階層叢集式行動隨意網路之動態存取控制機制

曹偉駿
大葉大學資訊管理學系
E-mail: wjtsaur@yahoo.com.tw

黃南翔
大葉大學資訊管理學系
E-mail: f9121224@hotmail.com

***摘要*—群組計算及通訊的應用，刺激行動隨意網路對群組存取控制的需求。由於動態的成員關係以及缺乏認證中心，這些因素造成行動隨意網路的存取控制更具挑戰。近年來，一些學者試圖以驗證門檻式簽章的方法，提出應用於行動隨意網路的群組存取控制機制。然而鄰近節點數量若小於機制所定的門檻值，則無法使用門檻式簽章，導致未能滿足行動隨意網路的動態環境。本研究基於階層叢集式架構，並結合有效率的橢圓曲線密碼系統及安全濾器技術(secure filter)，提出一套適用於行動隨意網路的群組存取機制，以解決現有使用門檻式簽章的存取控制機制所遭遇之困難。在我們所提的機制中，當群組的成員關係發生改變時，藉由更新少數叢集頭的安全過濾器即可達成安全的群組存取控制。**

***關鍵詞*—網路安全(network security)、群組存取控制(group access control)、階層式叢集(hierarchical clustering)、行動隨意網路(mobile ad hoc networks)**

一、前言

行動隨意網路(mobile ad hoc networks, MANETs)是一種不需要基礎設施的架構，且可以快速建立的網路[2]。在 MANETs 中，無線行動裝置可以不需要任何管理中心即可動態形成各自的網路。換句話說，MANETs 中的節點允許以自我組織的方式，動態加入或離開網路。由於近年來無線網路技術的發展，加上行動設備的普及，MANETs 的應用範疇越來越多也備受重視，在軍事任務、執法場合、緊急救援及人道援助都有許多知名的應用[17]。科技的進步無疑為人們帶來了便利，然而沒有穩定的基礎設施以及缺少管理中心，以及行動設備的電源、計算能力、儲存空間等資源有限，導致 MANETs 的安全性成為隱憂。再加上網路環境的變因，如動態節點造成拓樸改變，及不可靠的無線通道和有限的頻寬，進一步增加了問題的複雜性。因此，解決 MANETs 的安全問題是一個非常大的挑戰[21]。

一些學者試圖以驗證門檻式簽章的方法，提出應用於 MANETs 的群組存取控制機制。然而在這些機制中[10, 12]，鄰近節點的數量需要大於機制的門檻值才能使用，但在節點高流動的情況下，鄰近節點的數量有可能小於門檻值。且在網路建立的初期，鄰近節點的數量也有可能小於門檻值。門檻值的限制導致這些機制無法真正滿足 MANETs 的動態環境，以及快速建立通訊網路的優點。本研究基於高彈性的階層叢集式架構，並且結合有效率的橢圓曲線密碼系統及安全過濾器(secure filter)，發展一套適用於 MANETs 的群組存取機制，以解決現有使用門檻式簽章的存取控制機制所遭遇之困難。在本研究所提的機制中，當群組的成員關係發生改變時，藉由更新少數叢集頭的安全過濾器即可達成安全的群組存取控制。換句話說，確保受保護的資料只有經認證的使用者才能存取，且叢集頭能存取其下層群組之資料。此外，最重要的是能防止權限低的節點存取權限高的節點內容。

二、文獻探討

(一) 階層存取控制

階層存取控制的定義[6]為：在組織中，使用

者與資料被分成一群不相交集的類別，且每位使用者皆被指定到特定的類別，稱為存取權限。假設 m 個不相交集的類別 $C_1, C_2, ..., C_m, m \in N$ 。$C_j < C_i (i, j \in N)$ 表示類別 C_i 的存取權限大於類別 C_j ，換句話說，位在類別 C_i 使用者可以儲存或讀取類別 C_j 內的資料，反之則不允許。另一方面，每位類別 C_i 的使用者都擁有一把密鑰 K_i ，使用者可以藉由 K_i 加密欲傳送的訊息，或解密同為類別 C_i 使用者所加密的訊息。然而密鑰 K_i 只能用於類別 C_i 成員之間的訊息加密及解密，當類別 C_i 的使用者想解密下層類別 C_j 成員所加密的訊息時，必須要先取得正確的密鑰 K_j 。顯然地，若使用者想取得所有下層類別的訊息，則必須儲存所有下層類別的密鑰。以圖 1 為例，如果 C_1 想解密所有下層類別所加密的訊息時，就必須額外持有 $K_2, K_3, ..., K_9$ 等密鑰。但當類別數量越來越多時，越高權限的使用者就必須儲存越多數量的密鑰，如此勢必造成管理金鑰上的困難，且金鑰問題可能成為安全上的疑慮。

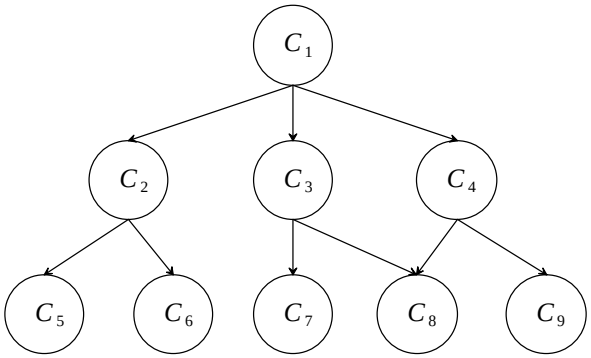

圖 1　各類別之階層架構

因此，如何建立一套良好的存取控制機制，讓使用者不需要擁有太多金鑰就能存取權限內所有的資訊，是這領域的研究者首要的工作。Wu 等學者[18]提出一種能實際解決問題的函數，稱為安全過濾器(secure filter)。在此方法中，類別 C_i 選擇專屬的密鑰 K_i 以及一個祕密的隨機參數 s_i 。之後 CA 產生類別 C_i 的安全過濾器 $f_i(x)$ 。若 $C_j < C_i$ ，則 $f_j(x) = \prod_i (x - g_j^{s_i}) + K_j$ ，g_j 為 C_j 所選擇的公開參數。如果位在類別 C_1 的使用者想存取類別 C_8 內的資料時。使用者首先需要取得 C_8 的公開資訊 $(f_8(x), g_8)$ 。然後使用只有類別 C_1 才知道的祕密參數 s_1 ，計算函數 $f_8(g_8^{s_1})$ 後即可獲得 C_8 的密鑰 K_8 ，如方程式(1)。

$$\begin{aligned} f_8(g_8^{s_1}) &= (g_8^{s_1} - g_8^{s_1})(g_8^{s_1} - g_8^{s_3})(g_8^{s_1} - g_8^{s_4}) + K_8 \\ &= K_8 \end{aligned} \quad (1)$$

Chung 等人[3]結合安全過濾器及橢圓曲線密碼系統(elliptic curve cryptosystem, ECC)提出一套階層存取控制機制，並與其他機制之效能分析如表 1。在表 1 中，Wu 及 Wei 之機制[20]與 Hwang 及 Yang 之機制[4]，其公開參數的數量與長度會隨著繼承者的數量而增加，因此其所需的儲存空間將漸漸變大。而 Chang 等人[1]和 Chung 等人[3]之機制，各類別所需儲存的公開參數皆為固定。在計算負載方面，ECC 所需的金鑰長度及頻寬皆優於離散對數系統及因數分解系統，同時也優於其他基於 RSA 密碼系統之階層存取控制機制[15, 22]。

雖然表 1 中的四種階層存取控制方法皆能有效解決 MANETs 動態拓樸之問題，但由於 MANETs 之傳輸及節點的能力有限，因此基於效能與儲存量之觀點，在上述各種方法中，結合 ECC 的安全過濾器是最適合應用於 MANETs 階層存取控制之方法。

表 1　階層存取控制機制之效能比較

項目	Chang 等人之機制[1]	Wu 及 Wei 之機制[20]	Hwang 及 Yang 之機制[4]	Chung 等人之機制[3]
金鑰產生	指數計算	指數計算	因式分解	ECC + secure filter
金鑰推導	指數計算	指數計算	因式分解	ECC + secure filter
新增/刪除類別	部份更新	部份更新	部份更新	部份更新
建立/廢止關係	部份更新	部份更新	部份更新	部份更新
改變密鑰	部份更新	部份更新	部份更新	部份更新
公開參數儲存量	固定且少量	大	大	固定且少量

(二) 階層叢集式 MANETs 動態金鑰管理機制

由於 MANETs 的節點資訊為公開，為防止非相關人士取得重要資訊，故節點資訊皆須加密保護，只允許擁有正確金鑰的節點才能進行解密以存取資訊，而其他節點因為無法解密訊息，所以沒辦法取得資訊。為達成保護資訊的目的，各節點皆須要擁有屬於自己的金鑰，做為後續的加解密之用，而金鑰的發行及分配需要一套專門的管理機制，稱為金鑰管理機制。

叢集架構是一種 MANETs 重要的研究主題[23]。這種架構可以確保系統在大量節點及高流動性的情況中，維持基本水平的效能，是一種有效的拓樸控制方法。叢集可提供三個優點。首先，叢集能促進系統資源再使用，以增加效能。藉由叢集頭的幫助，能更有效的協調傳輸事件。如此可以大量節省用於減少傳輸碰撞的資源。第二個優點是路由安全，叢集頭的集合可以組成跨叢集路由的虛擬骨幹，且路由資訊的產生和延伸可以限制在節點的集合中。最後，叢集架構能讓 MANETs 更穩定和方便管理，當行動節點改變自己所在的叢集時，只需更新對應叢集的資訊。因為局部的變化不需要更動整體網路，所以每個行動節點需要處理及儲存的訊息也能大大地減少。

然而，基於叢集架構的 MANETs 卻存在固有的缺點。相較於平坦式架構，叢集架構的建立和維持須要額外的成本。為彌補此一瑕疵，學者 Tsaur 及 Pai[14]以階層叢集式架構，結合有效率的橢圓曲線密碼系統、具自我認證的公開金鑰密碼系統[13]、安全的過濾器技術，提出階層叢集式動態金鑰管理機制以解決成本問題。在此機制中，每個叢集頭負責協調叢集中所有成員的群組金鑰，且進一步管理動態群組金鑰分配。值得一提的是，叢集頭無法取得叢集中成員的密鑰，除非叢集頭能破解單向雜湊函數，以及棘手的橢圓曲線離散對數困難度。此外，在叢集頭動態地變動後，包括節點加入/離開，以及叢集合併/分割的情況，機制能安全地廢除原先叢集的群組金鑰，且進一步更新再次形成之叢集的群組金鑰。表 2 為此機制與其 Wu 等人[19]以及 Zhang 等人[24]之機制比較[14]。相比於其他機制，Tsaur 及 Pai 所提之機制在機制弱點與前後向安全的項目上，優於另外兩者。因此本研究採用此金鑰管理機制，以管理各群組之金鑰。

表 2　各種金鑰管理機制之比較

項目	Wu 等人之機制[19]	Zhang 等人之機制[24]	Tsaur 及 Pai 之機制[14]
公開金鑰密碼系統	ElGamal	Pairing	基於 ECC 的自我認證公開金鑰
安全基礎	離散對數問題	Bilinear Diffie-Hellman	橢圓曲線離散對數問題
金鑰分配	(k, n) 祕密分享機制	(k, n) 祕密分享機制	安全過濾器技術
節點認證	Schnorr 簽章機制	以身份為基礎的公開金鑰密碼系統	基於 ECC 的自我認證公開金鑰
信賴管理	CA 與 k 個鄰近節點	CA 與 k 個鄰近節點	階層叢集式架構
故障節點	功能受到限制	功能受到限制	功能不受影響
機制弱點	當節點較少時可能無法取得群組金鑰	1. CA 知道所有使用者的密鑰 2. 當節點較少時可能無法取得群組金鑰	無(因為沒採用(k, n) 祕密分享機制, 且惡意叢集頭無法取得其他任何節點的密鑰)
動態更新群組金鑰	是	部份(只有節點加入時)	是
前向及後向安全	否	否	是

註：1. k與n分別為祕密分享機制的門檻值及總參與數量
2. CA (Certification Authority)表示認證中心

(三) MANETs 存取控制之相關研究

現有的一些 MANETs 存取控制機制大多使用門檻密碼學[25, 8, 7, 11, 9]，以允許新節點分享群組祕密，進而建立安全的通訊及存取群組內的資訊。不幸的，這些並非理想的機制。由於這些方法是在 MANETs 中事先分佈分散式的認證中心(certification authority, CA)，當節點取得大於門檻的信任值，即可認證成功，所以此方法只適用於鄰近節點數量大於門檻值的情況。然而，在網路環境不斷在改變的 MANETs 中，無法一直維持周遭節點數量大於門檻值。且快速建立通訊是 MANETs 的優點之一，在網路建置的初期，也無法事先分佈可信任的節點，種種的因素限制了這種方式在 MANETs 上的使用。

一些學者考慮到這個問題，因此提出一套基於 RSA 簽章的許可協定[10]。然而，已有學者提出這套機制並不安全[5]。由於至今所提出的 MANETs 群組存取控制機制並非理想，因此在 MANETs 的環境中，仍需要一套良好的群組存取控制機制。

在 MANETs 的環境下，由於節點可以自由移動，故一個動態的群組存取控制機制需要滿足下列安全條件[16]：

1. 前向安全 (forward secrecy)

當系統增加一叢集時，上層的叢集頭雖然可取得新加入叢集現有之群組金鑰，但卻無法以此金鑰解密該叢集未加入前的資訊，此一特性稱為前向安全。

2. 後向安全 (backward secrecy)

當叢集離開系統之後，上層的叢集頭雖然可以之前的群組金鑰，存取該叢集舊有的訊息。但無法再以此一群組金鑰，存取該叢集之後的群組資訊，此一安全稱為後向安全。

三、建構安全動態群組存取控制機制

(一) 符號定義

本小節介紹一些關於階層叢集式 MANETs 的名詞，以及機制所使用之符號，並以圖 2 為例子說明。

- 節點：在 MANETs 中，每個行動設備都可以稱為節點，以 αi_j 表示。而 $\alpha \in \{A \sim Z\}$，且 i 與 j 皆為大於零的整數。例如 $A4_2$、$G6_6$、$N8_3$ 等都是節點。
- 叢集成員：數個節點可被分在同一個群組形成叢集 αi，例如成員 $C1_2$、$C1_4$、$C1_8$ 以及 $C1_9$ 可構成叢集 $C1$。
- 叢集頭：每個叢集頭都是從叢集中所選出，且叢集頭可以管理叢集的成員，以 αi_1 表示叢集頭。例如圖 2 中的 $A4_1$、$C1_1$、$F2_1$ 都是叢集頭。
- 跨叢集頭：跨叢集頭是從數個叢集頭之中所選出，目的是為了協調這些叢集頭。例如圖 2 中的 $R9_1$、$X3_1$、$Y5_1$。
- 叢集頭的根節點：根節點是從所有跨叢集頭中所選出。例如圖 2 中的 $R9_1$。
- $K_{Group-\alpha i}$：叢集 αi 的群組金鑰，$K_{Group-\alpha i} \in [2, n-2]$。
- n：為一 160 位元的大質數。
- 橢圓曲線群 $E_P(a,b)$：$E_P(a,b)$ 定義為 $y^2 = x^3 + ax + b \pmod p$，$p$ 為一大質數，且其係數滿足 $4a^3 + 27b^2 \pmod p \neq 0$。
- B：從 $E_P(a,b)$ 中選取的一個基點，其秩為一非常大的值。

(二) 安全的動態群組存取控制機制

當 MANETs 的各節點執行完 Tsaur 及 Pai 於 2007 年所提出之階層叢集式動態金鑰管理機制後，各節點皆已向所屬的叢集頭註冊完成，並取得自己密鑰 。為了讓叢集頭可以取得下層叢集的資訊，以下說明本研究所提出之機制步驟：

1. 各叢集頭以密鑰 sk_{ai_1} 計算公開值 $V_{ai_1} = sk_{ai_1} \cdot B$。
2. 叢集頭產生自己叢集的安全過濾器 $f_{ai}(x) = \prod (x - sk_{ai_1} \cdot V_{t_1}) + K_{Group-ai} \pmod n$，$V_{t_1}$ 表示為上層叢集頭的公開值。

若上層的叢集頭想存取下層叢集 ai 內的資訊時，只需取得該叢集的過濾器 $f_{ai}(x)$ 以及公開的 V_{ai_1}，並以自己的私鑰 sk_{self} 計算 $f_{ai}(sk_{self} \cdot V_{ai_1})$，即可得到叢集 ai 的群組金鑰 $K_{Group-ai}$。方程式(2)驗證叢集頭 $Y5_1$ 可以正確取得叢集 $C1$ 的群組金鑰：

$$\begin{aligned} f_{C1}(sk_{Y5_1} \cdot V_{C1_1}) &= (sk_{Y5_1} \cdot V_{C1_1} - sk_{C1_1} \cdot V_{R9_1})(sk_{Y5_1} \cdot V_{C1_1} \\ &\quad - sk_{C1_1} \cdot V_{Y5_1}) + K_{Group-C1} \pmod n \\ &= (sk_{Y5_1} \cdot sk_{C1_1} \cdot B - sk_{C1_1} \cdot V_{R9_1})(sk_{Y5_1} \cdot sk_{C1_1} \cdot B \\ &\quad - sk_{C1_1} \cdot V_{Y5_1}) + K_{Group-C1} \pmod n \\ &= (V_{Y5_1} \cdot sk_{C1_1} - sk_{C1_1} \cdot V_{R9_1})(V_{Y5_1} \cdot sk_{C1_1} \\ &\quad - sk_{C1_1} \cdot V_{Y5_1}) + K_{Group-C1} \pmod n \\ &= K_{Group-C1} \end{aligned} \quad (2)$$

基於網路延展性的考量，因此需要注意新增叢集的情況。另外，叢集的存取權限也可能需要提升或降低。故探討叢集動態的情況，以及叢集頭存取權限變更的情況。

(三) 新增及刪除叢集

刪除底層叢集的動作對本機制而言非常簡單，離開的叢集頭 ai_1 只需刪除過濾器 $f_{ai}(x)$ 並且變更群組金鑰 $K_{Group-ai}$，即可保護叢集 ai 內的資訊不再被其他叢集存取。此外，在 MANETs 的動態環境中，可能視情況而需要增加新的叢集，且叢集頭也有離開網路的可能，故以下針對新增

叢集頭與變更叢集頭的情況分別說明。

1. 增加新叢集

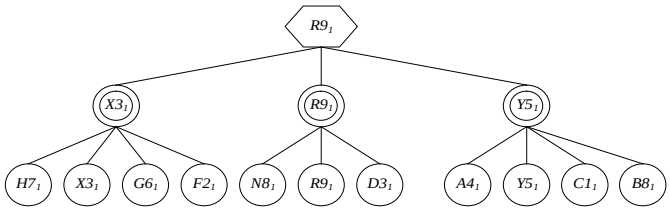

圖 2 新增叢集頭 $B8_1$

如圖 2，當叢集 $Y5$ 增加子叢集 $B8$ 後，為維持安全的群組存取控制，需要增加叢集頭 $B8_1$ 以便管理叢集 $B8$，故將執行下列步驟：

(1). 叢集頭 $B8_1$ 向 $Y5_1$ 提出註冊之請求，$Y5_1$ 產生 $B8_1$ 的密鑰 sk_{B8_1}，並將叢集 $B8$ 上層叢集頭之公開的 V_{Y5_1} 與 V_{R9_1} 回傳給 $B8_1$。

(2). $B8_1$ 取得需要的訊息後，計算 $V_{B8_1} = sk_{B8_1} \cdot B$ 並產生叢集 $B8$ 的安全過濾器：

$$f_{B8}(x) = (x - sk_{B8_1} \cdot V_{R9_1})(x - sk_{B8_1} \cdot V_{Y5_1}) + K_{Group-B8} (\text{mod } n)$$

(3). 當 $Y5_1$ 與 $R9_1$ 需要取得叢集 $B8$ 的資訊時，先從 $B8$ 處取得公開的 $f_{B8}(x)$ 以及 V_{B8_1}，再藉由計算 $f_{B8}(sk_{Y5_1} \cdot V_{B8_1})$ 及 $f_{B8}(sk_{R9_1} \cdot V_{B8_1})$ 可分別取得叢集 $B8$ 的群組金鑰 $K_{Group-B8}$。

2. 變更叢集頭

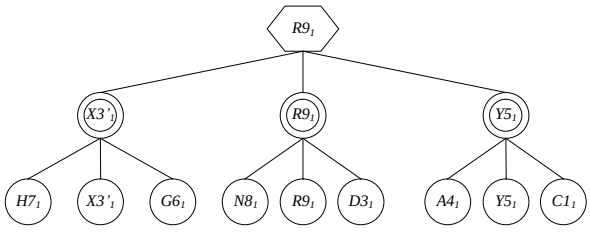

圖 3 將叢集頭 $X3_1$ 變更為 $X3'_1$

如圖 3，若原本的跨叢集頭 $X3_1$ 離開 MANETs 時，$R9_1$ 將執行下列步驟，以防止離開的叢集頭 $X3_1$ 繼續存取叢集 $X3$ 及其下層叢集資訊：

(1). $R9_1$ 從原本的叢集 $X3$ 中挑選一個新的跨叢集頭 $X3'_1$。

(2). $X3'_1$ 取得 $R9_1$ 公開的 V_{R9_1}，以及重新選取叢集 $X3$ 的群組金鑰 $K'_{Group-X3}$，並計算自己公開的 $V_{X3'_1} = sk_{X3'_1} \cdot B$，之後叢集 X3 產生新的安全過濾器：

$$f'_{X3}(x) = (x - sk_{X3'_1} \cdot V_{R9_1}) + K'_{Group-X3} (\text{mod } n)$$

(3). 叢集 $X3$ 的子叢集頭 $G6_1$ 及 $H7_1$ 選取新的群組金鑰後更新自己的過濾器：

$$f_{G6}(x) = (x - sk_{G6_1} \cdot V_{R9_1})(x - sk_{G6_1} \cdot V_{X3'_1}) + K'_{Group-G6} (\text{mod } n)$$

$$f_{H7}(x) = (x - sk_{H7_1} \cdot V_{R9_1})(x - sk_{H7_1} \cdot V_{X3'_1}) + K'_{Group-H7} (\text{mod } n)$$

(四) 變更叢集存取權限

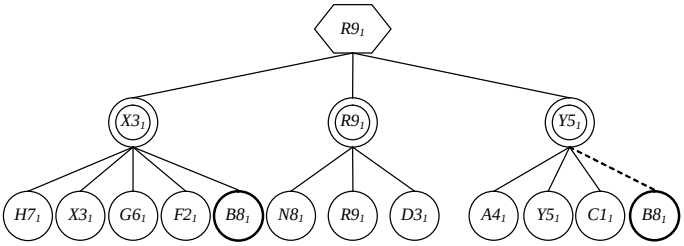

圖 4 變更 $B8$ 所在的叢集

以圖 4 為例，叢集頭 $B8_1$ 的虛線表示無法存取，為了防止叢集頭 $Y5_1$ 繼續存取叢集 $B8$ 內的資訊，故叢集頭 $B8_1$ 重新選取群組金鑰 $K'_{Group-B8}$，並更新過濾器：

$$f_{B8}(x) = (x - sk_{B8_1} \cdot V_{R9_1})(x - sk_{B8_1} \cdot V_{X3_1}) + K'_{Group-B8} (\text{mod } n)$$

除了改變叢集所在的叢集外，為因應一些現實應用中，需要提升及移除叢集頭的存取權限，故以下分別說明之。

1. 提升叢集頭存取之權限

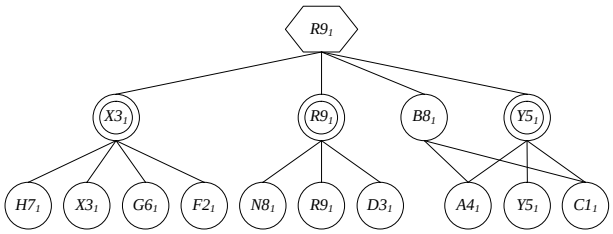

圖 5　提升叢集 *B8* 之存取權限

如圖 5 所示，若需要將原本位於叢集 *Y5* 下的 *B8* 提升權限，使其能存取叢集 *A4* 及 *C1*，且不再被叢集頭 $Y5_1$ 存取，則 $R9_1$ 將執行下列步驟：

(1). 叢集頭 $B8_1$ 重新選取群組金鑰 $K'_{Group-B8}$，並且更新自己的過濾器：

$$f_{B8}(x) = (x - sk_{B8_1} \cdot V_{R9_1}) + K'_{Group-B8} (\text{mod } n)$$

(2). 叢集頭 $A4_1$ 及 $C1_1$ 取得 $B8_1$ 公開的 V_{B8_1} 後，更新各自的過濾器：

$$f_{A4}(x) = (x - sk_{A4_1} \cdot V_{R9_1})(x - sk_{A4_1} \cdot V_{B8_1})$$
$$(x - sk_{A4_1} \cdot V_{Y5_1}) + K_{Group-A4} (\text{mod } n)$$
$$f_{C1}(x) = (x - sk_{C1_1} \cdot V_{R9_1})(x - sk_{C1_1} \cdot V_{B8_1})$$
$$(x - sk_{C1_1} \cdot V_{Y5_1}) + K_{Group-C1} (\text{mod } n)$$

2. 降低叢集頭存取之權限

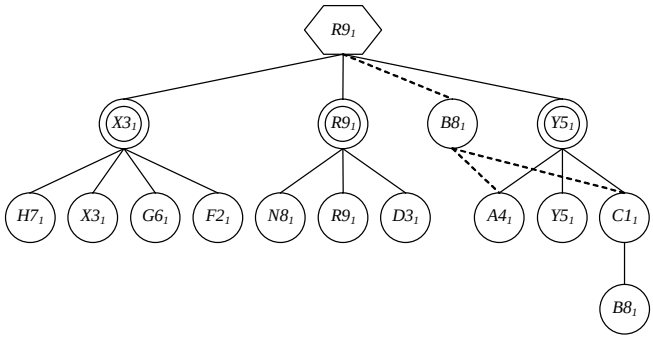

圖 6　將叢集 *B8* 的存取權限降低

以圖 6 為例，當叢集頭 $B8_1$ 的存取權限降低時，使其能被叢集 *C1* 存取，並防止下層的 $B8_1$ 能存取上層的叢集， $R9_1$ 將執行下列步驟：

(1). 叢集頭 $A4_1$ 及 $C1_1$ 重新選擇群組金鑰，然後各自更新過濾器：

$$f_{A4}(x) = (x - sk_{A4_1} \cdot V_{R9_1})(x - sk_{A4_1} \cdot V_{Y5_1})$$
$$+K'_{Group-A4} (\text{mod } n)$$
$$f_{C1}(x) = (x - sk_{C1_1} \cdot V_{R9_1})(x - sk_{C1_1} \cdot V_{Y5_1})$$
$$+K'_{Group-C1} (\text{mod } n)$$

(2). 叢集頭 $B8_1$ 重新選擇群組金鑰 $K'_{Group-B8}$ 後，更新過濾器：

$$f_{B8}(x) = (x - sk_{B8_1} \cdot V_{R9_1})(x - sk_{B8_1} \cdot V_{Y5_1})$$
$$(x - sk_{B8_1} \cdot V_{C1_1}) + K'_{Group-B8} (\text{mod } n)$$

由於 MANETs 動態的特性，知道群組金鑰的叢集頭會發生離開叢集或加入其他叢集等情況，因此必須確保該叢集頭只能存取權限內的資訊。下一節將分析本機制之安全性及效能。

四、安全性及效能分析

(一) 安全性分析

本研究所提出之機制，其安全性是基於橢圓曲線離散對數困難度 (elliptic curve discrete logarithm problem, ECDLP)。ECDLP 的定義為：在有限體 F_p 上選擇一個秩為 *n* 的基點 *B*，若另一點 $V = sk \cdot B$ (*sk* 為介於 1 到 *n*-1 的整數)，在知道 *V* 及 *B* 的情況下欲求得 *sk*，若 *n* 夠大時，其計算上為不可行。

當一個入侵者想存取叢集 *ai* 的資訊時，會因為沒有群組金鑰 $K_{Group-ai}$ 而無法正確的解密群組內的訊息。若入侵者想藉由叢集頭 ai_1 的過濾器 $f_{ai}(x)$ 計算出該叢集的群組金鑰 $K_{Group-ai}$，則會因為沒有對應的叢集頭私鑰 sk_{ai_1}，故而無法計算出正確的 $K_{Group-ai}$，此外，若入侵者想從叢集頭公

開的 V_{ai_1} 計算出 sk_{ai_1}，則必須先面對困難的 ECDLP，故可得知此機制能確保節點上的資訊不受惡意人士的存取。另外，當機制的叢集頭位置變動時，皆會造叢集頭存取權限的變動，為防止叢集頭存取不在自己權限內的資訊，以下將說明本研究所提之機制於第二節所描述的安全需求：

1. 新增叢集：當機制新增一個叢集 ai 時，該叢集的叢集頭 ai_1 會選出一個新的群組金鑰 $K'_{Group-ai}$，故上層的叢集頭雖可藉由過濾器計算出 $K'_{Group-ai}$，但卻無法取得 ai 之前的資訊，因此可滿足前向安全。
2. 刪除叢集：當機制刪除叢集 ai 時，叢集頭 ai_1 只要變更群組金鑰並更新過濾器，則可使舊有的存取權限失效，故可保持後向安全。
3. 變更叢集頭：當叢集頭 ai_1 離開網路時，機制將群組金鑰更改為 $K'_{Group-ai}$，以及重新選擇叢集頭 ai'_1，且相關的下層叢集也更新過濾器。對離開的叢集頭 ai_1 而言，因為群組金鑰已變更，所以無法再使用 $K_{Group-ai}$ 存取 ai_1 下層的叢集，故可達成前向安全。而對於新的叢集頭 ai'_1，由於不曾取得過 $K_{Group-ai}$，所以無法存取叢集 ai 的資訊，故可滿足後向安全。
4. 移動叢集頭：當叢集頭 ai_1 移動到另一個叢集時，ai_1 將群組金鑰變更為 $K'_{Group-ai}$ 並重新計算過濾器。因為上層叢集頭無法使用 $K'_{Group-ai}$ 存取 ai 之前的資訊，故可達成前向安全。另一方面，由於群組金鑰已變更，所以無法使用原本的 $K_{Group-ai}$ 存取 ai 後來的訊息，故可保證其後向安全。
5. 變更叢集頭權限：當叢集頭 ai_1 的存取權限變更時，ai_1 更新自己的群組金鑰 $K'_{Group-ai}$ 以及過濾器，因此不在權限內的叢集頭，無法再以自己的私鑰計算出叢集 ai 的群組金鑰 $K'_{Group-ai}$，因此無法取得 ai 後來的資訊，故可保持後向安全。另一方面，若 ai_1 的存取權限降低，相對的叢集也會更新群組金鑰以及過濾器，所以 ai_1 無法再以自己的私鑰存取不在自己權限內的叢集，故可滿足前向安全。

(二) 效能分析

Saxena 等人[12]使用基於雙變量多項式的祕密分享技術，提出一個適用於臨時性的 MANETs 節點許可協定稱為 BiAC，此協定之目的在於分享群組間的秘密以及建立安全的通訊。其特色與本機制之比較如表 3，值得一提的是，本機制不需要限制網路規模的大小。

BiAC 協定中有二個角色，分別為協助者(Sponsor, P_i)以及新加入的節點，此協定如圖 7 並簡述之。

$$P_{n+1} \rightarrow P_i : id_{n+1}, PK_{n+1}$$

$$P_i \rightarrow P_{n+1} : id_i, PK_i, E_{PK_{n+1}}(x_i(id_{n+1}))$$

圖 7　BiAC 協定

1. 新加入的節點 P_{n+1} 將自己的身份 id_{n+1} 及公鑰 PK_{n+1} 廣播到其他節點 P_i。

表 3　特色比較

項目	BiAC	本機制
公鑰系統	以身份為基礎的密碼系統	橢圓曲線密碼系統
祕密分享技術	雙變量多項式	安全過濾器
安全基礎	離散對數困難度	橢圓曲線離散對數困難度
認證中心	不需要	不需要
節點數量最小限制	須大於 $2t-1$	無限制
註：t 為祕密分享機制的門檻值		

2. 每個 P_i 確認 id_{n+1} 後，使用自己的多項式計算 P_{n+1} 所需的參數 $x_i(id_{n+1}) = f(id_{n+1}, id_i)$，並使用 PK_{n+1} 加密後連同 id_i 以及 PK_i 回傳到 P_{n+1}。
3. 當 P_{n+1} 收到大於門檻值的回應訊息後，使用自己的密鑰解密訊息取得每一個參數 $x_i(id_{n+1})$，即可計算出自己的多項式。若要進行通訊時，藉由多項式可推導出相同的密鑰。

在 BiAC 協定中，其計算複雜度取於設定的門檻值 t，公鑰加密需要 $O(t)$模指數運算，解密需要一次模指數運算。在建構分享的多項式時，P_i 需要執行 $O(t)$模乘法計算，而 P_{n+1} 需要執行 $O(t_3)$模乘法計算。通訊成本方面，若節點的身份(identity)長度為 $\log q$ 位元，公鑰長度為 $\log p$ 位元，則其節點加入請求 JOIN_REQ 的訊息長度為 $t\log q + t\log p$ 位元，而其回應 JOIN_RLY 的訊息長度為 $2t\log q + t\log p$ 位元，故總計為 $3t\log q + 2t\log p$。

而在本機制中，祕密分享技術是基於安全過濾器，在計算過濾器函數所需之時間與繼承的叢集頭數量 k 成比例，因此需要執行 $O(k)$模運算以及 $O(k)$ ECC 點乘法計算。而通訊成本方面，由於建構過濾器需要 k 個公開的 V_{ai_1}，所以其通訊成本為 $k\log p$，另外，在計算群組金鑰時需取得過濾器及對方的 V_{ai_1}，其通訊成本為 $(k+1)\log p$，故總通訊成本為 $(2k+1)\log p$。本研究與 BiAC 協定之效能比較如表 4。

對離開的節點而言，不必額外更新多項式，只需要刪除相關參數，因此運算量皆為零。另外，當節點加入時，由於模指數計算遠大於 ECC 點乘法及模乘法計算，因此只有 k 值遠大於 t 值的情況發生，本機制之計算複雜度才會高於 BiAC 協定。由於本機制為階層叢集式架構，若 k 值過大時，可將不必要之叢集移除，或者將節點數量稀少的叢集整合，藉以簡化階層，進而減少 k 值大小，所以 k 值遠大於 t 值的可能性較小。因此在一般的情況下，本機制之計算複雜度將低於 BiAC 協定。

五、結論與未來發展方向

(一) 結論

在資訊發達的現今，以網路存取資訊已成為一種常態。本研究結合有效率的橢圓曲線密碼系統及安全的過濾器技術，於階層式叢集的架構下，提出動態群組存取控制機制能。總結來說，本機制能達成以下優點：

1. 由於上層叢集頭由於不需要反覆推算，故可以快速地取得下層叢集的群組金鑰。
2. 當新增及刪除叢集時，本機制不需要變動網路中所有的金鑰，即可良好的完成目的。
3. 當各叢集變動群組金鑰後，只需要更新自己的過濾器，且不影響上層叢集頭存取該叢集的能力。

表 4　機制效能比較

項目			BiAC	本機制
計算複雜度	加入	模指數計算	$O(t)$	0
		模乘法計算	$O(t) + O(t^3)$	$O(k)$
		模加法計算	0	1
		ECC 點乘法計算	0	$O(k)$
	離開	運算量	0	0
通訊成本			$3t\log q + 2t\log p$	$(2k+1)\log p$

4. 本機制使用過濾器技術取代門檻式祕密分享技術，因此沒有 CA 及門檻值的限制。另外，基於效能及安全的觀點，本研究基於 ECC 所提之機制，對 MANETs 群組存取控制的安全而言具有實際的助益。

(二) 未來發展方向

本機制所提出的動態群組存取控制，能確實的防止沒有權限的叢集頭存取資訊。但對於內部之節點可能出現洩密者，換句話說，原本合法之節點可能將群組資訊外洩。雖然本機制能將損害控制在有權限制之節點上，在發生資訊外流後能根據存取權限，鎖定可能的洩密成員，但卻無法主動追蹤洩密成員，因此在未來發展部份，若能主動將洩密成員逐出群組，以制止成員持續洩密群組內的機密，是相當值得後續研究者深入研究。

六、參考文獻

[1] C. C. Chang, I. C. Lin, H. M. Tsai, and H. H. Wang, “A key assignment scheme for controlling access in partially ordered user hierarchies”, Proceedings of the 18th IEEE International Conference on Advanced Information Networking and Applications, pp.376-379, 2004.

[2] J. S. Chou, Y. L. Chen, and T. H. Chen, “An efficient session key generation for NTDR networks based on bilinear paring”, Computer Communications, Vol. 31, No. 14, pp.3113-3123, 2008.

[3] Y. F. Chung, H. H. Lee, F. P. Lai, and T. S. Chen, “Access control in user hierarchy based on elliptic curve cryptosystem”, Information Sciences, Vol. 178, pp.230-243, 2008.

[4] M. S. Hwang and W. P. Yang, “Controlling access in large partially-ordered hierarchies using cryptographic keys”, Journal of Systems and Software, Vol. 67, No. 2, pp.99-107, 2003.

[5] S. Jarecki, N. Saxena, and J. H. Yi. “An attack on the proactive RSA signature scheme in the URSA ad doc network access control protocol”, Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks, pp.1-9, 2004.

[6] F. G. Jeng and C. M. Wang, “An efficient key-management scheme for hierarchical access control based on elliptic curve cryptosystem”, Journal of Systems and Software, Vol. 79, pp.1161-1167, 2006.

[7] Y. Kim, D. Mazzocchi, and G. Tsudik, “Admission control in peer groups”, Proceedings of Second IEEE International Symposium on Network Computing and Applications, pp.131-139, 2003.

[8] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, “Adaptive security for multilevel ad hoc networks”, Wireless Communications and Mobile Computing, Vol. 2, No. 5, pp.533-547, 2002.

[9] F. Lu and T. Zhou, “Research on identity-based cluster access control model with dynamic trust agent for mobile ad hoc networks”, Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing , pp.1-5, 2006.

[10] H. Y. Luo, J. J. Kong, P. Zerfos, S. W. Lu, and L. X. Zhang, “URSA: ubiquitous and robust access control for mobile ad Hoc networks”, IEEE-ACM Transactions on Networking, Vol. 12, No. 6, pp.1049-1063, 2004.

[11] N. Saxena, G. Tsudik, and J. H. Yi, “Identity-based access control for ad hoc groups”, Lecture Notes in Computer Science, Vol. 3056, pp.326- 379, 2005.

[12] N. Saxena, G. Tsudik, and J. H. Yi, “Efficient node admission and certificateless secure communication in short-lived MANETs”, IEEE Transactions on Parallel and Distributed Systems, Vol. 20, No. 2, pp.158-170, 2009.

[13] W. J. Tsaur, “Several security schemes constructed using ECC-based self-certified public key cryptosystems”, Applied Mathematics and Computation, Vol. 168, No. 1, pp.447-464. 2005.

[14] W. J. Tsaur, and H. T. Pai, “Dynamic key

management schemes for secure group communication based on hierarchical Clustering in mobile ad hoc networks", Lecture Notes in Computer Science, Vol. 4743, pp.475-484. 2007.

[15] W. G. Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy", IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 1, pp.182-188, 2002.

[16] N. C. Wang and S. Z. Fang, "A hierarchical key management scheme for secure group communications in mobile ad hoc networks", Journal of Systems and Software, Vol. 80, pp.1667-1677, 2007.

[17] J. Wu and I. Stojmenovic, "Ad hoc networks", Computer, Vol. 37, No. 2, pp.29-31, 2004.

[18] K. P. Wu, S. J. Ruan, C. K. Tseng, and F. P. Lai, "Hierarchical access control using the secure filter", IEICE Transactions on Information & Systems, Vol. E84-D, No. 6, pp.700-707, 2001.

[19] B. Wu, J. Wu, E. B. Fernandez, M. Ilyas, and S. Magliveras, "Secure and efficient key management in mobile ad hoc networks", Journal of Network and Computer Applications, Vol. 30, No. 3, pp.937-954, 2007.

[20] J. Wu and R. Wei, "An access control scheme for partially ordered set hierarchy with provable security", Lecture Notes in Computer Science, Vol. 3897, pp.221-232, 2006.

[21] H. Yang, H. Y. Luo, F. Ye, S. W. Lu, and L. X. Zhang, "Security in mobile ad hoc networks: challenges and solutions", IEEE Wireless Communications, Vol. 11, No. 1, pp.38-47, 2004.

[22] J. H. Yeh, "A secure time-bound hierarchical key assignment scheme based on RSA public key cryptosystem", Information Processing Letters, Vol. 105, pp.117-120, 2008.

[23] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks", IEEE Communications Surveys & Tutorials, Vol. 7, No. 1, pp.32-48, 2005.

[24] Y. Zhang, W. Liu, W. Lou, & Y. Fang, "Securing mobile ad hoc networks with certificateless public keys", IEEE Transactions on Dependable and Secure Computing, Vol. 3, No. 4, pp.386-399, 2006.

[25] L. Zhou and Z. J. Haas, "Securing ad hoc networks", IEEE Network, Vol. 13, No. 6, pp.24-30, 1999.