypt its own private key ’SK’ to generate the open parameters R. All these parameters(public keys PK, open parameters R) and their relationship are to open public by using an open table called RAP (Relation-And-Parameter) table.

When comes to data encryption, groups derive the encryption key from its private key ‘SK’ by using a open function called F function, encrypt the files and upload them to the cloud. When the group is authorized (groups that are the ancestor groups’), they look the RAP table and find the path between his group and the authorized group, recursively computed the group’s private key ‘SK’ in the path, and finally use the F function to derive the decrypt encryption key. We also compared the proposed mechanisms with those by AKL, Lo-Hwang-Liu, and Chu-Hsing Lin in multiple faces like operation of member, efficiency, new members join and leave, etc.. As a result, the proposed one provides little CA (Certificate Authority), simple operation, fewer public parameters changes when come to a hierarchical structure expansion.

Keywords: hierarchical, public key, cloud computing

簡單快速雲端階層式組織授權之應用

壹、導論

一、研究背景

雲端運算(Cloud Computing)是近年來非常熱門的話題，個人及組織也紛紛導入使用，將資料上傳至雲端運算平台之中，除降低實體設備的需求外，也能滿足隨時、隨處資訊服務的需要。只是現今的雲端運算平台服務考量重點停留在便利性、傳輸速度及儲存空間的大小，對服務應用的需求如組織內部結構授權與資訊權責分級等的設計大多缺乏。

階層式(Hierarchy)架構需要在雲端運算服務中應用，原因在於階層式架構在現今組織中被廣泛使用，近年來企業、政府與軍隊中，會根據自己在階層式架構中的位置進行授權(MacKinnon et al,1985;Denning et al,1987)。例如一個組織中總經理往往是擁有最高權限的人物，研發、會計、業務等部門則依序往下分階層。並且在階層式架構中的組織存取權限(access control)中，位於上級的使用者，能夠使用其下屬之權限；反過來講位於較低階級的使用者不能去使用其上級之權限(陳正鎔，2013)。

在資訊技術發展快速的時代中，幾乎所有的組織都有導入階層式架構之應用需要。想要雲端運算進一步的發展，必需讓組織能夠將階層式資料文件授權概念放置於雲端平台，來滿足資訊權責分級的安全性需求。相信不久將來雲端資訊的權責分級化在未來發展必然成為主流，所以階層式架構下的組織如何透過雲端運算的便利性及找到有效的方式來保護雲端平台上資料安全避免被非法使用是未來不能忽視的重要課題。

二、研究動機與目的

在密碼學的領域之中，能夠採用階層式金鑰分配機制(Hierarchical Key assignment mechanism)來達成資訊權責分級，其使用方式為各自擁有的金鑰去進行加密，而有能力解密的使用者必定為自己或者是直接上級的使用者能夠推導出來(Chen,2004)。

在階層式架構中使用者加入一段時間後，有可能會有成員的加入及成員的離開或者是到其他部門時，這時候原始的階層式金鑰分配機制，就必須將舊金鑰銷毀來避免之前的使用者惡意盜用組織內部資料。而另一方面管理者必須創造出新金鑰交給新進使用者及其他相關使用者，這樣反而會讓組織內部金鑰長期做更換，導致金鑰分配(Key Distribution)的問題。

在 Akl & Taylor(1983)兩人所提出利用質數的運算來做階層式金鑰的加密推導，此方法是階層式金鑰推導的原身，但是質數的分配導致儲存空間會因為階層數量而需要倍數的成長及結構上限制過大，在組織中成員加入及離開時每次都需要 CA 進行操作，無法很好的應用於組織中。

Lo, Hwang & Liu (2011)等人所提出之方法改良了 Akl & Taylor(1983)，讓他們能夠在階層式架構中先進行位子上的判別。中間節點、末端節點以及共同上級等來分配質數的數量，但是結構上仍然有限制並且會有群組聯合攻擊的可能性以至於金鑰的安全性更加危險，另外在組織中成員加入及離開也需要 CA 進行運作。

Lin(1997)所提出方法，雖然在階層式架構下的結構上沒有任何的限制，整體運作流程則是完全的交給 CA 進行處理，雖然有 CA 運作是很便利但是這也是他最大的問題，如果 CA 被非法入侵或遭受到惡意攻擊時，Lin(1997)所提出方法就無法繼續使用，而在組織當中可能會因為這些原因產生更加劇大的損失。

本論文提出一套簡單快速在雲端運算中階層式授權之應用。其方法為讓組織內部群

組各自擁有一組公開金鑰 PK 及一組私密金鑰 SK，而階層較低的群組能用直接上屬群組的公開金鑰 PK 加密後產生公開參數 R，並且使用直屬結構表將各群組的公開金鑰 PK 及相對應的公開參數 R 等資訊公佈給所有群組知道。群組使用者就能使用相對應的公開金鑰 PK 將檔案資料使用群組私密金鑰 SK 進行加密後上傳至雲端中，上級群組使用者在雲端看到之後則可以將檔案資料下載下來，並使用自己群組所擁有的私密金鑰 SK 進行解密，這樣能使資料保有隱密性，且利用這套加密機制能得到身分確認以至於在雲端傳輸中更加安全。

本論文提出之方法與 Akl & Taylor(1983)、Lo,Hwang & Liu (2011)、Chu-Hsing Lin(1997)三位所提機制在多個多個面向(運作成員、效率、新成員加入及離開等)做比較，本論文具不用 CA(Certificate Authority，憑證管理中心)、運算簡單、具當階層式結構擴大只需較少公開參數改變的優點。

貳、相關文獻探討

本章節將介紹本論文中相關的文獻探討，包含了質數運算做階層式金鑰推導、使用公開金鑰系統、雲端運算以及相關研究之階層式金鑰運作方法等。

一、使用質數運算做階層式金鑰推論

1983 年由 Akl & Taylor(1983)兩人所提出利用質數的運算來做階層式金鑰的推導，其運作的方式要在階層式群組中由 CA 先制定出母金鑰 K 與兩大質數 p、q 另 M=pq，且每個群組自己都會有一個質數 p_i，然後每個群組會有一個公開參數其公開參數為所有上級的質數連乘積，利用公開參數進行 mod 運算能夠得出金鑰，這個方法雖然能夠快速推論出金鑰，但是如果公開參數的變動以及群組使用者加入和離開，都要從新計算所有公開參數和金鑰。在 1985 年 MacKinnon et al.(1985)所提出了一種改進的方案稱為「規模分配方案」，這是減少了一個用戶群組階層，雖然此方法可以降低質數的數量，但能需要大量的儲存空間才能使用此方法。Hwang & Yang (2003)提出對於那些大部分有序的階層式結構訪問方案，其方法為基於與組合數學概念改良 Akl & Taylor(1983)的方法，但是在 Wang & Laih(2005)所提出的計畫，是以減少選擇質數的數量並且證明 Hwang & Yang (2003)方法是無法對抗共同攻擊的(Wang & Laih,2005)。

二、公開金鑰系統

公開金鑰系統是由 Diffie & Hellman(1976)所提出的 D-H 金鑰交換演算法以及 Rivest , Shamir & Adleman(1978)所提出的 RSA 非對稱金鑰演算法，公開金鑰基礎建設的目的在於電腦使用者可以在非保護的通信中建立共享金鑰的方法以提出雙方認證，並使用公鑰憑證內的公鑰資訊加密給對方。解密時，每個使用者使用自己的私密金鑰進行解密。這樣的機制能夠達成機密性、訊息完整性以及使用者的認證。

(一)、 RSA 公開金鑰密碼系統

RSA 公開金鑰密碼系統是在 1978 年由美國麻省理工學院的三位教授 Rivest et al.(1978)所提出的 RSA 非對稱金鑰演算法是一份基於因數分解困難度(factorization，FAC)以作為基礎的一套公開金鑰密碼系統，可用於加解密，以下我們將介紹產生金鑰對以及加解密之步驟。

一、產生金鑰對

1. 假設有 *A* 與 *B* 兩個人，其中 A 隨機產生兩個大質數 *p*、*q* 兩數，並計算 *n*=*p***q*。
2. *A* 隨機找出一個滿足 gcd(*e*,ϕ(n))=1 之整數 e_i，這邊的ϕ(n)表式為尤拉商數，並且要符合比 n 小且與 n 互質之整數個數，當 n=*p***q* 時，ϕ(n)之值為 (*p*-1)(*q*-1)，其中（e_i，n）為公開金鑰並讓 *B* 知道。
3. A 計算私密金鑰 d_i，d_i 滿足 $e_i\, d_i \equiv 1 (\mathrm{mod}\ \phi(n))$。

二、加解密

金鑰對產生後，*A* 將公開金鑰（e_i,n）傳送給 *B*，*A* 則保有私密金鑰 d_i，則 *B* 可以利用 *A* 的公開金鑰（e_i,n）執行加密的動作 $E(e_i, m) = C = m^{ei} \mod n = m$，將明文 m 加密後傳送給 A，則 A 利用私密金鑰 d_i 作解密的動作 $D_{di}(m) = C^{di} = (m^{ei})^{di} \mod m = m$ 。

(二)、 ElGamal 公開金鑰密碼系統

Elgamal 公開金鑰密碼系統在 1985 由 ElGamal, T.(1985)所提出來的其方式為建立在 Diffie & Hellman(1976)所提的系統上，基於離散對數問題(discrete logarithm problem，DLP)作為基礎的一套公開金鑰密碼系統，可用於加解密，運算方法如以下之步驟：

密鑰產生：

首先選擇一個質數 p，兩個隨機數 g 和 x，g 和 x 要小於 p，計算 $y=g^x \mod p$，公鑰為 y、g 及 p，私鑰為 x。

加密：

對要加密之明文 m，且 1<m<p-1，隨機產生亂數 k 且 gcd(g,p-1)=1，g 和 p-1 必須互質，並計算 $a=g^k \mod p$ 及 $b=y^k m \mod p$ 後，則(a,b)為對明文 m 加密之密鑰。

三、雲端運算

雲端運算是一種基於網路的運算方式，透過這種方式共享軟體資源和資訊可以按需求提供給其他裝置(Jaeger & Schiffman,2010)。其中包含四大佈署模型、三大服務模式：

1. 公有雲(Public Cloud)：是將雲端廠商將雲端運算服務公開給使用者的模式，對於使用者的限制是最小的，也因為這個原因安全性也是最低的。
2. 私有雲(Private Cloud)：一般由自己或者是企業內自行使用的雲端環境，而私有雲的資料可以自行管理或者交給信任的第三方管理，因為這個原因只有企業內部或者是自己能使用，所以安全性較公有雲高。
3. 混合雲(Hybrid Cloud)：作為私有雲及公有雲的媒介，使用者通常將非企業重要資訊外包並在公有雲上使用，作為企業及企業之間的互通資源。
4. 社群雲(Community Cloud)：社群雲是由眾多利益相同的組織掌握使用。例如：特定宗教或特定安全上使用，其社群雲內部資料由成員共享。

三種服務模式：

1. 基礎設施即服務(Infrastructure as a service, IaaS)：IaaS 是所有雲端服務的基礎，主要在於提供使用者的處理、儲存、網路及各種基礎運算資源，讓使用者不用購買伺服器回來，而在網路上建置作業系統及使用應用軟體。
2. 平台即服務(Platform as a service, PaaS)：PaaS 是提供運算平台及解決方案堆疊的服務，主要是將研發軟體的平台做為一種服務，例如 Facebook、Plurk、Google App Engine(GAE)等，都是屬於 PaaS 的服務，整體來說 PaaS 是 SaaS 的一種應用。
3. 軟體即服務(Software as a service, SaaS)：有人稱 SaaS 為”即需即用軟體”是一種軟體交付模式，在這種交付模式中雲端集中式託管軟體及相關數據，僅需要透過網路而不用安裝軟體，例如：雲端防毒軟體、會計系統、客戶關係管理系統等，使用者通常透過客戶端經由一個網頁瀏覽器來使用軟體及服務。

四、相關研究

(一)、 2.4.1Akl & Taylor 所提機制

表 1：Akl & Taylor 所提機制-符號定義

符號	定義
u_i	為群組使用者。
t_i	群組使用者的公開參數。
K_0	為 CA 系統隨機產生出的母金鑰。
p 及 q	CA 系統選擇出的兩大質數。
M	M 為系統選出兩大質數 p 及 q 相乘，M=pq。
f_0	f_0函數是某種固定的運算函數。
E	E 為公開金鑰加密方法 $C = E_{K^e(m)}$
D	D 為公開金鑰解密方法 $m = D_{K^d(C)}$
P_j	P_j 是由 CA 分配給每個群組的質數。
CA	一個受信任的角色，負責產生及維護所需的參數。

1983 年由 Akl & Taylor(1983)兩人所提出利用質數的運算來做階層式金鑰的加密推導，首先必須知道群組階層之間的關係以及分配整數給所有群組，例如 $u_j \geq u_i$ 則分配整數 t_j 及 t_i 並且 $t_i = zt_j$。

另外我們定義 $f_{(m)}$是一種單向的函數，如 $f_{(mz)} = fm$。fz ，m 跟 z 為隨機的整數，然後由 CA 隨機選擇產生母金鑰 K_0 並作下列運算

$$K_i = f_{ti}(K_0)$$

$$K_i = f_{ti}(K_0) = f_z \circ f_{tj}(K_0) = f_z(K_j)$$

知道能夠由 K_i 推導出 K_j 後，CA 選擇兩大質數 p、q，且令 M=pq 另外 M 為公開值，並給予每個使用者(群組)一個質數 p_i，

$$f_m(K) = K_m \pmod M$$

而經由上述金鑰推導後，群組 u_j 為 u_i 之上級，可利用下列式子以 K_j 導出金鑰 K_i。

$K_i = K_0^{ti} = [K_0^{tj}]^{ti/tj} = [K_j]^{ti/tj} \pmod M$

並且上級群組用者不會分配到自己以及下級群組的質數。

$$t_i = \prod_{U_j \leq U_i} P_j$$

在 AKL 階層式金鑰分配當中，階層群組不會擁有自己及下級的金鑰，使用此方法群組上級能夠直接推導出群組下級的金鑰的方式，推導金鑰過程快速、簡單，但是如果群組使用者增加或者群組使用者離開時，就必須要重新計算所有群組的金鑰與公開參數以及 CA 需要進行運作處理，相當的浪費時間及人力。

(二)、 2.4.2Lo, Hwang & Liu 等人所提機制

表 2：Lo, Hwang & Liu 等人所提機制-符號定義

符號	定義
C_i	群組使用者。
CA	一個受信任的角色，負責產生及維護所需的參數。
p 及 q	CA 系統選擇出的兩大質數。
m	m 為 CA 系統選出兩大質數 p 及 q 相乘，m=pq。
K_0	為系統隨機產生出的母金鑰。$2<K_0<(m-1)$

(e_i，d_i)	e_i 由 CA 系統選擇的一個質數並和 d_i 作乘法反原數的運算，$d_i = e_i^{-1} \bmod \phi(m)$，並將($e_i$，$d_i$)分配給群組 C_i。
H	H 是一個單向的雜湊函數運算。
⊕	⊕是做一個 exclusive or 的計算。
PB_i	PB_i(public parameters)是由 CA 公開發佈的公開參數，要讓所有群組能夠知道之間的關係。
K_i	CA 將 K_i(secret key)透過安全通道傳遞給每一個群組使用者。
中間節點	中間節點是在階層式群組當中，群組屬於擁有上級群組以及下級群組的節點。
共同上級	共同上級是在階層式群組當中，群組同時擁有兩個或兩個以上的上級群組。
末端節點	末端節點是在階層群組當中，群組只會有一個上級群組，並且不會有下級群組的情況下。

2011 年由 Lo,Hwang & Liu (2011)等人所提出，其方法是改良了 1983 年由 Akl & Taylor(1983)兩人所提出利用質數的運算來做階層式金鑰的加密推論，其金鑰產生方式以及金鑰推導方式如下：

1. 金鑰的產生

(1) 由 CA 選擇兩個大質數 p 和 q，然後計算公開參數 m=p*q，但是 p 和 q 兩數不得公開。

(2) CA 隨機產生亂數 K_0，K_0 必須是和 m 互質的質數，且要符合 $2<K_0<(m-1)$。

(3) 在階層式結構中，群組是屬於中間節點或者是擁有兩個共同上級的情況下，CA 會選擇質數 e_i 並且計算在 mod m 情況下的乘法反原數 d_i，並將(e_i，d_i)分配給群組 C_i。

$$d_i = e_i^{-1} \bmod \emptyset(m)$$

(4) 在階層式結構中，群組是屬於末端節點的情況時，CA 隨機產生私密金鑰 K_i，並且計算公開參數 PB_i

$$PB_i = K_i \oplus H(K_j, C_i, C_j)$$

其中 H 是一個單向雜湊函數的計算，C_j 是 C_i 的直接上級。

假設 C_K 是群組中的中間節點或是擁有共同上級的群組，CA 必須計算私密金鑰 K_k

$$K_k = K_0^{\left(d_k \cdot \Pi_{C_t} d_t \bmod \emptyset(m)\right)}$$

及公開參數 PB_k

$$PB_k = e_k \cdot \prod_{C_t} e^t$$

而(e_t，d_t)是由 CA 所給予的，C_t 是 C_K 的下級群組且 C_t 是必須是沒有下級群組。

(5) 最後 CA 將 K_i 透過安全的管道傳送給每個群組 C_i 並且公佈所有公開參數的關係。

2. 金鑰的推導

假設階層式群組當中，私密金鑰 K_i 推導可分為以下三種方式：

(1) 群組中只擁有一個上級如 $C_i < C_j$，私密金鑰 K_i 公式如下：

$$K_i = PB_i \oplus H(K_j, C_i, C_j)$$

(2) 群組之間的關係如 $C_i < C_k < C_j$，則私密金鑰 K_i 推導方式如下：

$$K_i = PB_i \oplus H(K_j^{PB_j/PB_k}, C_i, C_j)$$

(3) 群組同時擁有兩個上級的情況，私密金鑰 K_i 推導方式如下：

$$K_j^{(PB_j/PB_i)} \bmod m$$

群組 C_j 在不知道群組 C_i 私密金鑰 K_i 的情況下，群組 C_j 想推導出私密金鑰 K_i 時可透過上述的三種方式進行推導，利用知道公開參數 PB_i 及 PB_j 就能快速推導出 C_i 的私密金鑰 K_i 了。Lo,Hwang & Liu (2011)等人改良了 Akl & Taylor(1983)兩人所提機制當中最大的特點就是先判斷組織中的位子使屬於中間節點、共同上級和末端節點並且先分配質數然後在進行金鑰的推導，但是缺點的結構有限制沒有進行改善以及有機會採用群組聯合攻擊得到上級的金鑰，並且加入成員及成員離開時一樣也需要 CA 進行改良運作。

(三)、 2.4.3 Chu-Hsinh Lin 所提機制

表 3：Chu-Hsinh Lin 所提機制-符號定義

符號	定義
S	所有群組所形成的集合
S_i	集合 S 中的元素，代表每個群組。$1 < i < n$。
P	為 CA 隨機產生的一個大質數。
ID_i	群組 S_i 的識別碼。
K_i	群組 S_i 所擁有的群組金鑰。
r_{ji}	群組 S_i 與群組 S_j 間的公開參數，其中 i<j
CA	一個受信任的角色，負責產生及維護所需的參數。
⊕	⊕是做一個 exclusive or 的計算。
$E_{key}(X)$	以金鑰 key 將資料 X 做加密。
$D_{key}(X)$	以金鑰 key 將資料 X 做解密。

在 1997 年 Lin(1997)時提出了以離散對數為基礎的動態金鑰管理架構，其架構中使用了 RSA 作資料保密的演算法，此架構分三個階段說明如下。

產生金鑰階段：

CA 先隨機產生一個大質數 P 和原跟$Z \epsilon GF(P)$，且這兩個數會公開，接著 CA 依照 RSA 加密機制選擇一把公開金鑰 PK 和一把對應的私密金鑰 SK，然後將公開金鑰公佈給群組所有人使用。

每個群組 S_i 選擇一把群組金鑰 K_i，然後使用 CA 公佈的公開金鑰 PK 加密成 $E_{PK}(K_i)$ 傳送給 CA。CA 利用私密金鑰 SK 解密 $D_{PK}(E_{PK}(K_i))$得出群組金鑰 K_i 並根據群組關係 $S_j > S_i$，則計算公開參數 r_{ji}。

$$r_{ji}=(Z^{kj \oplus IDi} \bmod P) \oplus K_i$$

推導群組金鑰階段：

當有兩個群組關係為 $S_j > S_i$，則群組 S_j 跟 CA 要求參數 r_{ji}，CA 則先判斷此次要求是否安全，若 CA 判斷安全則將參數 r_{ji} 傳送給群組 S_j，S_j 執行下列式子則可推導出群組 S_i 的金鑰 K_i。

$$K_i=(Z^{kj \oplus IDi} \bmod P) \oplus r_{ji}$$

更新群組金鑰階段：

先判斷群組之間的關係如果群組關係為$(S_j > S_i)$要跟新金鑰，則 S_i 將新的金鑰 K_i^*用公開金鑰 PK 進行加密 $E_{PK}(K_i^*)$傳送給 CA，CA 收到資料後利用私密金鑰 SK 進行解密 $D_{PK}(E_{PK}(K_i^*))$得出 K_i^*並根據群組關係，更新公開參數 r^*_{ji}。如果 S_i 也有下級群組也需要重新再計算。

$$r^*_{ji}=(Z^{kj \oplus IDi} \bmod P) \oplus K_i^*$$

Lin(1997)所提機制中，幾乎所有的運算處理都是由 CA 進行操作以及結構上不像 Akl & Taylor(1983)兩人及 Lo,Hwang & Liu (2011)等人所提機制中有限制階層式的結構

不能夠採用單一垂直結構的情況下，比較能彈性應付群組的變化。最大的問題則是 CA 被非法入侵或遭受到惡意竄改攻擊的情況，Lin(1997)所提出方法就無法繼續使用，而在組織當中可能會因為這些原因產生更加劇大的損失。

參、所提機制

此章節將介紹本論文所提出在雲端運算中使階層式結構中群組之間運用金鑰加密進行資料授權能夠簡單快速的方法。

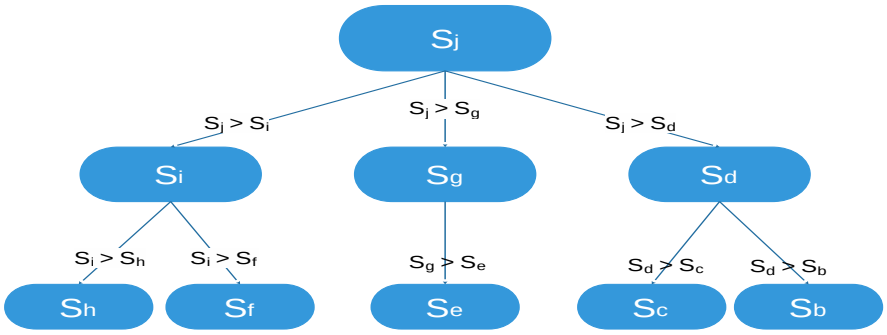
一、簡介

在組織內部中階層式管理已經是眾所皆知的，由於這個原因擁有較高權限的管理者需要使用其下屬資料授權的權限，屬於階層較低的使用者則無法使用上級的權限，組織因為在這個情況之下傳送檔案資料時都必須先進行加密的處理。當進行加解密處理的時候，金鑰的產生、擁有、管理以及分配就成為一個重大考量的因素。

為了解決這些問題，本論文採用密碼學中的傳統式加密系統和公開金鑰密碼系統的技術就能夠保護檔案資料的安全，本論文所提出方法將套用於階層式群組當中，結合密碼學、階層式架構.....等技術，以確保群組使用者在組織內部交換檔案資料的安全性不被非法破解、盜取、利用。本論文並不需要有 CA、階層式結構擴大較少公開參數改變、適用所有非對稱演算法、金鑰推導方式運算較簡單、新成員加入時原先群組影響較小及成員離開時群組改變較少等以上這些特點，能夠有效改變原本階層式架構下加解密的方式與效率。

二、研究架構

本論文藉由將組織先進行分級，並劃出組織的階層架構圖，讓使用者透過這張組織階層架構圖能夠輕易知道自己的所屬的階層以及和群組之間的直接直屬關係和隔代質屬關係。



Sj和Si及Si和Sh擁有直接直屬關係，但是Sj和Sh的關係為隔代直屬關係。

圖 1：組織階層架構圖

三、系統流程

我們將此系統分成六階段，分別為 1.基本運作 2.系統啟始設定 3.公開參數的產生 4.下級加密金鑰推導程序 5.加入新成員 6.成員離開時，符號定義表如下

表 4：本論文所提機制-符號定義

符號	定義
S_i	S_i 表示識別號為 i，所定義出的群組成員的集合。 $1 < i < n$。
>	表示直屬關係，如 $S_j > S_i$ 表示群組 S_j 為 S_i 直接上級，若 $S_j > S_i > S_k$ 表示 S_j 為 S_i 的直接上級 S_i 為 S_k 的直接上級，S_j 為 S_k 隔代上級。
>>	表是隔代直屬，如 $S_j > S_i \ldots > S_k$ 而群組 $S_j >> S_k$ 表示 S_j 為 S_k 的隔代上級，反過來

符號	定義
	S_k 為 S_j 的隔代下級。
PK_i，SK_i	群組 S_i 中的公開金鑰 PK_i 及私密金鑰 SK_i，其中公開金鑰 PK_i 是系統公開公佈參數。
E(K,M)	E 為傳統對稱式加密(Encryption)，如 AES 等 第一個參數 K 為金鑰，第二個參數 M 為訊息
D(K，M)	D 傳統對稱式解密(Decryption)，如 AES 等 第一個參數 K 為金鑰，第二個參數 M 為訊息
PE(K，M)	其中 PE 為非對稱式的加密機制，如 RSA、ElGamal…等 第一個參數 K 為加密金鑰，第二個參數 M 為訊息。
PD(K，C)	其中 PD 為非對稱式的解密機制，如 RSA、ElGamal…等 第一個參數 K 為解密金鑰，第二個參數 C 為密文。
F(K)	F()函數是某種固定的運算函數，如 MD5 等 自變數為 K。$K_i=F(SK_i)$，$K_i=SK_i$
R_{ji}	R_{ji} 是執行公開參數的計算，如 $R_{ji}=PE(PK_j，SK_i)$，且對所有群組公開 R_{ji}。
M	M 為群組所要加密之訊息，如 M_i 表示群組 S_i 使用對稱式金鑰 K_i 將訊息加密。

表 5：群組之間關係定義

群組之間關係	定義
直接直屬	直接直屬表示兩個群組有直接關係
隔代直屬	隔代直屬代表群組之間有連續直屬關係
直接上級	如 $S_j>S_i$ 表示群組 S_j 為 S_i 直接上級，反之 S_i 為 S_j 直接下級
隔代上級	如 $S_j>S_i>…S_k$，表示群組 S_j 為 S_k 的隔代上級，反之 S_k 為 S_j 的隔代下級

(一)、 基本運作

本論文提出一套階層式加密的方法套用於階層式群組架構中，在群組中使用一套經過驗證的公開金鑰系統，並發送給每位人員都擁有一組公開金鑰 PK 及一組私密金鑰 SK，且需將公鑰資訊公佈在直屬結構表給所有群組知道。

(二)、 系統啟始設定

1.建立直屬結構表：確定群組中的直接直屬關係、隔代直屬關係等情況，直接直屬關係是兩個群組之間會有直接相關的情形，就像是父與子的關係。如 $S_j>S_i$，$S_j>S_g$，代表 S_j 分別是 S_i、S_g 的直接上級。

隔代直屬關係為兩個群組之間會有連續直屬關係。如 $S_j>S_i$、$S_i>S_h$ 則 $S_j>>S_h$ 的關係為隔代直屬關係，S_j 為 S_h 的隔代上級。依照以上關係而建立直屬結構表。如下表：

表 6：直屬結構表(對圖 1)

群組	直接上級	直接下級	公開金鑰	公開參數
S_j	Null	S_i	PK_j	Null
S_i	S_j	S_h	PK_i	R_{ji}
S_g	S_j	S_e	PK_g	R_{jg}
…	...	…	…	…
S_h	S_i	Null	PK_h	R_{ih}
F()運算公開				

根據直屬結構表中資訊群組能夠從中看到所有人的直屬關係、F()函數運算、公開金鑰 PK 以及相對應之公開參數 R 等。

2.每個群組 S 都有一組公開金鑰 PK 及私密金鑰 SK，其中公開金鑰 PK 是公佈資訊

可以在直屬結構表中查詢。

3.群組 S 將該組的私密金鑰 SK，經過 F(SK)的運算得出該群組資料加密所使用的群組加密金鑰 K，使用傳統式加密將訊息 M 利用群組加密金鑰 K 進行加密。

(三)、 公開參數的產生

群組之間的關係確認之後，如 S_j>S_i 情況中，位階於較低的群組 S_i 要計算公開參數 R_{ji} 並對所有群組進行公開。公開參數 R_{ji} 計算程序為群組 S_i 從結構表中找出直接上級群組 S_j(S_j>Si)的公開金鑰 PK_j，群組 S_i 使用直接上級的公開金鑰 PK_j 將群組 S_i 的私密金鑰使用非對稱式加密 PE 得出公開參數 R_{ji}，並將 R_{ji} 公開至私有雲端上與直屬結構表中，必須讓所有群組皆能看到這些公開參數。

$$\mathbf{R_{ji}=PE(PK_j，SK_i)}$$

(四)、 下級加密金鑰推導程序

當群組 S_j 想推導下級(直接或隔代下級)時：

1. 群組 S_j 從直屬結構表中可以看到所有公開參數 R。
2. 群組 S_j 找到直接直屬下級 S_i 的公開參數 R_{ji} 後，群組 S_j 用私密金鑰 SK_j 對公開參數 R_{ji} 進行解密，得出直接下級私密金鑰 SK_i。
3. 當群組 S_j 想繼續推導隔代下級金鑰時，到直屬結構表中找與直接直屬下級 S_i 有相關的公開參數 R，並使用步驟 2 得到的直接下級私密金鑰 SK_i 對公開參數 R 進行解密，得出隔代下級私密金鑰 SK_h。依照下列演算法進行推導：

推導私密金鑰(S_j → S_h)
{
找出路徑(S_j → S_h)
If(not found) return
$T=S_j$
While(T<>S_h)
{
找出路徑中 T 的直接下屬 S_i
從直屬結構表 S_i 裡找出公開參數 R_{ti}
$SK_i=PD(SK_t，R_{ti})$
$T=S_i$
}
$K_h=F(SK_h)$
}

由於知道公開參數 R_{ji} 推導方法後，直屬下級 S_i 將訊息 M 用傳統式加密法加密後上傳至雲端，直接上級 S_j 就能夠將此加密訊息 M_i 解開。

$$\mathbf{E(K_i，M) \rightarrow M_i}$$

而訊息加密金鑰 K_i 是由群組 S_i 將自己的私密金鑰 SK_i 做 F()的運算。

$$\mathbf{K_i=F(SK_i)}$$

F()運作方式為將私密金鑰取固定長度 160 位元做運算變成群組加密金鑰 K_i。

直接上級 S_j 只要使用私密金鑰 SK_j 將公開參數 R_{ji} 解開後就能得到群組 S_i 的私密金鑰 SK_i，並將 SK_i 做 F()運算得到 K_i，利用 K_i 就能將訊息 M_i 解密成訊息 M。

$$\mathbf{PD(SK_j，R_{ji}) \rightarrow SK_i}$$

$$\mathbf{F(SK_i) \rightarrow K_i}$$

$$\mathbf{D(K_i，M_i) \rightarrow M}$$

階層較低的群組要傳送給階層高的直屬關係群組就能使用以上步驟進行資料傳遞。

(五)、 加入新成員

當群組之中有新成員加入階層時有以下兩種情況：當新加入的成員為末端節點的時候，新成員 S_a 加入時會得到一組非對稱式加密的公開金鑰 PK_a 及私密金鑰 SK_a，並且透過直屬結構表中確認自己的直接上級，利用直接上級群組 S_i 的公開金鑰 PK_i 將自己的私密金鑰 SK_a 作 PE 運算得出公開參數 R_{ia}，並將 R_{ia} 上傳至雲端及更新直屬結構表中的資訊。(結構不變及結構改變時並不影響)

$R_{ia}=PE(PK_i，SK_a)$

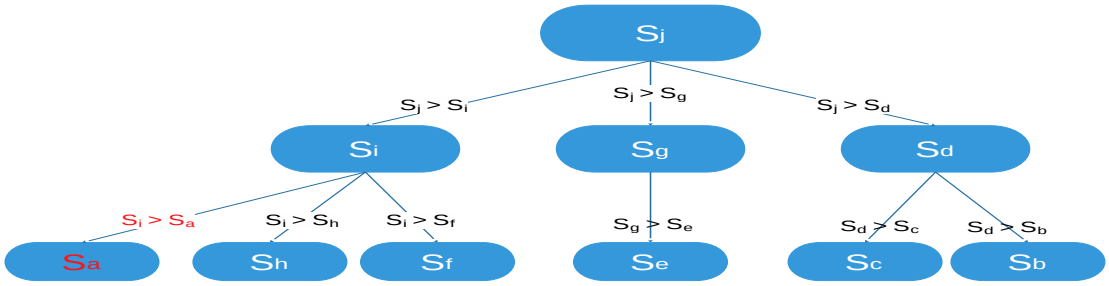


圖 2：加入新成員-末端節點時(結構不變)

結構改變時使用的方法相同。

$R_{ha}=PE(PK_h，SK_a)$

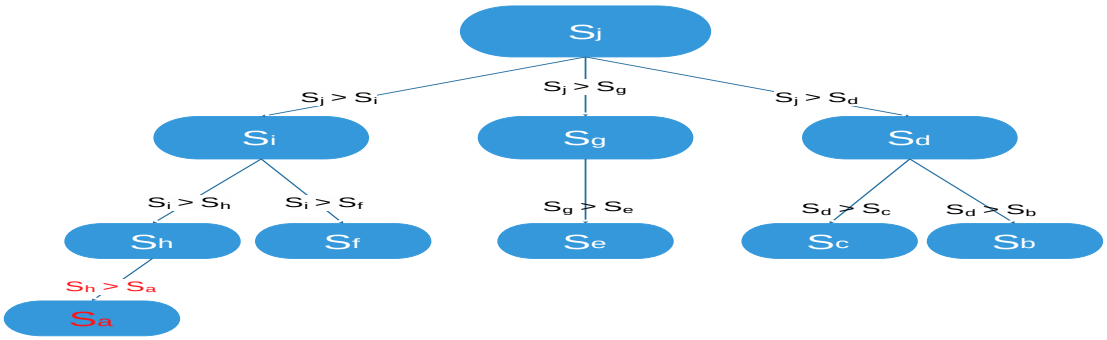


圖 3：加入新成員-末端結點時(結構改變)

當新加入的成員為中間節點時，新成員 S_a 必須先從直屬結構表中尋找與自己的關係為直接直屬關係，然後新成員 S_a 會得到一組非對稱式加密的公開金鑰 PK_a 及私密金鑰 SK_a，新成員 Sa 透過直屬結構表找到直接上級群組 S_i 並使用直接上級群組的公開金鑰 PK_i 將自己的私密金鑰 SK_a 作 PE 運算得出公開參數 R_{ia}，並將 R_{ia} 上傳至雲端及更新直屬結構表中的資訊。另外新成員的直接下級，透過直屬結構表找到新成員的資訊並更新公開參數 R_{ih} 改為 R_{ah} 上傳至直屬結構表中。

$R_{ia}=PE(PK_i，SK_a)$

更新 $R_{ih}\rightarrow R_{ah}=PE(PK_a，SK_h)$

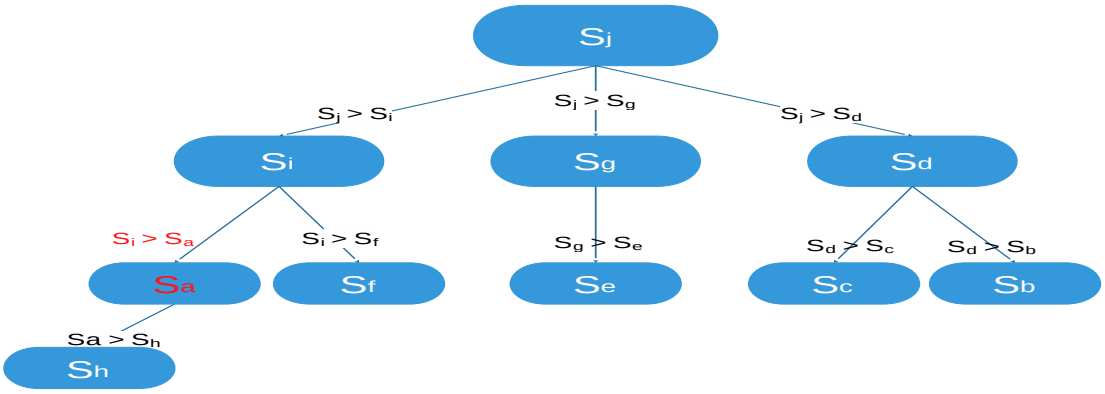


圖 4：加入新成員-中間節點時

(六)、 成員離開時

當群組之中成員離開時有以下幾種狀況：當成員離開時為末端節點的時候，假設群組 S_a 離開時必須先將群組 S_a 的公開金鑰 PK_a 及私密金鑰 SK_a 回收刪除，並將刪除直屬結構表中 S_a 的資料及公開參數。 (結構不變及結構改變時並不影響)

刪除 S_a 公開參數 R_{ia}：

刪除 $R_{ia}=PE(PK_i，SK_a)$

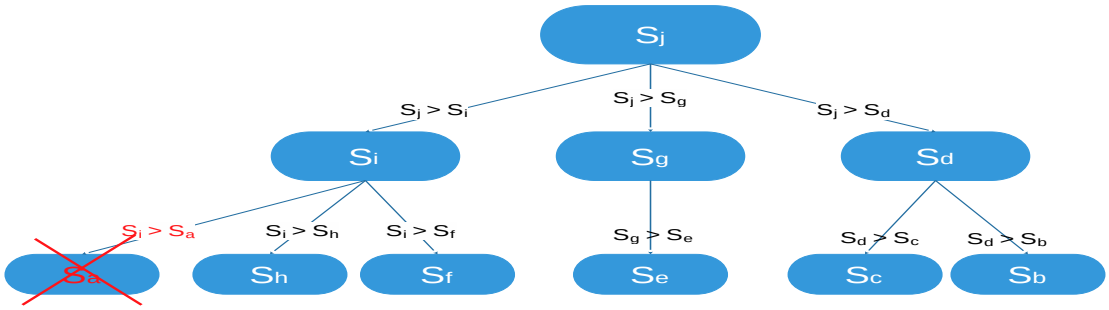


圖 5：成員離開-末端節點時(結構不變)

結構改變時使用的方法相同。

刪除 $R_{ha}=PE(PK_h，SK_a)$

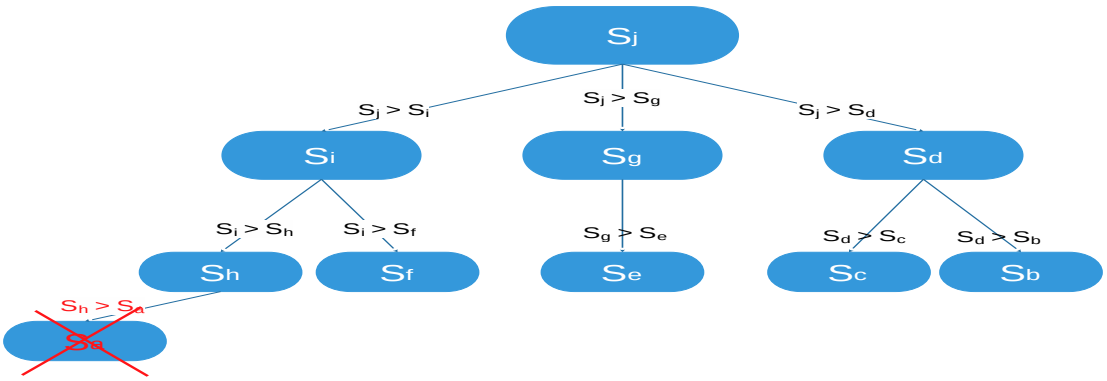


圖 6：成員離開-末端節點時(結構改變)

當群組中成員離開時為中間節點時，假設群組 S_i 離開，必須先將群組 S_i 的公開金鑰 PK_i 及私密金鑰 SK_i 回收，原先屬於群組 S_i 的資料讓直接上級 S_j 擁有，並且將刪除直屬結構表中 S_i 的資料。原本 S_i 的直接下級 S_h 及 S_f 群組，公開金鑰 PK、私密金鑰 SK、公開參數 R_{ih} 及 R_{if} 都要進行刪除。並且從新分配新的公開金鑰 PK 及私密金鑰 SK 給群組 S_h 及 S_f 及更新直屬結構表。

刪除 $R_{ji}=PE(PK_j，SK_i)$

刪除 R_{ih} 及 R_{if}

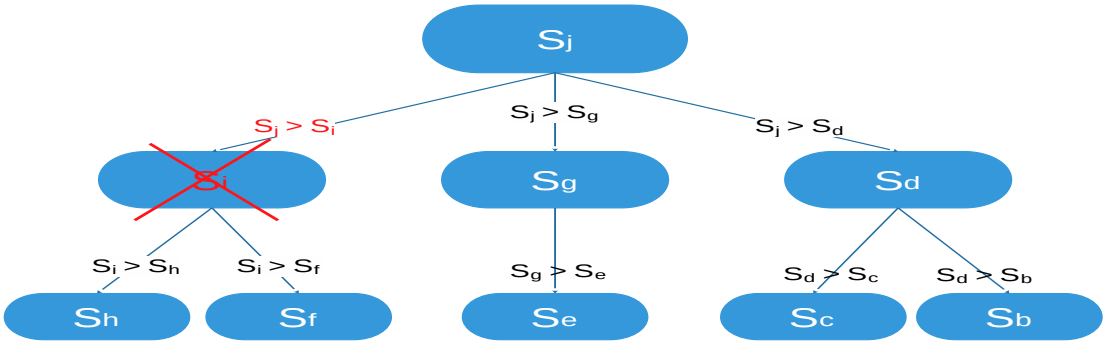


圖 7：成員離開時-中間節點時

肆、相關階層式金鑰比較

本章節將介紹效率及安全性分析以及和相關階層式金鑰方法做多個面向進行比較後建立比較分析表。

一、效率及安全性分析

本方法，結構中群組有一把公開金鑰 PK 及私密金鑰 SK，並且將群組私密金鑰 SK，用直接上屬群組公開金鑰加密產生公開參數 R。利用直屬結構表公開各群組公開金鑰，相對公開參數 R 及直接上屬群組等資訊。此方法省去訪問與回應動作，金鑰分配的數量也不會過多，使用者(群組)只需要保護好自己的一把私密金鑰 SK_i。

(一)、 反向攻擊

假設 $S_j > S_i$，但下級群組 S_i 想要得到直接上級群組 S_j 的私密金鑰 SK_j，由於 SK_j 只會用更上級的公開金鑰 PK 進行加密，而 S_i 無法解密得到更高階級的私密金鑰 SK，故無法得到直屬上級的私密金鑰 SK_j。

(二)、 資料收集攻擊

群組內部使用者雖然能在直屬結構表中找到公開參數 R_{ji}，但是如果要解開公開參數 R_{ji}，只能夠一層一層往上推得到最高層級群組的私密金鑰 SK_j 才能夠將公開參數 R_{ji} 推解出來，但是最高層級群組不會將私密金鑰 SK_j 放至雲端上，所以無法進行破解。

(三)、 群組之間互相攻擊

當群組 S_i 跟 S_g 都有一個直屬上級 S_j 的情況下，當 S_i 想要破解 S_g 的訊息群組加密金鑰 K_g，在群組 S_i 裡面只擁有 K_i 以及能獲得 R_{ji}、R_{jg}，但是從這資訊中使用者 S_i 依然無法破解出 K_g。

(四)、 群組聯合攻擊

假設群組 S_j 有兩個子群組 S_i 及 S_g，而兩個群組聯合起來想要得到 S_j 的私密金鑰 SK_j，但是能夠得到的資訊只有 K_i、K_g、R_{ji}、R_{jg} 及 PK_j，依無法得到私密金鑰 SK_j。

二、比較面向特點的說明

在此小節，我們將第二章相關研究中各位學者所提機制及本論文所提機制作多個面向比較分析及討論。

(一)、 Akl & Taylor(1983)所提機制

在 Akl & Taylor(1983)等人所提機制當中，在階層式架構當中利用質數的分配及運算來作為階層式金鑰的加密推導，讓上級群組能夠直接推導出下級群組的私密金鑰，但是這其中也包含了幾個優點及缺點如下

優點：

1. 不用找路徑

缺點：

1. 一定要有 CA 系統來分配。
2. 群組的路徑越大，會有 n 個大質數相乘。
3. 如果是單一垂直結構的群組會有問題，因為不能使用自己及下級群組的質數，這樣整條群組將分配不到任何質數。
4. 成員的加入及離開時會影響到整個階層式架構下的群組，並且需要 CA 進行操作更新。

(二)、 Lo, Hwang & Liu 等人所提機制

在階層式架構中由 Lo,Hwang & Liu(2011)等人所提機制當中，改良了 1983 年 Akl & Taylor(1983)兩人所提出的機制，讓推導金鑰時需要先去判斷每個群組在階層架構中是屬於中間節點、末端節點還是擁有共同上級的情況，才能夠去進行金鑰的推導。

在 Lo,Hwang & Liu(2011)等人的方法中雖然改善金鑰分配方式，但是結構上則有限

制(例如不能使用單一垂直結構)及安全上群組使用聯合攻擊就能夠得到直接上級的金鑰。另外 Lo,Hwang & Liu(2011)等人的方法當中最大的問題就是 CA 系統的存在沒有進行改良，在加入成員及成員離開時一定需要有 CA 進行操作。如果 CA 系統被惡意入侵更改裡面的資料或金鑰在發送給群組使用者，這樣群組使用者使用此金鑰加密後惡意使用者就能夠解開這些資料並得到相關機密資料，或是 CA 系統掛掉或是網路遭到癱瘓時，這整個階層式架構就猶如幻影的消失，使這個模式完全無法使用。

(三)、 Chu-Hsinh Lin 所提機制

在 Lin(1997)所提機制中，由 CA 選擇一把公開金鑰 PK 和一把對應的私密金鑰 SK。每個群組選擇一把群組金鑰 K 並使用 CA 公佈的公開金鑰 PK 加密傳給 CA，CA 解密後得出群組關係後(如 $S_j>S_i$)計算公開參數 r_{ji}，金鑰推導階段時 CA 先判斷群組此次要求是否安全，若安全將公開參數 r_{ji}，群組藉由公式能推導群組金鑰 K_i，群組更新金鑰則使用相同方法傳送給 CA 計算。雖然此方法方便管理，但是攻擊者如果得到舊的群組金鑰並針對計算公式得出新的群組金鑰以及群組聯合攻擊藉由識別碼 ID 推算公式得出直接上級的群組金鑰 K。另外最大缺點為所有運做一定需要透過 CA 系統。

綜合以上優缺點，本論文所提機制將無需使用 CA 系統也能夠進行運作，並且在結構上無限制能夠輕鬆導入組織階層中，以及適用於所有非對稱式演算法不會因為演算法而限制使用，除此之外，此系統操作上簡單以及金鑰推導快速，實際上對使用者可能存在著一些負擔但是方法完整性較於 Akl-Tayor 等人所提機制、Lo-Hwang-Liu 等人所提機制 Chu-Hsing Lin 較高。在安全性方面適用各種非對稱式演算法的關係，所以困難度在於解離散對數以及因數分解之問題。

另外在本論文中，也提出組織內部加入成員及成員離開對群組之間的影響，相較於其他方法本論文所提機制會改變到的群組相對較少。

由表 4-1 與表 4-2 中，將本論文所提機制及相關研究之三位學者所提機制依照各個特點進行比較分析及描述。

表 7：相關研究之比較功能表

	本論文所提機制	Akl-Tayor 等人所提機制	Lo-Hwang-Liu 等人所提機制	Chu-Hsing Lin
有無 CA	無*	有	有	有
公開參數個數量	2N-1 個	N 個	N 個	2N 個
公開參數的數值大小(N 變大或結構改變時)	不變	N 增加一個，會多乘一個大質數結構改變公開參數的絕對數質會更改	N 增加一個，會多乘一個大質數結構改變公開參數的絕對數質會大大改變	不變
方法的演算核心	只要非對稱性皆可(RSA、ElGamal、DSA….等)	模數為兩大質數相乘之模指數運算	模數為兩大質數相乘之模指數運算以及末端節點的雜湊函數運算	只能 RSA
比較金鑰推導所需要的計算(例如：S_j 到 S_h 假設路徑長度是 n)	1.要算出路徑 2.計算總量 n*PD 個運算時間加上 F()運算時間	1.不用計算路徑 2.一個大除法後加指數部分，至少為 n 個大質數相乘的模指數運算	1.不用計算路徑 2.跟結構有關最差狀況為一個大除法後加指數部分為 n 個大質數	1.要計算路徑 2.n 個模指數運算

			相乘的模指數運算	
安全性問題	1.結構無限制 2.離散對數或因數分解的困難度	1.結構有限制不能單一垂直結構 2.因數分解的困難度	1.結構有限制安全有疑慮(可群組聯合攻擊)	1.結構無限制 2.因數分解的困難度

*註：本論文所提機制不需要有 CA 系統，只需要組織圖。

表 8：組織內成員加入及離開比較表

	本論文所提機制	Akl-Tayor 等人所提機制	Lo-Hwang-Liu 等人所提機制	Chu-Hsing Lin
加入新成員				
新成員為末端節點且不改變結構	原先群組不會改變，多一個公開參數	*除了直屬關係以外全部改變	*原先群組不會改變，多一個公開參數	*原先群組不會改變，多一個公開參數
新成員為末端節點但改變結構	原先群組不會改變，多一個公開參數	*除了直屬關係以外全部改變	*所有有直屬關係皆會改變	*原先群組不會改變，多一個公開參數
新成員為中間節點	新成員的直接下屬全部改變，加上新成員的公開參數	*除了直屬關係以外全部改變，加上新成員公開參數	*所有新成員上屬的公開參數皆要改變，加上新成員的公開參數	*新成員的直接下屬全部改變，加上新成員的公開參數
成員離開				
成員離開為末端節點且不改變結構	減少一個公開參數	*除了直屬關係以外全部改變。	*減少一個公開參數	*減少一個公開參數
成員離開為末端節點但改變結構	減少一個公開參數	*除了直屬關係以外全部改變	*與該成員有直屬關係皆要改變	*減少一個公開參數
成員離開為中間節點	該成員所有下屬全部要改變	*除了直屬關係以外全部改變	*與該成員有直屬關係皆要改變	*該成員所有下屬全部要改變

*註：機制運算時需要 CA 輔助

伍、結論與建議

在網際網路應用、無線化及行動化發展日漸普及下，使得雲端運算在個人、組織和企業中廣泛使用。階層式架構下各群組資料使用的管控，為確保組織和企業在雲端運算安全的重要考量。主要考量的議題包含加解密的安全性、系統運作的效率，組織群組結構或成員變更的影響。

本論文提出一套應用在雲端運算中，使階層式結構中群組間進行資料授權能夠簡單快速方法。本方法中，結構中群組有一把公開金鑰 PK 及私密金鑰 SK，並且將群組私密金鑰 SK，用直接上屬群組公開金鑰加密產生公開參數 R。利用直屬結構表公開各群組公開金鑰，相對公開參數 R 及直接上屬群組等資訊。另外本論文相較於 Akl-Tayor 等

人、Lo-Hwang-Liu 等人以及 Chu-Hsing Lin 所提的方法，如表 4-1 及表 4-2 所示，本論文並不需要有 CA、階層式結構擴大較少公開參數改變、適用所有非對稱演算法、金鑰推導方式運算較簡單、新成員加入時原先群組影響較小及成員離開時群組改變較少等以上這些特點，能夠有效改變原本階層式架構下加解密的方式與效率。

參考文獻

[1] 陳正鎔，民 102。『階層式金鑰產生之方法-以乘法反元素為研究基礎』，2013 企業架構與資訊科技國際研討會

[2] Akl, S. G., & Taylor, P. D. (1983). Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer Systems (TOCS)*, *1*(3), 239-248.

[3] Cacic, B. J., & Wei, R. (2007). Improving Indirect Key Management Scheme of Access Hierarchies. *IJ Network Security*, *4*(2), 128-137.

[4] Chen, H. Y. (2004). Efficient time-bound hierarchical key assignment scheme.*Knowledge and Data Engineering, IEEE Transactions on*, *16*(10), 1301-1304.

[5] Denning, D. E., Akl, S. G., Heckman, M., Lunt, T. F., Morgenstern, M., Neumann, P. G., & Schell, R. R. (1987). Views for multilevel database security. *Software Engineering, IEEE Transactions on*, (2), 129-140.

[6] Diffie, W., & Hellman, M. E. (1976). New directions in cryptography.*Information Theory, IEEE Transactions on*, *22*(6), 644-654.

[7] ElGamal, T. (1985, January). A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology* (pp. 10-18). Springer Berlin Heidelberg.

[8] Hwang, M. S., & Yang, W. P. (2003). Controlling access in large partially ordered hierarchies using cryptographic keys. *Journal of Systems and Software*, *67*(2), 99-107.

[9] Jaeger, T., & Schiffman, J. (2010). Outlook: Cloudy with a chance of security challenges and improvements. *Security & Privacy, IEEE*, *8*(1), 77-80.

[10] Lin, C. H. (1997). Dynamic key management schemes for access control in a hierarchy. *Computer communications*, *20*(15), 1381-1385.

[11] Lo, J. W., Hwang, M. S., & Liu, C. H. (2011). An efficient key assignment scheme for access control in a large leaf class hierarchy. *Information Sciences*, *181*(4), 917-925.

[12] MacKinnon, S. J., Taylor, P. D., Meijer, H., & Akl, S. G. (1985). An Optimal Algorithm for Assigning Cryptographic. *IEEE Transactions on computers*,*100*(34).

[13] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126.

[14] Wang, Shyh-Yih, and Chi-Sung Laih. "Cryptanalysis of Hwang–Yang scheme for controlling access in large partially ordered hierarchies." *Journal of Systems and Software* 75.1 (2005): 189-192.