國 立 交 通 大 學

資訊科學與工程研究所

碩 士 論 文

Halo:適用於同儕網路中機會性合作的階層式身份基礎公鑰系統

Halo: A Hierarchical Identity-Based Public Key Infrastructure for Peer-to-Peer Opportunistic Collaboration

研 究 生：曾輔國

指導教授：邵家健 博士

陳榮傑 博士

中 華 民 國 九 十 七 年 六 月

Halo:適用於同儕網路中機會性合作的階層式身份基礎公鑰系統

Halo: A Hierarchical Identity-Based Public Key Infrastructure for Peer-to-Peer Opportunistic Collaboration

研 究 生：曾輔國　　Student：Fu-Kuo Tseng

指導教授：邵家健 博士　　Advisor：Dr. John Kar-Kin Zao

陳榮傑 博士　　Dr. Rong-Jaye Chen

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master in Computer Science

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

Halo:適用於同儕網路中機會性合作的階層式身份基礎公鑰系統

學生：曾輔國　　指導教授：邵家健 博士
陳榮傑 博士

國立交通大學資訊科學與工程研究所　碩士班

摘要

同儕網路系統中，因缺乏資訊安全保護，所以無法使其中的服務常常缺乏安全性與強健性。這個問題來自無伺服器 (server-less) 的架構和 ad-hoc 方式操作情境的需求。同時也造成同儕網路無法使用傳統的對稱/不對稱的密碼學技術。我們發現：階層式的身份基礎密碼系統和分散式金鑰產生機制提供了可行的解決之道。在本論文中，我們將提出 Halo 我們設計的一個階層式的身份基礎公開基礎架構。它用到一些新穎的技術來達到遞迴式的 private key generators (PKG) 和建立一個無限層數 (unlimited number) 限制的信任階層。因此，這個公開金鑰基礎架構不僅能部屬階層式身份基礎加密、簽章、簽密 (signcryption) ，再加上成對的 (pair-wise) 認證式金鑰協定 (authenticated key agreement) 來保護同儕網路中的應用。

Halo: A Hierarchical Identity-Based Public Key Infrastructure for Peer-to-Peer Opportunistic Collaboration

Student: Fu-Kuo Tseng

Advisor: Dr. John Kar-Kin Zao

Dr. Rong-Jaye Chen

Institute of Computer Science and Engineering
National Chiao Tung University

ABSTRACT

In peer-to-peer network system, services usually fail to provide security and robustness due to the absence of information security. The security weakness is rooted in the server-less architecture and demand for ad-hoc operation scenario. Besides, they also stop us from using scalable key management by traditional symmetric/ asymmetric cryptographic techniques.. We have found that hierarchical identity-based cryptography and distributed key generation scheme provide us possible solution to this problem. In this paper, we present the design of Halo, a hierarchical identity-based public key infrastructure that uses these novel technologies to perform recursive instantiation of private key generators and establish a trust hierarchy with unlimited number of levels. Therefore, The PKI thus enables the employment of hierarchical identity-based public key encryption, signature, and signcryption operation. In addition, pair-wise authenticated key agreement also provided for the protection of peer-to-peer applications.

誌 謝

本論文能順利的完成，首先要先感謝　邵家健老師和 陳榮傑老師的指導，不論是在為學或是做人處事方面，兩位老師都給予我莫大協助與啟發。另外，感謝張仁俊教授、曹孝櫟教授及胡鈞祥博士擔任畢業口試委員，給予我許多寶貴的意見，讓本論文更加完備。

此外，特別感謝時常協助我的兩位同學：芳伯與用翔，芳伯在邵老師的指導下，對於同儕網路的相關知識非常的熟稔，在系統方面的知識與實作能力更是令我欽佩。用翔在陳老師的指導下，對於橢圓曲線密碼系統及其背的數學理論學有專精，兩位對於輔國的研究助益良多，在此致上誠摯的謝意。

接著感謝 Cryptanalysis Lab 的成員：志賢學長、定宇學長、順隆學長、佩娟與用翔與 Pervasive Embedded Technology Lab 的成員：國晉學長、芳伯、哲民，學弟們：星閔、嘉錡、彥霖、博政、勝焜與學妹梅瑛，大家一起運動、吃飯、聊天及討論想法，是我研究生活中，不可或缺的的一環，謝謝你(妳)們。

此外，感謝系上的桌球隊，校園網路策進會成員與諮商中心志工團的眾多伙伴，有你(妳)們的陪伴，讓我的研究生活更加的健康與充實。另外還有系上的學弟妹，讓我的研究生活增添活力與色彩。在此，我想特別感謝敬達、藝睿、逸朴、宗霖、宜君與書平。

最後，感謝永遠支持我的家人，有你們做為我的後盾，讓我研究的路更加的平順與穩健。感謝我的女友郁欣，陪伴我渡過許多重要的時刻，與我分享種種的心情。謝謝在研究過程中協助我的人，也許是一句鼓勵的話、一個特別的啟發或是一個及時的幫助，你(妳)們都是這篇論文重要的推手，感謝你(妳)們。

謹以本論文獻給所有關心我的家人、師長及朋友，謝謝你(妳)們。

目錄

圖目錄

表格目錄

第 1 章 綜論

同儕網路應用已超越其最初檔案分享的用途，變成一個分散式計算的典型。除了廣受歡迎的同儕式 VoIP 服務：Skype[16]外，交通監控服務[17][26]、永久儲存系統[8][10]和視訊串流程式[13][20][21]等，都已證明有顯著的市場潛力。其中許多的程式採用了機會性合作 (opportunistic collaboration) 的操作模式：網路中的節點因擁有部分的機能或是資源，而被招募並組成 ad-hoc 社群，結合大家手邊的資源與服務，合力完成某項工作。通常這些被招募的同儕，先前都沒有接觸過，所以很難有協議的秘密。但是，在合作的過程，這些同儕或許會需要交換一些敏感的資訊，因而得到更好的合作效果，所以，這些在同儕間的機會性合作中，應當存在適當的安全保護機制。然而，同儕網路應用中缺乏安全的基礎架構，如固定存在的伺服器，以及同儕經常移動、變換領域的特性，都讓安全通訊所必要的服務，如：同儕認證、訊息完整、訊息祕密...等，更加的難以達成。

1.1 問題陳述

一、 在同儕網路環境裡，使用公開金鑰系統時，我們想要增強系統金鑰的安全性，避免單一同儕掌握完整系統金鑰，也不希望有可信任莊家的存在，希望能夠將信任分散給多個實體來實現。

二、 在同儕網路環境裡，使用公開金鑰系統時，我們想要增強系統權威的的可利用性，避免因權威的移動或是離開，造成同儕無法進行認證，希望能夠將資訊複製多份，儲存於網路系統中。

三、 在同儕網路環境裡，使用公開金鑰系統時，在不增加太多計算複雜度的前提下，我們想要降低系統的管理複雜度。希望能夠避免複雜的分散式憑證產生與困難的憑證管理。

1.2 研究方法

本論文將提出*Halo*，一個階層式的身份基礎公開金鑰架構 (Hierarchical IDentity-based Public Key Infrastructure, IDPKI) ，用來支援網路上同儕間的機會性合作。*Halo*將會發給每個使用者一把私鑰，而對應私鑰的公鑰則對應於使用者的身份和使用者被授權提供或使用的服務。這些公/私鑰對能夠用在 Gentry-Silverberg [3] 的階層式的身份基礎公鑰系統加密法 (hierarchical identity-based encryption scheme, HIBE) 、簽章法 (hierarchical identity- based signature scheme, HIBS) 和 Sherman Chow 的簽密 (hierarchical identity-based signcryption scheme, HIBES) [15]。這些密碼學的運算法為實作安全通訊服務中，最基本的機制。

Halo 特別設計用於無伺服器的環境。允許新的同儕加入和管理行政領域 (peer administrative domains) 的工作是由一群同儕，而非特別指定的同儕來管理。它們能夠執行分散式金鑰產生的動作。*Halo* 同時也具備漸近式的階層成長和自我管理的能力。從數個兩兩互相相信的可信任同儕 (trusted peer) 開始，*Halo* 能夠允許新的同儕加入，產生新的同儕領域和產生更多可相信的同儕，而這些同儕能夠做為 private key generators (PKG) 。行政領域的管理者執行安全政策 (security policy) ，同儕需向他們提出足夠的證明，而被允許新的身份或使用新的服務。我們將設計的系統命名為 *Halo*，是期望連結同儕的身份與他們被授與服務的公鑰，能夠用來認證使用者，一旦加入了 *Halo* 同儕領域，即能夠使用其提供的安全服務，就如同榮耀的光環能夠分辨聖者與凡人。

1.3 論文大綱

接下來的文章將分成八章節：第二章說明最基本的知識背景，包括橢圓曲線密碼系統、同儕網路系統、身份基礎公開金鑰系統及其變形、階層式身候基礎公開金

鑰系統及其變形、門檻式加密法及分散式金鑰產生協議。第三章說明系統設計的原則，與操作設計的原則。第四章開始回顧在 *Halo* 上使用的密碼學技術：Gentry、Silverberg 的階層式身份基礎加密(HIBE)/簽章演算法(HIBS)、改良式的 Shamir 秘密分享方式(Shamir secret sharing, SSS)以及 Tang 的分散式金鑰產生方式(Tang's distributed key generation, TC-DKG)。這個章節主要展示如何將密碼學上的技術，整合進 Gentry-Silverberg 的設計中，第五章將討論 *Halo* 系統初始過程及自我增長與維護的方式，另外也討論一個合法的同儕，在 *Halo* 系統中的生命史。第六章將討論另一密碼系統：門檻式 RSA (threshold RSA or TRSA) ，我們將 TRSA 原先兩層的設計，延伸應用於多層次的同儕網路系統，並與我們設計的系統比較。第七章將討論如何實作 *Halo* 成為 JXTA 的服務模組，分成密碼學方面與系統設計方面。最後一章總結我們的貢獻和未來可行的方向。

第 2 章 背景知識

2.1 橢圓曲線密碼學

橢圓曲線密碼系統是一種公開金鑰系統，利用基礎於有限場 (finite field) 的橢圓曲線，其通用格式為 $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ ，也稱為通用 Weierstrass 方程式 （generalized Weierstrass equation） ，滿足曲線方程的所有點及一個無窮遠點 (point at infinity) 所成的集合，形成一個加法群 (additive group) 。橢圓曲線上的點能夠進行兩點間的加法，以幾合的角度來看，我們想算出通過這兩點的直線，並與橢圓曲線相交於第三點，再將此點對 x 鏡射，即可得到結果。顯示各種可能的加法，P+Q=R，若 P 不等於 Q,，P=Q 及 P=-Q 三種情形，若橢圓曲線上的三點共線，則三點相加的和即為無窮遠點。

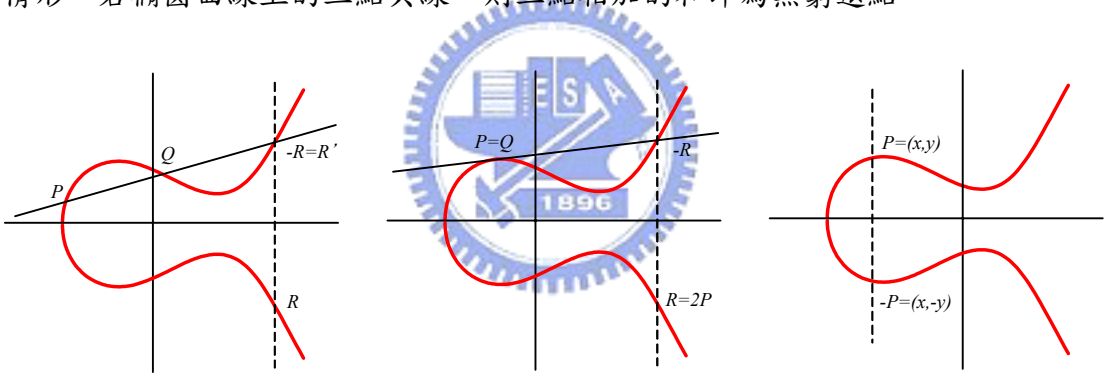

圖 1：橢圓曲線加法運算

假設 $p > 3$ 是一個質數，基礎於大小為 p 的有限體之橢圓曲線可以記為 $GF(p)$ ，其方程式寫做 $y^2 = x^3 + ax + b$ ，其中 $a,b \in GF(p)$ 且 $4a^3 + 27b^2 \neq 0 \pmod p$ ，則存在一群 (x,y) 的有序對，其 x , y 皆屬於 $GF(p)$ 且滿足曲線方程式，加上定義的無窮遠點，這些點形成可交換群(abelian group)。假設欲相加在曲線四的兩點：$P = (x_p, y_p)$ 和 $Q = (x_Q, y_Q)$ ，首先我們先計算通過 P 、Q 兩點直線的斜率 λ ，則 $P + Q = (x_{P+Q}, y_{P+Q})$ 為：

$$x_{P+Q} = \lambda^2 - x_p - x_Q$$
$$y_{P+Q} = \lambda(x_P - x_{P+Q}) - y_P$$
，其中：
$$\begin{cases} P \neq Q,\ \lambda = \dfrac{y_Q - y_P}{x_Q - x_P} \\ P = Q,\ \lambda = \dfrac{2x_P^2 + a}{2y_P} \end{cases}$$

在群中的無窮遠點O扮演加法單位元素的角色，對於任意在群中的點P，$P+O=O+P=P$，每個任意點P也有一個唯一的反元素$-P$，使得$P+(-P)=O$，在基礎於$GF(p)$之橢圓曲線上的點P，而反元素記為$-P=(x_P,-y_P)$。

另一種基礎於大小為2^n的有限體之橢圓曲線可以記為$GF(2^n)$，其方程式寫做$y^2+xy=x^3+ax^2+b$，其中$a,b\in GF(2^n)$且$b\neq 0$。則存在一群(x,y)的有序對，其x,y皆屬於$GF(2^n)$且滿足曲線方程式，加上定義的無窮遠點，這些點形成可交換群。假設欲相加在曲線四的兩點：$P=(x_p,y_p)$和$Q=(x_Q,y_Q)$，首先我們先計算通過P、Q兩點直線的斜率λ，則$P+Q=(x_{P+Q},y_{P+Q})$為：

$$x_{P+Q} = \lambda^2 + \lambda + x_P + x_Q + a$$
$$y_{P+Q} = \lambda(x_P + x_{P+Q}) + x_{P+Q} + y_P$$
，其中：
$$\begin{cases} P \neq Q,\ \lambda = \dfrac{y_Q + y_P}{x_Q + x_P} \\ P = Q,\ \lambda = x_P + \dfrac{y_P}{x_P} \end{cases}$$

對於任意在群中的點P，都有一個唯一的反元素$-P$，使得$P+(-P)=O$，在基礎於$GF(2^n)$之橢圓曲線上的點P，而反元素記為$-P=(x_P,x_P+y_P)$。不論是基礎於何種有限體，在運算時（加法、減、乘法、除法、反元素計算），需在相關的有限體進行。

在橢圓曲線中，群中的運算為加法而不存在乘法，所以在乘法群中的指數運算能夠對應到橢圓曲線加法群中的純量乘法(scalar multiplication)，我們將自加k次的點P記為$k\cdot P$，若$k\cdot P=O$，則k稱之為P的級數（order），我們會找一個級數大於2^{160}的基點（base point）並利用純量乘法的運算來建構橢圓曲線密碼系統的基礎。

橢圓曲線密碼系統基礎於一些難題：假設 G 是一個橢圓曲線加法群，是由 $P \in E(F_q)$ 產生出來並且 P 的 order 為 q，Q 為曲線上除了 P 以外的點。

- 橢圓曲線離對數問題 (ECDLP) ：

  鑄定 P 、$Q \in G$，求得一個整數 $a \in \mathrm{Z}_q^*$ 滿足 Q=aP

- 計算 Diffie-Hellman 問題 (CDHP) ：

  給定 $P, aP, bP \in G$，$a, b \in \mathrm{Z}_q^*$，求得 abP

- 決定 Diffie-Hellman 問題 (DDHP) ：

  給定 $P, aP, bP, cP \in G, a, b, c \in \mathrm{Z}_q^*$, 決定是否 $c = ab$

2.1.1 Divisor 理論[27][28]

假設 f 是一個非零的有理函數（rational function），而且點 $P \in E$，若 $f(P) = 0$ 則說 f 在 P 點有個 zero；若 $f(P) = \infty$ 則說 f 在 P 點有個 pole，而 divisor 就是用來了解有理函數的 zeros 和 poles，同時也能提供在橢圓曲線上的有理函數，其上發生 zero 或是 pole 的點和這些點的次數（order），divisor D 可以定義為橢圓曲線 E 上點的 formal sum：$D = \sum_{p \in E} n_p(P)$，其中 n_p 皮表點 P 的 zero 或 pole 的特性與其對應的次數。若 $n_p > 0$ 則代表點 P 為 zero，若 $n_p < 0$，則代表點 P 為 pole。例如：若 $P, Q, R \in E$，$D_1 = 1(P) + 2(Q) - 5(R)$ 代表 divisor 在 P, Q 有 zero，且在 R 有 pole，P, Q, R 的次數分別為 $1, 2, 5$。$D_2 = 7(P) + 2(-2P) - 2(O)$ 則是代表在 $P, -2P$ 有 zero，且在 O 有 pole，$P, -2P, O$ 的次數分別為 $7, 2, 2$。要注意的是，括號是用來分開次數與特定點，例如 $2(P)$ 指的是在點 P 有個 zero 且次數為 2，而 $(2P)$ 指的是在點 $2P$ 有個 zero 且次數為 1

在 E 上的一群divisors組成可交換群，記為 $Div(E)$，並有下列的加法特性：若 $D_1, D_2 \in Div(E)$，且 $D_1 = \sum_{p\in E} n_p(P)$，$D_2 = \sum_{p\in E} m_p(P)$，則 $D_1 + D_2 = \sum_{p\in E} n_p(P) + \sum_{p\in E} m_p(P) = \sum_{p\in E} (n_p + m_p)(P)$。對於一個divisor $D = \sum_{p\in E} n_p(P)$，我們定義 $supp(D) = \{P \in E \mid n_p \neq 0\}$ 為divisor D 的support，$deg(D) = \sum_{p\in E} n_p$ 為divisor D 的次數。例如：$D_1 = 1(P) + 2(Q) - 5(R)$、$D_2 = 7(P) + 2(-2P) - 2(O)$ 則 $supp(D_1) = \{P, Q, R\}$，$supp(D_1) = \{P, -2P, O\}$ 且 $deg(D_1) = 1 + (-2) + 5 = 4$、$deg(D_2) = 7 + 2 + (-2) = 7$。

接下來我們討論次數為 0 的 divisor 集合，記為 $Div^0(E)$，假設 f 是一圈有理函數，從 $K \times K \to K$，其中 K 為有限場。例如：$f(x,y) = \frac{5y - 2x - 4}{7y + 2x - 2}$，我們定義在點 P 時，有理函式值的算法為 $f(P) = f(x_P, y_P)$，在 divisor D 時，有理函式值的算法為 $f(D) = \prod_{P\in supp(D)} f(P)^{n_P}$。定義有理函式 f 的 divisor 為 $div(f) = \sum_{P\in E} n_{P,f}(P)$，其中 $n_{P,f}$ 為點 P 在 f 上的 zero 或是 pole 的 order。定理證明[27]有理函式 f 的 divisor，其次數必需要為 0。換句話說，對任意的有理函式而言，$div(f) \in Div^0(E)$。所以對於兩個有理函式 f_1, f_2 而言，有列的特性：$div(f_1) + div(f_2) = div(f_1 f_2)$ 且 $div(f_1) - div(f_2) = div(f_1 / f_2)$。

一個 divisor $D \in Div^0(E)$ 若存在有理函式 f，使得 $D = Div(f)$，稱之為 *principal*。一個 principal divisor $D = \sum_{p\in E} n_p(P)$ 有著 $\sum_{p\in E} n_p P = O$ 的特性。其中 $\sum_{p\in E} n_p P$ 代表橢圓曲上點的加法。例如，若 $D_3 = P + (-P) - 2(O)$，因 D_3 滿足 $deg(D_3) = O$，且 $P + (-P) - 2O = P - P = O$，所以 D_3 是 principal。

假設有兩個 divisor D_1 和 $D_2 \in Div^0(E)$，若 $D_1 - D_2$ 是 principal，則稱 D_1, D_2 等價，記為 $D_1 \sim D_2$，對任意的 divisor 而言，$D = \sum_{p\in E} n_p(P) \in Div^0(E)$，存在唯一的一

點 $P=\sum_{p\in E} n_p P \in E$ 使得 $D \sim (P)-(O)$。換句話說，D 永遠可以寫成 canonical 形式：$D=(P)-(O)+div(f)$，其中 f 為有理函式。

最後要介紹的是如何相加兩個 canonical 形式的 divisor，使得結果任為 canonical 形式。這個過程提供了一個在給定 divisor D 的悄況下，找尋有理函數 f 的方法，使得 $div(f)=D$。假設 D_1，$D_2 \in Div(E)$，且 $D_1=(P_1)-(O)+div(f_1)$、$D_2=(P_2)-(O)+div(f_2)$，$P_1+P_2=P_3$，假設 $h_{P_1,P_2}(x,y)=ay+bx+c$ 為通過 P_1 或 P_2 的直線，$h_{P_3}(x,y)=x+d$ 為垂直通過 P_3 的直線。接著我們有 $div(h_{P_1,P_2})=(P_1)+(P_2)+(-P_3)-3(O)$，因為 $P_1, P_2, -P_3$ 在直線 $h_{P_1,P_2}(x,y)$ 上，所以為 zeros，而 $div(h_{P_3})=(P_3)+(-P_3)-2(O)$，因為 P_3 和 $-P_3$ 在直線 h_{P_3}。所以 divisor D_1, D_2 的加法寫做：

$$\begin{aligned} D_1+D_2 &= (P_1)+(P_2)-2(O)+div(f_1f_2) \\ &= (P_3)-(O)+div(f_1f_2)+div(h_{P_1,P_2})-div(h_{P_3}) \\ &= (P_3)-(O)+div(f_1f_2h_{P_1,P_2}/h_{P_3}) \end{aligned}$$

上式將用於 Weil pairing 的計算。

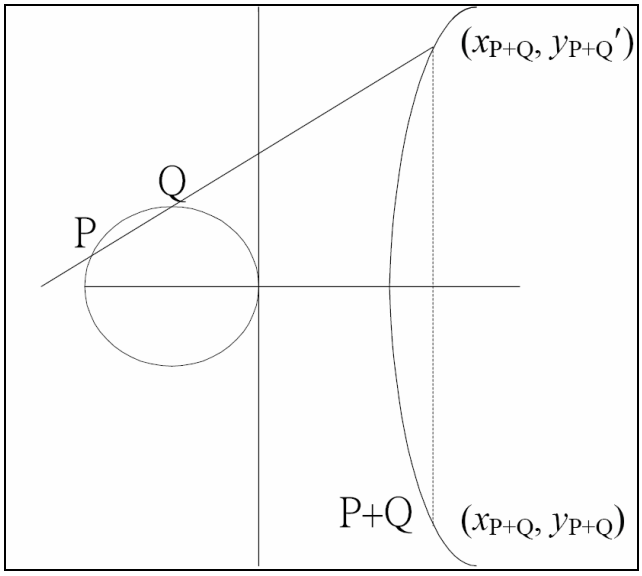

圖 2：橢圓曲線的加法律

2.1.2 Weil Pairing 和 Miller 演算法

Weil 對為一線性映射函數，由一個群對應到另一群，而定義域 (G_1) 通常為定義在大質數上的加法群，群上的點都符合橢圓曲線方程式，我們假設其生成元素 (generator) 為 P，而對應域 (G_2) 通常為定義在大質數上的乘法群。我們假設在這 G_1 和 G_2 的離散對數問題是困難的。

給定一個基礎於有限體 K 的橢圓曲線 E，假設 m 和 k 的 characteristic $char(K)$ [29]互質。例如：$char(GF(p)) = p$、$char(GF(2^n)) = 2$。Weil pairing 為一個函數對應關係：

$$e : E[m] \times E[m] \rightarrow U_m$$

其中 $E[m] = \{P \mid mP = O, P \in E\}$，稱為 m-torsion 群，$U_m$ 是在 $\overline{K}$ 中 1 的 m 次方根，而 $\overline{K}$ 為 K 的 algebraic closure[29]。Weil pairing $e(P,Q)$ 的計算定義如下：給定兩個 m-torsion 的點 P，$Q \in E[m]$，存在 divisor $D_P, D_Q \in Div^0(E)$，使得 $D_P \sim (P) - (O)$，$D_Q \sim (Q) - (O)$，任意選擇兩點 T 和 U，並指定 $D_P = (P + T) - (T)$、$D_Q \sim (Q) - (O)$。因為 $mP = mQ = O$，概據定理，mD_P 及 mD_Q 為 principal divisor，且存在有理函數 f_P 和 f_Q，其中 $div(f_P) = mD_P$，$div(f_Q) = mD_Q$，假設 D_P 和 D_Q 有完全不一樣的 support，也就是說 $supp(D_P) \bigcap supp(D_Q) = \phi$，則 $e(P,Q) = \dfrac{f_P(D_Q)}{f_Q(D_P)}$。■

第一個計算 $\hat{e}(P,Q)$ 的演算法為 Miller 演算法[30]，我們將步驟說明如下：

輸入：$P, Q \in E[m]$

輸出：$e(P,Q)$

步驟一：任意選擇兩點 T 和 U，滿足 $P+T, T, Q+U, U$ 皆不同，並指定

$D_P = (P+T)-(T)$ 、$D_Q \sim (Q)-(O)$

步驟二：計算 $f_P(Q+U), f_P(U), f_Q(P+T), f_Q(T)$，其中 f_P, f_Q 滿足 $div(f_P) = mD_P$ ，$div(f_Q) = mD_Q$

步驟三：計算 $e(P,Q) = \dfrac{f_P(D_Q)}{f_Q(D_P)} = \dfrac{f_P(Q+U)f_Q(T)}{f_Q(P+T)f_P(U)}$ ■

我們列舉 Weil 雙線性映射函數 $\hat{e} : G_1 \times G_1 \to G_2$ ，滿足下列特性：

一、 雙線性 (Bilinearity) ：$\forall P,Q,R \in G_1, \hat{e}(P+R,Q) = \hat{e}(P,Q)\hat{e}(R,Q)$ 且 $\hat{e}(P+Q,R) = \hat{e}(P,R)\hat{e}(Q,R)$

二、 不退化性 (Non-degeneracy) ： $\exists P,Q \in G_1$， $\hat{e}(P,Q) \neq 1$

三、 可計算性 (Computability) ：$\forall P,Q \in G_1$,計算 $\hat{e}(P,Q)^{ab}$ 是多項式時間 (Polynomial Time) 可完成。

以下說明雙線性密碼系統所基硬的難題：

- 雙線性 Diffie-Hellman 問題 (BDHP) ：

給定 P, aP, bP, cP， $a,b,c \in Z_p^*$， 計算出 $d \in G_2$ ，使得 $d = \hat{e}(P,P)^{abc} \in G_2$

2.2 身份基礎公開金鑰密碼學

Identity Based Encryption (IBE) 密碼系統的構想最初是由 Shamir[1]提出。它利用使用者的身份作為公鑰，而非傳統以無意義的字串。因此，公鑰能夠由使用者的身分資訊來導出，所以不需要公鑰憑證驗證。請見圖 3。使用者先向私鑰產生中心 (Private Key Generator, PKG) 認證自己，過程中可能會需要現身在註冊櫃台或是提出相關的證明文件，一但認證通過，私鑰產生中心即會產生對應於使用者

身份的私鑰。我們假設寄送著是 Alice，收件者是 Bob，Alice 藉由 Bob 公開的資訊 (圖中例子為電子郵件信箱或是手機號碼) 導出 Bob 的公鑰，Alice 藉由此公鑰加密文件給 Bob，而 Bob 可以事先或是收到密文後，向 PKG 認證並取得對應於自己身份的私鑰，如此就能夠解開密文。而簽章的過程也是類似，只是過程中使用的的是簽章和認證。

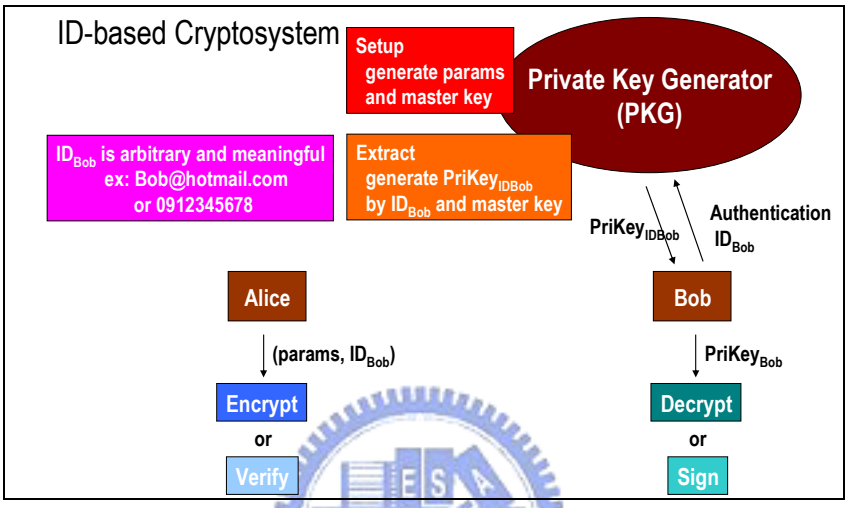

圖 3：IBE 運作圖

2.2.1 身份基礎加密系統設計

直到 2001 年，Cock[22]提出了基礎於 quadratic residuosity 難題的 IBE，但其最大的問題在於一次僅能加密一個位元，若使用長度為 l 的模數，則密文會是明文的 l 倍，所以僅適合小量的資料或是 session key 的保護。同年 Boneh 和 Franklin[5]個提出更有效率的 IBE 設計，此設計基礎於 BDH 和 CDH 問題上，他們設計的關鍵在於：寄送者和收件者能夠利用雙線性對的運算性質，和雙方各自擁有的資訊算出同樣的 session key，而這把即可用來保護之間的通訊。接下來詳述 Boneh 和 Franklin 的設計：

- 系統初始：PKG 選擇系統參數 $\langle G_1, G_2, H_1, H_2, E, P, Q_0, e \rangle$ 並選擇系統的金鑰 s_9，將系統參數公開，但系統金鑰咻持祕密。其中 P 為 G_1 中的一個次數為 q 的生

成元（generator），G_1 為橢圓曲線 E 上的點，$H_1 : \{0,1\}^* \rightarrow G_1$、$H_2 : G_2 \rightarrow \{0,1\}^*$，$s_0 \in F_q^*$，並設定 $Q_0 = s_0 P$。

- 金鑰抽取：PKG 認證使用者，其密碼系統中的公鑰為 $P_{ID} = H_1(ID)$，對應的私鑰為 $S_{ID} = s_o P_{ID}$。如圖 4 中 Bob 的公鑰為 P_{Bob}，對應的私鑰為 $S_{Bob} = s_0 P_{Bob}$。

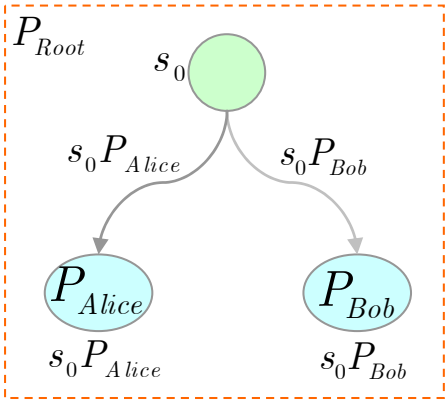

圖 4：IBE 私鑰抽取的示意圖

- 加密算法：假設 Alice 要加密文件 M 給 Bob，Alice 任選一個亂數 $r \in F_q^*$ 並計算使用者 Bob 的公鑰 $P_{Bob} = H_1(Bob)$，則 $C = \langle rP, M \oplus H_2(e(P_{Bob}, Q_0)^r) \rangle$
- 解密算法：假設 Bob 收到密文 $C' = \langle U, V \rangle$，則計算 $M' = V \oplus H_2(e(S_{Bob}, U))$。

2.2.2 階層式身份基礎加密系統設計

Hierarchical Identity-based Encryption (HIBE) ，最初由 Horwitz 和 Lynn[9]提出，他們設計出二層的架構，主要是要解決單一 PKG 設計中，root PKG 工作量太大，容易成為系統的瓶頸，但這個設計有安全上的瑕疵。Gentry 和 Silverberg[3]修改他們設計，並提出可以更多階層的身份基礎密碼系統，系統中的每個實體位於某個階層中，並且讓 root PKG 在第 0 層。有別於非階層式的 IBE，這個系統中實體的身份表示為 ID-tuple，由 root 追蹤到該實體所在的階層。也就是說，如果一個實體的 ID-tuple 為 $\langle ID_1, \ldots, ID_t \rangle$，那他的祖先 (位於第 i 階層，除去 Root) 的

ID-tuple 為 $\langle ID_1,...,ID_i\rangle, 1 \le i < t$。每一個內節點 (internal node) 的點會選擇該層的祕密點 (secret point) ，並用它來產生下一層使用者的私鑰。在這樣的設計下，所有的運算只需要 root PKG 的公開資訊。

此外，Gentry 和 Silverberg 也提出了雙重 (dual) 版本 HIBE[3]，有別於前者的直接 (direct) 版本，能夠節省計算和通訊上的花費，其中的技巧在於，在雙重版本中，藉由雙線性對的運算和 Q 值的應用，雙方算出的共同秘密，此秘密和他們最低層次祖先有關，一方面平均雙方的計算量，一方面也縮短密文的長度。

最後，Chow[15]巧妙的結合 Gentry 和 Silverberg 提出的 HIBE/HIBS，將他們合併，產生階層式簽密 (Hierarchical Identity-based Signcryption, HIBSE) ，經過仔細的分析，能夠比單獨先做 HIBS 再做 HIBE 的情況節省一些密文長度，若有簽密的需求，Chow 的設計也是一種選擇。

有關 HIBE/HIBS/HIBSE 的運算將總結如下，符號與 Boneh-Franklin 的論文[6]相同，每個運算的系統初始與金鑰抽取的步驟皆相同，所以獨立出來說明，請參考圖 5。

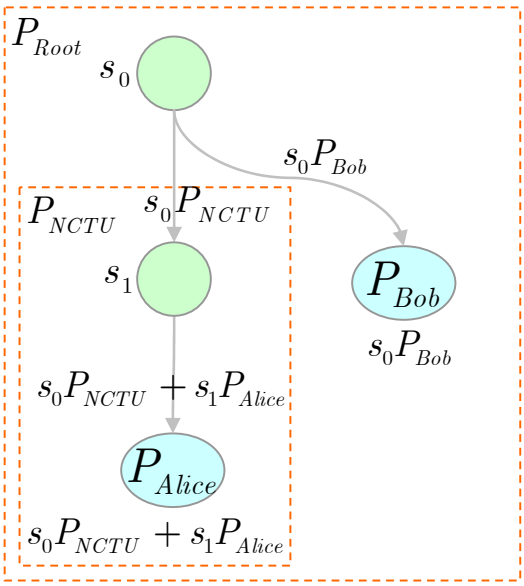

圖 5：HIBE 私鑰抽取的示意圖

- 系統參數初始：root PKG 選擇系統參數 $\langle G_1, G_2, e, n, P_0, Q_0, H_1, H_2 \rangle$ 及系統金鑰 s_0 。其中 P_0 代表 G_1 中的生成元素，$Q_0 = s_0 P_0$ 代表系統的公鑰，系統參數必需公開給所有系統使用者知道，但系統金鑰必須保持秘密。

- 低階層參數初始：低階層 PKG (位於階層 t) ，選擇此層的秘密點 s_i 並發給下層使用者 (位於階層 $t+1$) 使用者金鑰

- 使用者私鑰金鑰抽取：.位於 $t-1$ 階層的 PKG ，其身份識別為 $(ID_1, ..., ID_{t-1})$ ，能夠產生位於 t 階層的使用者金鑰，其身份識別為 $(ID_1, ..., ID_{t-1}, ID_t)$ 。 PKG 利用系統參數，自己的私鑰 S_{t-1} 和秘密點 s_{t-1} 進行這個運算。. PKG 先計算 $P_t = H_1(ID_1, .., ID_t) \in G_1$ 並設定使用者私鑰為 $S_t = S_{t-1} + s_{t-1} P_t \in G_1$ $(t = 1, S_0 = 1_{G_1})$ 。此外，PKG 也計算 Q 值： $Q_i = s_i P_0$, for $1 \le i \le t-1$ 並安全的傳送使用者私鑰及 Q 值。

基本運算 (Direct Version of Operations)

直接版本 HIBE

- *加密法：*要將訊息 M 加密給使用者，其階層中的身份識別為 $(ID_1, ..., ID_t)$ ，加密者先計算 $P_i = H_1(ID_1, .., ID_i) \in G_1$ ，$1 \le i \le t$ ，並且隨機選擇亂數 $r \in Z_q^*$ 。則密文為 $C = \langle rP_0, rP_2, ..., rP_t, M \oplus H_2(e(P_1, Q_0)^r) \rangle$

- *解密法：*當使用者收到密文 C' ，其格式為 $C' = \langle U_0, U_2, ..., U_t, V \rangle$ ，則解密者使用自己的私鑰 S_t 以及在金鑰抽取時得到的 Q 值： $Q_i = s_i P_0$ ， $1 \le i \le t$ ，將可以得回明文 $M' = V \oplus H_2(e(S_t, U_0) \prod_{i=2}^{t} e(U_i, Q_{i-1})^{-1})$ ■

直接版本 HIBS

- *簽章法：*要將訊息 M 簽發給使用者，簽章者在階層中的身份識別為 $(ID_1, ..., ID_t)$ ，簽章者先計算 $P_M = H_1(M) \in G_1$ 並且隨機選擇亂數 $r \in Z_q^*$ 。最後利

用自己的私鑰 S_t 設定簽文 $Sig = S_t + rP_M$ 。將 Sig 和驗證所需 Q 值 $Q_i = s_i P_0$ ，$1 \le i \le t$ 傳遞給使用者。

- *驗證法：*當使用者收到簽文 Sig ，且要驗證是否真的從簽章者而來，假設簽章者的身份識別為 $(ID_1,...,ID_{t-1})$ ，若 $e(P_0, Sig) = e(Q_0, P_1)e(Q_t, P_M)\prod_{i=2}^{t} e(Q_{i-1}, P_i)$ 則驗證通過。■

直接版本 HIBSE

- *簽章加密法：* 簽密者 Alice (身份識別為 $(ID_1,...,ID_t)$) 想簽章加密訊息 M 給使用者 Bob，他先計算 $P_M = H_1(M) \in G_1$ 並且隨機選擇亂數 $r \in Z_q^*$。接著利用自己的私鑰 S_t 設定簽文為 $Sig = S_t + rP_M$ 。接著要將 Sig 加密給使用者 (身份識別為 $(ID_1,...,ID_{t-1})$) ，簽密者 Alice 先計算使用者的 $P_i = H_1(ID_1,..,ID_i) \in G_1$ ，$1 \le i \le t$ 並且隨機選擇亂數 $r \in Z_q^*$ 。接著計算 $Q_M = s_t P_0$ 和 $C = \langle rP_2,...,rP_t, (M \| Sig \| ID_{Alice}) \oplus H_2(e(P_1, Q_0)^r), Q_1,..Q_M \rangle$ 其中 $Q_i = s_i P_0$ ， $1 \le i \le t-1$ 。

- *解密驗證法：* 當 Bob 收到密文 C'，其格式為 $C' = \langle U_2,...,U_t, V, Q_1,...Q_M \rangle$，Bob 使用自己的私鑰 S_t 和金鑰抽取時得到的 Q 值：$Q_i = s_i P_0$ ， $1 \le i \le t$ 來得回明文 $M \| Sig \| ID_{Alice} = V \oplus H_2(e(S_t, U_0)\prod_{i=2}^{t} e(U_i, Q_{i-1})^{-1})$ 。接著 Bob 驗證簽章，利用式子 $e(P_0, Sig) = e(Q_0, P_1)e(Q_t, H_3(M))\prod_{i=2}^{t} e(Q_{i-1}, P_i)$ ，若等號成立，則驗證通過。 ■

雙重運算 (Dual Version of Operations)

不同於直接版本 HIBE/HIBS/HIBES，雙重版本 HIBE/HIBS/HIBES 能夠平衡通訊雙方的計算量 (直接版本中，加密/簽章較解密/驗證容易) 。此外，密文的長度也能大大的縮減。加密方現在僅需準備 $m-l$ 個 Q 值以便解密，簽章方也僅需同數量的 Q 值以供驗證。如圖 6：HIBE 中，直接版本與雙版本的對照所示，寄

送者階層為m，收件者階層為n，他們共同的祖先位於l階層。由圖 6 可以清楚的發現 (比較下半圖中，兩虛線的長度)，只要寄送者和共同先的距離小於共同祖先的階層，雙重版本表現較直接版本好。

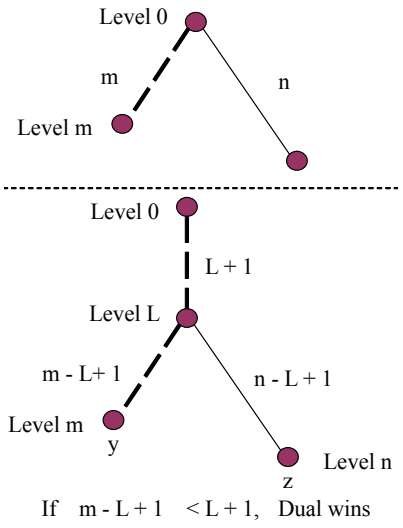

圖 6：HIBE 中，直接版本與雙版本的對照

雙重版本 HIBE

假設有兩個實體 Alice 和 Bob，在階層中的身份識別分別是$(ID_{y1},...,ID_{yl},...,ID_{ym})$和 $(ID_{z1},...,ID_{zl},...,ID_{zm})$，也就是說 Alice 位於階層$m$而 Bob 位於階層$n$，他們共享從 root (階層 0) 到階層$l$相同的祖先。

- *加密法：* 加密者 Alice 想要加密文件M給使用者 Bob。Alice 先計算$P_{i(z)} = H_1(ID_1,..,ID_i) \in G_1$，$l+1 \le i \le n$並且隨機選擇亂數$r \in Z_q^*$，接著設定密文為$C = \langle rP_0, rP_{l+!(z)},...,rP_{t(z)}, M \oplus H_2(g_{l(y)}^r) \rangle$，其中$g_{l(y)} = e(P_0, S_{m(y)}) \prod_{i=l+1}^{m} e(Q_{i-1(y)}, P_{i(y)})^{-1}$

- *解密法：*當 Bob 收到密文C'並假設其格式為$C' = \langle U_0, U_{l+1},...,U_n, V \rangle$，Bob 使用自己的私鑰$S_t$以及在私鑰抽取時得到的$Q$值：$Q_i = s_i P_0$， $l+1 \le i \le n$來回復明文$M' = V \oplus H_2(e(U_0, S_z) \prod_{i=l+!}^{n} e(Q_{z(i-1)}, U_i)^{-1})$ ■

雙重版本 HIBS

- *簽章法*: 簽章者 Alice 想要簽章訊息 M 給使用者 Bob。Alice 先計算 $P_M = H_1(M) \in G_1$ 並且隨機選擇亂數 $r \in Z_q^*$。接著 Alice 使用自己私鑰 S_t 設定 $Sig = S_t + rP_M$ 和簽章 $Sig' = e(P_0, Sig)\prod_{i=1}^{l} e(Q_0, P_i)^{-1}$，將 Sig' 和驗證用 Q 值：$Q_i = s_i P_0$ ，$l+1 \le i \le m$ 傳給 Bob。

- *驗證法*：Bob 收到來自於 Alice 的簽章，驗證是否 $e(P_0, Sig') = e(Q_t, P_M)\prod_{i=l+1}^{m} e(Q_{i-1}, P_i)$，若等號成立，表示驗證通過。 ■

雙重版本 HIBSE

- *簽章加密法*：Alice 想要簽章加密訊息 M 給 Bob，她先計算 $P_M = H_1(M) \in G_1$ 並且隨機選擇亂數 $r \in Z_q^*$。接著 Alice 使用自己的私鑰 S_t 並設定簽章為 $Sig = S_t + rP_M$。接著用 Bob 公鑰加密訊息，Alice 先計算 Bob 的 $P_i = H_1(ID_1, .., ID_i) \in G_1$，$1 \le i \le t$ 並且隨機選擇亂數 $r \in Z_q^*$，計算 $Q_M = s_t P_0$ 並設定密文 $C = \langle rP_2, ..., rP_t, (M \| Sig \| ID_{Alice}) \oplus H_2(g_{l(y)}{}^r), \quad Q_{1(y)}, ...Q_{M(y)} \rangle$，其中 $Q_{i(y)} = s_i P_0 \quad 1 \le i \le t-1$，$g_{l(y)} = e(P_0, S_{m(y)})\prod_{i=l+1}^{m} e(Q_{i-1(y)}, P_{i(y)})^{-1}$。

- *解密驗證法*：當 Bob 收到來自 Alice 的密文 C'，密文格式為 $C' = \langle U_2, ..., U_t, V, Q_{1(y)}, ...Q_{M(y)} \rangle$，Bob 使用自己的私鑰和在私鑰抽取過程中得到的 Q 值：$Q_{i(z)} = s_i P_0$，$1 \le i \le t$ 來得回明文 $M \| Sig \| ID_{Alice} = V \oplus H_2(e(U_0, S_z)\prod_{i=l+1}^{n} e(Q_{z(i-1)}, U_i)^{-1})$。Bob 接著驗證是否 $e(P_0, Sig) = e(Q_t, H_3(M))\prod_{i=2}^{t} e(Q_{i-1}, P_i)$，若等號成立，表示訊息完整且是從 Alice 傳來的。 ■

直接版本的運算分析如表格 1，傳送者位於階層 m ，接收者位於階層 n ：

	Setup	Extract	Encrypt	Decrypt	Sign	Verify
Scalar multi. in (G_1)	1	2	n		2	
Addition in (G_1)		1			1	
Pairing			1	n		$n+2$
Exponent in (G_2)			1			
Hash operation		$n+1$	2	1	2	
XOR operation			1	1		

表格 1：直接版本的運算分析

雙重版本的運算分析如表格 2，傳送者位於階層 m ，接收者位於階層 n ，共同祖先位於階層 l ：

	Setup	Extract	Encrypt	Decrypt	Sign	Verify
Scalar multi. (G_1)	1	2	$n-l+1$		2	
Addition in (G_1)		1			1	
Pairing			$m-l+1$	$n-l+1$		$n+2$
Exponent in (G_2)			1			
Hash operation		$n+1$	2	1	$n+1$	
XOR operation			1	1		

表格 2：雙重版本的運算分析

2.3 秘密分享與門檻密碼學

秘密分享 (secret sharing) 和門檻密碼學 (threshold cryptography) 都是讓一群人共享祕密的方式，但其中最大的不同就是：在秘密分享中，祕密是先確定的，也就是會有一個可信任的莊家握有祕密，並利用秘密分享函式來分給使用者部分祕

密。而門檻密碼學則是讓使用者藉由訊息的交換，共享一個祕密，但沒有但何一個人知道最終的祕密，通常也不允許他們重建，若加強門檻加密法的安全性，移除可信任莊家的需求，我們稱此設計為分散式金鑰產生 (distributed key generation, DKG) 。接下來的章節，我們會先介紹秘密分享，並說明門檻法的概念並介紹我們使用的分散式金鑰產生。以下說明門檻法密碼系統的設計需求：

- t-安全 (t-secure) 門檻法系統

若至多 t 個實體合作，是無法重建分享的秘密，也無法得知任何關於秘密的資訊。

- t-強健 (t-robustness) 門檻法系統

若至多 t 個惡意或是被麥解的實體參與，其餘實體仍可有效率的重建分享秘密。

給定 P, aP, bP, cP， $a,b,c \in Z_p^*$， *計算出* $d \in G_2$ *，使得* $d = \hat{e}(P,P)^{abc} \in G_2$

2.3.1 Shamir 秘密分享

Shamir[3]祕密分享的設計，主要的構想是利用多項式的內插法 (polynomial interpolation) 。此設計中假說了一個可信任的莊家 TD ，他會發給不同的實體部分祕密，讓他們能夠共享一個祕密。這個設計我們說明如下：

1. TD 選擇在 Z_q 上，次數為 $t-1$ 的多項式 $f(z) = \sum_{i=0}^{t-1} a_i z^i \pmod q$ 令 t 是設定的門檻值，並設定 $f(0)$ 為欲分享的祕密 x 。

2. TD 幫每一個實體 ID_i 計算他們該有的部分祕密 $x_i = f(ID_i)$

任何包涵t個實體的群組能利用Lagrange內插法，回復分享的祕密 x 。方法如右：$f(z) = \sum_{i=1}^{t} x_i \lambda_i(z)$ ，其中 $\lambda_i = \prod_{j=1}^{t} x_i \lambda_i(0) \pmod q$ 。任何小於 t 個實體的群組將無法得知任何有關 x 的資訊。

2.3.2 分散式金鑰產生法

在利用 Shamir 的設計時，我們不想讓某個實體能夠獨自選定並擁有這個祕密分享函式，也就是說，我們在設計中不想假設有一個可信任的莊家 (TD)，而是會利用某種機制，能讓決定和擁有這兩個動作，能夠由多個實體來執行。第一個設計出此種分享設計的人為 Peterson[18]，他的設計讓參與者自選部分秘密及部分秘密分享函式，藉由交換手中函式的資訊，他們能夠共享一個秘密，但最終被證明有安全上的漏洞。經過修改，Gennaro[1]提出安全的合作產生秘密的方式，稱為分散式金鑰產生 (distributed key generation, DKG)，我們記為 GJKR-DKG，在 GJKR-DKG 中，不再需要假設有一個可信任的莊家，取而代之的是一群玩家(P)來決定祕密分享函式。我們要強調的是，在 DKG 的設計中的玩家(P)不再需要是可信的，因為在玩家互相交換訊息的過程，他們就能共同決定一個可信任的群組，並且能偵測並移除掉不誠實之玩家對於最後祕密的影響。

2.3.2.1 Tang's 分散式金鑰產生法

Tang 等人設計出 TC-DKG[4]，其運作的原理和 GJKR-DKG[1]是一致的，只是前者應用於基礎在加法群上的密碼系統，而後者應用於基礎在乘法群上的密碼系統，在這兩種設計中，分成四個步驟，完成這四個階段，每個玩家擁有部分的祕密分享函式，和部分的祕密。而所有玩家共同分享的祕密不會被任何一個玩家所算出。這個設計在惡意玩家出現時，仍可以藉由其後驗證階段 (KV) 找出，並消除惡意玩家帶來的影響。我們說明 TC-DKG 如下：

T 為各個同儕領域的公鑰 P_i，而 T' 是系統公開的參數，即為HIBE中的 P_0，Q_0 則為root 同儕領域，執行完TC-DKG後即可得到，HIBE系統中其餘的系統參數，如Hash function H_1、H_2、q、$E(F_q)$、$\hat{e}$ 都可以事前先決定，成為系統公開的資訊。此外，ID_i (Identifier，身份識別) 代表身份基礎公開金鑰系統中，同儕實體

或是領域實體在系統中的相關資訊，用於導出實體的公鑰。而 P_i 侍表同儕實體或是領域實體所擁有的公鑰，此公鑰是由身份識別衍生而來。

- Key Distribution (KD) phase: 所有 SFG / PKG 成為一個集合 S 。

使用者 ID_i 選擇 $2t+1$ 個數：$a_{ik} \in GF(q)$ 和 $b_{ik} \in GF(q)$ ，$0 \le k \le t$ ，利用他們產生兩個次數 (degree) 為 t 的多項式：$f_i(z) = \sum_{0}^{t} a_{ik} z^k$ ， $f_i^{'}(z) = \sum_{0}^{t} a_{ik} z^k$ 。ID_i 幫 $ID_j \in S$ 計算 $s_{ij} = f_i(c)$ 和 $s_{ij}^{'} = f_i(c)$，其中 c 代表 P_j 所對應到的隊伍，利用秘密通道將 s_{ij} 和 $s_{ij}^{'}$ 傳送給 ID_j 。

- Key Generation (KG) phase:

使用者 ID_i 計算並廣播 $A_{i0} = a_{i0}T$ 。並從 $P_j \in S$ 得到 $A_{j0} = a_{j0}T$ ，計算出此層的 Q 值 $Q_l = y = \sum_{j \in S}^{\oplus} A_{j0}$ 。

上述的分散式金鑰產生方式運作於全為誠實的同儕，若要增加強健性 (robustness) ，TC-DKG 也提供了方法，不僅能找出說謊的人，並能移除他們對金鑰的影響，在過程中會定義一個 $QUAL$ 集合，包括所有合法的分散金鑰參與者，根據文獻中的證明，所有的合法參與者會算出相同的 $QUAL$ 集合，或利用 Key Verification (KV) 階段與 Key Check (KC) 階段中，我們將過程說明如下。

- Key Distribution (KD) 階段:

除了利用秘密通道將 s_{ij} 和 $s_{ij}^{'}$ 傳送給 P_j ，且利用廣播通道公開關於手中多項式的資訊 $P_{ik} = (a_{ik}T) \oplus (b_{ik}T^{'})$ ， $0 \le k \le t$ 。

- Key Verification (KV) 階段:

當使用者 P_i 收到從 $P_j \in S$ 來的 s_{ji} 和 $s_{ji}^{'}$ ，利用式子 (1) 驗證正確性：

$$(a_{ik}T) \oplus (b_{ik}T') = \sum_{k=0}^{t} (ID_i)^k P_{jk} \quad (1)$$

其中其中 ID_i 代表 P_i 所對應到的身份識別。

1) 若對於 ID_j 的 s_{ji} 或 s'_{ji} 驗證 (1) 不通過，則廣播一個對 ID_j 的抱怨 (complaint) 。

2) 若 ID_j 收到從 ID_i 來的抱怨，則廣播滿足 (1) 的 s_{ji} 和 s'_{ji} 。

- Key Check (KC) 階段:

若滿足下列兩種情況，使用者 ID_j 將被移出 Q ，而且 P_i 更新其秘密為 $\sum_{j \in Q} s_{ji}$

1) 收到多於 $t+1$ 個來自不同使用者的抱怨 ID_j

2) 收到 ID_j 廣播的辯駁 s_{ji} 或 s'_{ji} ，但仍然不能滿足 (1)

- Key Generation (KG) 階段: 和先前版本相同 ■

TC-DKG 的運算複雜度如表格 3，詳細分析請參考[4]：

	乘法運算	加法運算	最差情況記憶體使用
KD	$(2t+2, 2cU(t+1))$	$(t+1, 2ct)$	$2(t+1)\log_2(p)+K$
KV	$(c(t+3), cU(t+1))$	$((c-1)(t-1), 0)$	$(2c+t+1)\log_2(p)+K$
KC	$(2t, 0)$	$(t, c-1)$	$(n-1)\log_2(p)+K$
KG	$(1, 0)$	$(n-1, 0)$	常數

表格 3：TC-DKG 運算複雜度分析

若參與實體皆為誠實的同儕，則僅需要 KD 和 KG 兩步驟，若有說謊的同儕，則要再加入 KV 和 KC 兩個步驟，表格中的 (x, y) 分別代表在橢圓曲線主子群 (main subgroup) 的運算及在 $GF(q)$ 的運算。其中 $U(t)$ 代表 $t\log_2 t$ ，C 代表參與的人數，p 代表 $E/GF(q)$ order 中的大質數。而最差情況記憶體使用的表格中，K 是最差情況時，乘法的記憶體需求，此外，TC-DKG 通訊的複雜度如表格 4：

	接收訊息次數	傳送通道
KD	0	$(n-1,1)$
KV	$c-1$	$(0,t+1)$
KC	$t(c-2t-1)+t$	$(0,0)$
KG	$c-1$	$(0,1)$

表格 4：TC-DKG 通訊複雜度分析

其中傳送通道中的(x,y)分別代表秘密通道與廣播通道。

第 3 章 系統概論

Halo 的設計是為了解決二個在實作同儕網路安全時遇到的困難：

1. 缺乏伺服器的架構：同儕網路因缺乏伺服器的本質，讓它很容易的被部屬 (deploy) 和調整規模 (scale) ，但也讓同儕間的通訊很難被保護。由於缺乏安全的節點來作為信任關係的開端，當然也就無法簽發必要的安全標記 (secure credential) 。先前已有文獻記載在mobile ad-hoc網路中使用門檻式的憑證權威 (threshold certificate authority，TCA) [6]，這樣的機制同樣能夠保護同儕網路系統，然而TCA需要做到門檻式的計算RSA的公/私鑰和憑證簽章[19][14]，藉由門檻式的計算RSA模數 (Moduli) [5] 來達到。到目前為止，沒有一個可擴充的方法，在沒有可信任的莊家協盼助下或是不用麻煩的信息交換來支援門檻式的憑證簽章。目前所用的信息交換技術相當於安全的多方運算，這是個很煩雜的運算。在信任關係階層需維持的情況下，簽發公鑰憑證和憑證廢除列表 (revocation list) 的計算和通訊的複雜度，對於同儕網路系統中實際的應用，都會是個沉重的負擔。

對於機會性合作的需求：不像主從式 (client-server) 的交易，在同儕網路中的服務提供者 (service provider) 通常是根據操作的需求，以 ad-hoc 的方式選出。候選的服務者通常由分散式的搜尋或是 routing 的方式，如透過 distributed hash tables (DHT) 被發現。真實的服務提供者通常由一些操作上的需求選出，像是網路的連線、同儕的聲望 (reputation) ，甚至是同儕身上的工作量。因為服務提供者通常也是個普通的節點，所以在同儕網路交易中，沒有辦法預先得知或是限制參與者的行為。在缺乏操作控制和安全的基礎架構下，同儕網路中，若要提供同儕認證 (peer authentication) 、信息完整 (message integrity) 和信息祕密 (confidentiality) 這些基本的安全服務，面臨相當多的固難。

3.1 設計原則

此系統將發給每個合法使用者一對金鑰，而公鑰是從同儕的身份 (peer identity)、能提供的服務 (service) 以及授權的角色 (提供者或是需求者) 導出。由於在同儕網路中缺乏安全伺服器 (security server)，所以此系統使用多個同儕作為 *PKG*，這些 *PKG* 將使用 (t, n) 門檻式祕密分享運算來產生私鑰。所以同儕若想取得私鑰，甚至是成為某個服務的提供者或需求者，就必需要尋求至少 t 個 *PKG* 的授權認可。

圖 7 顯示的是我們設計的系統 *Halo*。為了要管理同儕，*Halo* 將他們安排成階層式的同儕群組 (peer group)。每個同儕群組被賦與可以提供或是訂閱某種特定的服務。就如同我們設計的 *PKG* 和 *SFG / PKG* 是分屬不同的同儕群組。而一個同儕群組若同時包涵 *PKG* 群組和 *SFG / PKG* 群組，且這個同儕群組就成為了同儕領域 (peer domain)。一個同儕領域擁有 *Halo* 最基本的管理元件，因為它擁有自治的能力：簽發私鑰給同儕並能夠再提升同儕為 *PKG*，之後再成為 *SFG / PKG*。一個網路節點需要得到某個同儕領域中 *PKG* 所簽發的私鑰，才能夠被允許加了該同儕領域。一個同儕領域可以藉由簽發和新的服務相關的私鑰，而產生多個同儕群組。一個同儕領域也可以讓自身的 SFG 產生新領域的 SFG，因而產生新的子同儕領域。

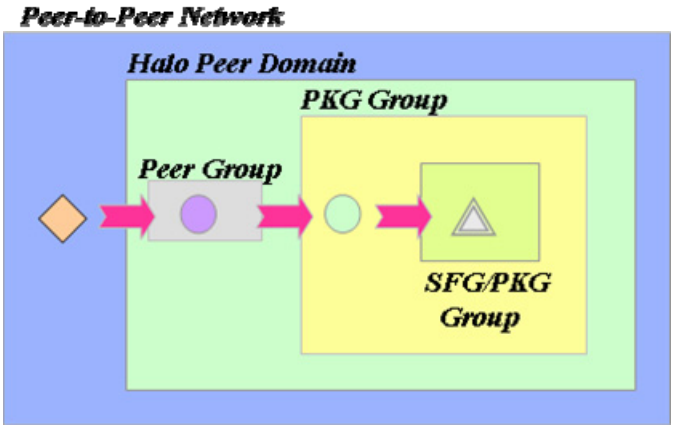

圖 7：Halo 系統同儕身份轉換的過程

3.2 操作原則

Halo 能夠用下面的兩種方式保護機會性合作中的同儕： (1) 它建立一基本階層式的同儕群組架構，用來管理並授權服務的提供 (provision) 或是訂閱 (subscription) ； (2) 它藉由身份基礎公/私鑰來來保護同儕間的交易 (transaction) ，而此私鑰是和同儕的身份、服務的型態和同儕在合作中的角色 (提供者或是訂閱者) 有關。圖 8 中顯示典型 *Halo* 對於授權服務發動 (service invocation) 的程序。為了能夠取得足夠的授權來參與機會性合作和發起授權的服務，同儕必需先被至少一個或多個同儕領域所認證，而每個領域需要至少 t tPKG 來參與認證的過程。這個允許程序允許同儕成為同儕領域的成員並且得到能夠得到身份基礎的公鑰和私鑰，而這對金鑰將能夠提供或是使用該同儕領域或是該領域之子領域的服務。

一旦被允許進入同儕領域，提供者將允許提供特定的服務，訂閱者也將能由同儕領域提供的搜尋服務 (discovery service) 來找尋欲使用的服務。搜尋服務將提供訂閱者有關一群被授權的提供者之身份與位置。

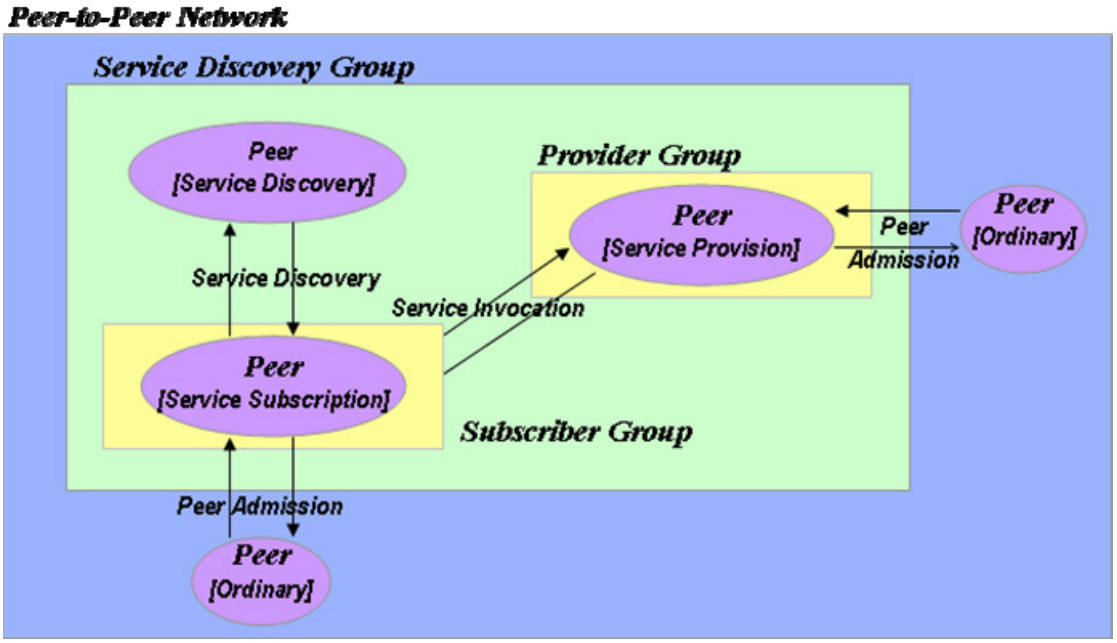

圖 8：Halo 系統同儕提供或訂閱服務的過程

第 4 章 系統機制

本章節將描述 *Halo* 系統所基硬的 PKI 架構及同儕網路結構，並提出系統中在結構上 (structural) 、功能上 (functional) 及惡意者 (adversary) 的假設，最後詳細說明滿足各項假設所需要的數學方法。以密碼算法而言，Boneh 和 Franklin 設計了身份基礎的加密算法 (IBE) ，對於系統設計而言，支持這樣的算法需要的系統基礎架構，我們稱之為身份基礎公開金鑰基礎架構 (IDPKI) 。在階層式的身份基礎加密算法，Gentry、Silverberg 提出了他們的加密算法，而支持的系統基礎架構，則因系統初始化的方式而分成 centralized 和 distributed (server-less) 的架構，*Halo* 系統即是提供了分散式的初始化方式，並且利用門檻法來達成，所以 *Halo* 系統可以稱為一種門檻式 HIBE (threshold HIBE, THIBE) 。

因 *Halo* 系統將會部屬在同儕網路中，所以我們在設計的過程也做了下列的假設：

- 功能上的假設:
  - 無伺服器 (server-less) 環境：

    在同儕網路環境中，無法事前安排永遠在線上的伺服器，且能被所有人信任。
  - 沒有可信任莊家 (trusted dealer) 存在：

    在同儕網路中，無法假設存在一個可信任的莊家，由他來發放系統部分秘密。
- 結構上假設:
  - 管理的階層沒有限制：

    在 *Halo* 的設計中，管理的層次是不受限制的，只要有需要，都能夠產生出下一個層次，以符合實際應用。
- 通訊上假設:
  - 訊息傳送：在訊息傳送時，要求有順序性，先傳送的訊息要先到達。並假設在過程中沒有訊息會遺失，所以訊息一旦被送出，會在均一 (uniformly bounded) 的時間範圍內傳達。
  - 廣播通道：假設廣播通道的訊息，能夠傳達到所有的人，不然就是沒有人能夠收到。若能夠收到，則收到訊息的先後順序是隨機的。

- 秘密通道：假設秘密通道的訊息，不會被非預期接收者攔截或是監聽。

惡意者假設：

- 停止惡意者：在系統運作過程，不小心或是遭到問題而無法回應訊息。
- 監聽惡意者：能夠被動的監聽並攔截所有在公開通道的資訊。
- 靜態惡意者：在系統運作前即還定要攻破(corrupt)的使用者，在系統運作中無法視情況改變這個選擇。
- 重送惡意者：能夠儲存監聽到的訊息，並在假扮成某使用者時使用。

以下的符號將在接下來的論文中使用：

- ID_i (Identifier，身份識別) ：身份基礎公開金鑰系統中，同儕實體或是領域實體在 系統中的相關資訊，用於導出實體的公鑰。
- P_i：身份基礎公開金鑰系統中，同儕實體或是領域實體所擁有的公鑰，此公鑰是由身份識別衍生而來。
- S_i：身份基礎公開金鑰系統中，同儕實體或是領域實體所擁有的私鑰，此私鑰對應於公鑰 P_i。
- Credential (標示) ：身份基礎公開金鑰系統中，和同儕實體或是領域實體有關的資訊，於註冊或是認證時出示。
- Secret Sharing Function Generator (SFG) ：身份基礎公開金鑰系統中，擁有系統金鑰，金鑰分享函式的實體，用於新領域的產生，一旦完成初始，即可離開。每個SFG也會是個 PKG，我們也會將他們記作 SFG/PKG。
- Private Key Generator (PKG) ：身份基礎公開金鑰系統中，擁有系統金鑰，並以此產生同儕私鑰的實體。
- Registration Authority (RA) ：身份基礎公開金鑰系統中，將實體資格 (physical credential) 轉為數位資格 (digital credential) 的實體。

其餘未提及的符號與 Boneh-Franklin 的論文[5]相同。

4.1 密碼學加密法

Halo 將發給系統合法使用者公、私鑰對，其中公鑰是從同儕的身份識別、能提供的服務以及授權的角色 (提供者或是需求者) 導出。利用這個公私鑰對，合法

使用者能夠使用階層式身份基礎金鑰系統中的服務，如加密/解密，簽章/驗證、簽密 (sign-cryption) 、金鑰協議等。使用者也能根據其需求，使用不同版本的運算服務，如直接版本和雙重版本。下面將總結使用者如何利用 *Halo* 授權的金鑰對，執行身份基礎金鑰系統中各種運算，詳細的步驟與符號說明請見第二章。

直接版本 HIBE/HIBS/HIBSE

Operation	HIBE
Encrypt	$C = \langle rP_0, rP_2, ..., rP_t, M \oplus H_2(e(P_1, Q_0)^r) \rangle$
Decrypt	$C' = \langle U_0, U_2, ..., U_t, V \rangle$
	$M' = V \oplus H_2(e(S_t, U_0) \prod_{i=2}^{t} e(U_i, Q_{i-1})^{-1})$
Operation	HIBS
Sign	$Sig = S_t + s_t P_M$
Verify	Output verified
	if $e(P_0, Sig) = e(Q_0, P_1) e(Q_t, P_M) \prod_{i=2}^{t} e(Q_{i-1}, P_i)$

圖 9：直接版本 HIBE/HIBS 運算摘要

Operation	HIBES
Sign + Encrypt	$Sig = S_t + rP_M$ $C = \langle rP_2, ..., rP_t,$ $(M \parallel Sig \parallel ID_{sender}) \oplus H_2(e(P_1, Q_0)^r),$ $Q_1, ...Q_M \rangle$
Decrypt + Verify	$C' = \langle U_2, ..., U_t, V, Q_1, ...Q_M \rangle$ $M \parallel Sig \parallel ID_{sender} = V \oplus H_2(e(S_t, U_0) \prod_{i=2}^{t} e(U_i, Q_{i-1})^{-1})$ Output verified if $e(P_0, Sig) = e(Q_0, P_1) e(Q_t, H_3(M)) \prod_{i=2}^{t} e(Q_{i-1}, P_i)$

圖 10：直接版本 HIBSE 運算摘要

雙重版本的運算在同儕網路中有著很高的利用價值，由於通訊的雙方，通常處於同一個同儕群組或是分屬一個同儕群組中的子群組，所以有很高的機會會有低於 root 的祖先，而且寄送者越靠近共同組先，利用雙版本的運算效果就更好。我們將雙重版本的通訊稱為本地通訊 (local communication) ，而牽涉到較遠，且共同祖先離寄送者較遠的通訊稱作國際通訊 (global communication) ，在同儕網路中視情況使用直接版本或雙重版本的運算，將大大的減少密文的長度，並且能夠保證，在大部分的情況下，密文長度會落在某個限度之中 (比較寄送者和接收者的距離和接收者到 root 的距離) 。接下來我們繼續介紹雙重版本的運算服務。

雙重版本 HIBE/HIBS/HIBSE

假設有兩個實體 Alice 和 Bob，在階層中的身份識別分別是 $(ID_{y1},...,ID_{yl},...,ID_{ym})$ 和 $(ID_{z1},...,ID_{zl},...,ID_{zm})$，也就是說 Alice 位於階層 m 而 Bob 位於階層 n，他們共享從 root (階層 0) 到階層 l 相同的祖先。圖 11、圖 12 摘要 HIBE、HIBS 及 HIBES 之運算。

Operation	Dual-HIBE
Encrypt	$C = \langle rP_0, rP_{l+1(z)},...,rP_{n(z)}, M \oplus H_2(g_{l(y)}{}^r) \rangle$
	Where $g_{l(y)} = e(P_0, S_{m(y)}) \prod_{i=l+1}^{m} e(Q_{i-1(y)}, P_{i(y)})^{-1}$
Decrypt	$C' = \langle U_0, U_{l+1},...,U_n, V \rangle$
	$M' = V \oplus H_2(e(U_0, S_z) \prod_{i=l+!}^{n} e(Q_{z(i-1)}, U_i)^{-1})$
Operation	Dual-HIBS
Sign	$Sig = S_t + s_t P_M$
	$Sig' = e(P_0, Sig) \prod_{i=1}^{l} e(Q_0, P_i)^{-1}$
Verify	Output verified
	if $e(P_0, Sig') = e(Q_t, P_M) \prod_{i=l+1}^{m} e(Q_{i-1}, P_i)$

圖 11：雙重版本 HIBE/HIBS 運算摘要

Operation	Dual-HIBES
Sign + Encrypt	$Sig = S_t + rP_M$ $C = \langle rP_2, ..., rP_t, (M \parallel Sig \parallel ID_{sender}) \oplus H_2(g_{l(y)}{}^r), Q_1, ...Q_M \rangle$ where $g_{l(y)} = e(P_0, S_{m(y)}) \prod_{i=l+1}^{m} e(Q_{i-1(y)}, P_{i(y)})^{-1}$
Decrypt + Verify	$C' = \langle U_0, U_{l+1}, ..., U_n, V, Q_1, ...Q_M \rangle$ $M \parallel Sig \parallel ID_{sender} = V \oplus H_2(e(U_0, S_z) \prod_{i=l+!}^{n} e(Q_{z(i-1)}, U_i)^{-1})$ Output verified if $e(P_0, Sig) = e(Q_t, H_3(M)) \prod_{i=2}^{t} e(Q_{i-1}, P_i)$

圖 12：雙重版本 HIBSE 運算摘要

Ephemeral Diffie-Hellman Key Agreement with HIBE/HIBS Protection

在同儕網路的機會性合作中，或許需要往來多個訊息，若每個訊息皆使用加密方式保護安全性，則系統運作的效率將的大大的降低，我們可以考慮金鑰協議 (key agrement) ，並利用上述的加密法或是簽章法來達成，做法如下：Alice 和 Bob 事先協議一個橢圓曲線加法群$\langle P \rangle$，其 group order 為r，Alice 隨機選擇$r_A \in Z_q^*$並計算$r_A P$，Bob 也選擇一個亂數$r_B \in Z_q^*$並計算$r_B P$，他們將計算出來的$r_A P$、$r_B P$利用加密或是簽密傳給對方，以確保訊息是被認證過的 (authenticated) ，防止中間人 (man-in-the-middle) 攻擊，交換訊息後 Alice 計算$r_A(r_B P)$，Bob 計算$r_B(r_A P)$，則 Alice 和 Bob 協議出一把金鑰為$r_A r_B P$，能夠用於需要保護的 session。

保護 ephemeral key 的方法通常以加密或是簽章來達成。一般來說，簽章/驗證所需的運算較少，其處理過的密文長度也較短，但先決條件就是使用者必需先取得自己的私鑰，才能夠進行簽章保護的 Diffie-Hellman 金鑰協議。而加密保護 ephemeral key 的好處在於：使用者只需要知道對方的身份，即可導出對方的公鑰，接著就可以做加密保護的 Diffie-Hellman 金鑰協議。此外，加密保護的

Diffie-Hellman 金鑰協議也能減緩 DoS 的攻擊，因為在這個情境中，拿私鑰和解密文兩個步驟是最耗資源的，當使用者發覺密文可能是從可疑的地方傳送，或是不停的發送同樣的需求(request)，使用者判斷，並決定是否要參與交易(transaction)，如此能夠過濾不合法的訊息，避免受到攻擊。 ■

4.2 門檻式私鑰產生

由於在同儕網路中缺乏集中式且被信任的驗證機構，所以 *Halo* 採用多個同儕作為 PKG，為了提高 PKG 的安全性，避免單點失敗 (single point failure) 和增加使用者的可利用性，這些 PKG 將使用 (n,t) 門檻式秘密分享運算來產生私鑰，此外，我們將 PKG 根據同儕的身份識別分組，簡單的說，如果要將 PKG 要分成八個隊伍 (cohort)，我們將依照同儕身份最後三個位元來決定，若以 001 結尾的為第一組，002 為第二組，以此類推。此外，根據 Gentry- Silverberg 的設計，各階層的秘密和上層有密切的關係，為了讓 *Halo* 中各個階層的部分秘密能夠累加，組成階層中合法的部分秘密，PKG 手中的秘密不再是無意義的序號，而是對應於自己身份識別的隊伍，如此在同一個隊伍的 PKG 拿到的為同一塊部分秘密，也因為分組的政策 (policy) 固定，在驗證進入某個群組時，對應到的隊伍皆相同，不論進行幾次的認證，取得的部分秘密皆相同，如此能避免惡意的重複認證，拿到多個合法的金鑰對，而破壞系統整體的安全。所以同儕若想取得私鑰以成為某個服務的提供者或需求者，就必需在 c 個隊伍中，尋求至少 t 個 PKG 的授權認可 (一個隊伍找一個 PKG 認證，共找 t 個 PKG)，取得部分使用者私鑰，所以我們的門檻系統也稱作 (c,t) 秘密分享，因為我們把 n 個管理權威分成 c 個隊伍，每個隊伍中的 PKG 擁有相同部分的秘密。若拿到 t 個隊伍所產生的使用者部分私鑰，即能回復完整的使用者私鑰。而一般使用者能夠自己隊伍的 PKG 申請，並取得對應於該隊伍的部分秘密，成為該隊伍新的 PKG。

在 *Halo* 系統中，金鑰是採取 Gentry-Silverberg 的設計，如圖 13 所示，虛線方框

代表某個同儕領域管理的範圍，箭頭及上方的文字分別代表訊息流動的方向與欲傳送的訊息，圓形代表 PKG ，而橢圓形代表一般的同儕。最上層（第零層）為 Root PKG ，第一層的使用者，其公鑰為 P_{U_3} ，其私鑰（$S_{U_3} = 1_{G_1} + s_0 P_{U_3}$）為上層的領域私鑰（系統設計中，第零層的領域私鑰為 G_1 中的加法單位元素）加上直接祖先的秘密點 s_0 乘上公鑰 P_{U_3}。若新的 PKG 階層要產生，則下層選出一個使用者 P_{U_k} ，其私鑰如同上所述，為 $S = s_0 P_{U_k}$ 。此外，上層領域權威還賦予他新領域權威的職責，假設新領域的公鑰為 P_{D_1} ，則上層領域權威交給他領域的私鑰 $S_{D_1} = s_0 P_{D_1}$ ，用於發行次層使用者的私鑰。如圖中的使用者 P_{U_4} ，其私鑰（$S_{U_4} = s_0 P_{D_1} + s_1 P_{U_4}$）為上層領域的私鑰，加上祖先的秘密點 s_1 乘上公鑰 P_{U_4} 。

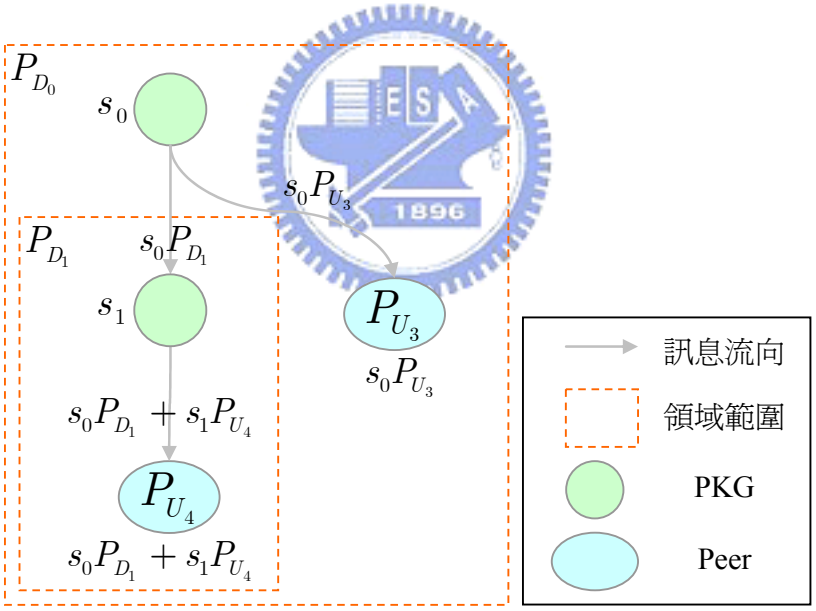

圖 13：HIBE 金鑰與階層的關係圖

為了聚焦在我們加強 Gentry-Silverberg 的設計上，我們先將 PKG 如何共享秘密的方式當作黑盒子，先假設同一個階層的 PKG 能夠利用門檻法的方式共享一個秘密，之後用專門的章節說明這個方法。圖 14 說明 *Halo* 系統中，PKG 的產生與安排，虛線方框代表某個同儕領域管理的範圍，箭頭及上方的文字分別代表訊息流動的方向與欲傳送的訊息，圓形代表某一隊的 PKG ，而橢圓形代表一般的

同儕。其中 s_1^2 的符號，上標代表隊伍索引 (cohort index) ，下標代表階層索引 (level index) ，所以 s_1^2 表示第一個階層的秘密點 s_1 ，依門檻法拆開時，屬於第二個隊伍。*Halo* 系統中，為了要能夠維持 HIBE 的設計，一方面能利用門檻法的方式分散系統金鑰風險，所以每個同儕領域的 PKG 拿到的不再是完整的同儕領域的私鑰，而是與自己身份對應的隊伍，其保存的部分同儕領域金鑰。此外，我們也需讓每一層的分類法都是一樣，如此同分類不同階層的部分同儕領域金鑰才能疊加，如圖中不同同儕領域 PKG 的階層，關於某個隊伍的資訊，一條鞭的往下傳，所以同一個身份的使用者，一旦經過驗證身份，確定分發的隊伍後，是無法得到其他隊伍的資訊，所以我們每層將會有 c 隊個隊伍。使用者必須向 t 個 PKG (一個隊伍找一個 PKG ，共 t 個) 認證，並取得使用者部分私鑰，也就是說，拿到 t 種使用者部分私鑰。如圖中的使用者 P_{U_3} ，在 $(c,t)=(3,2)$ 的設計中，必需取得 $s_0^1 P_{U_3}$ 和 $s_0^2 P_{U_3}$ 後，才能重建出自己的使用者私鑰。■

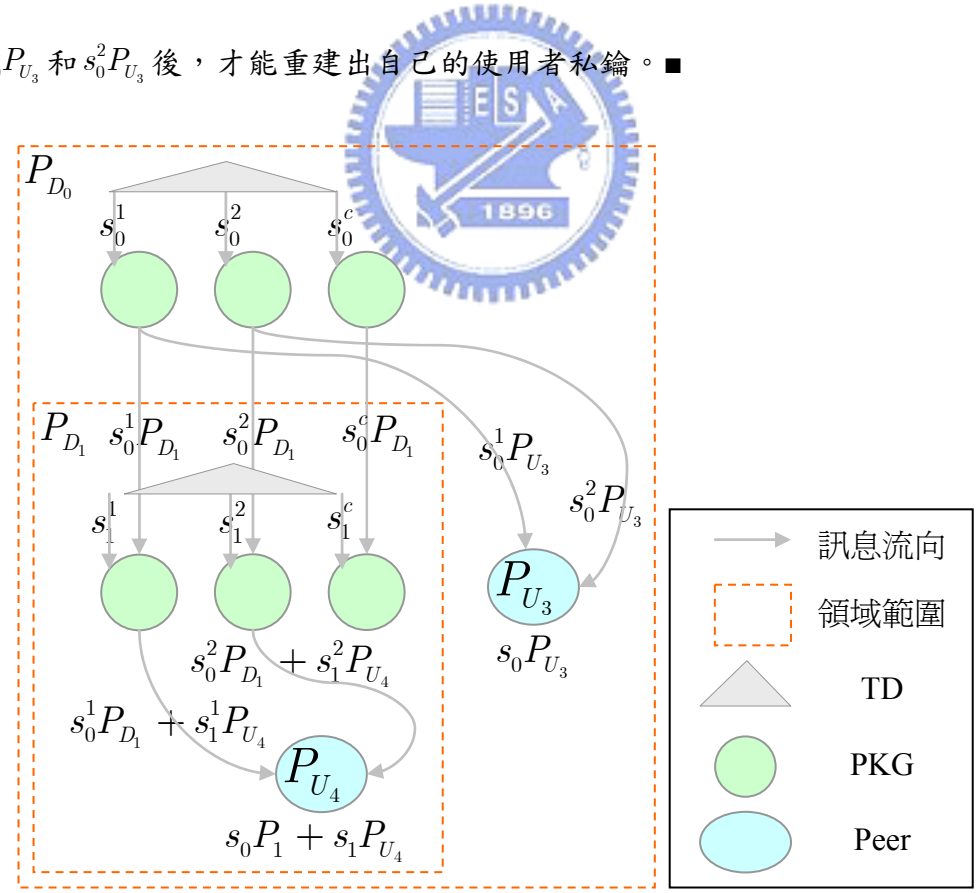

圖 14：HIBE 在假設 TD 下的金鑰抽取

4.3 分散式秘密分享函式產生

由於在門檻式秘密分享時，需要秘密分享函式來計算部分秘密，在同儕網路中的應用，甚至不能假設有可信任莊家的存在，選擇並持有該函式。在這樣的情況在，使用者必需能夠初始化整個密碼系統，並且產生系統所需的 PKG，以及 PKG 手中的部分秘密。也就是說，我們將分散式的初始化系統，因此，我們在 PKG 同儕群組中，獨立出一個新的同儕群組：SFG/PKG 同儕群組，群組中的所有的 SFG/PKG 協力決定秘密分享函式，一旦初始化，這此 SFG/PKG 的成員即可開。此外，如同 PKG 的做法，我們也將此 SFG/PKG 同儕群組根據同儕的身份識別分隊，每個隊伍只有一個 SFG/PKG，所有 SFG/PKG (有 c 隊就會有 c 個 SFG/PKG) 參與決定秘密分享函式的過程，並持有對應於該隊公鑰的部分 PKG 同儕領域金鑰，和部分同儕領域金鑰秘密分享函式。因此，一般使用者也可以藉由向對應於自己公鑰隊伍的的 PKG 申請，取得該群組的部分 PKG 同儕領域金鑰，成為該群組新的 PKG。

在 *Halo* 系統中，採用分散式金鑰產生方式：TC-DKG 初始化 HIBE，讓 SFG/PKG 共同決定 PKG 同儕領域秘密分享函式及 PKG 同儕領域部分秘密，因此 SFG/PKG 亦有 PKG 的功能，所以我們讓 SFG/PKG 同儕群組和 PKG 同儕群組分隊伍的方式相同，用公鑰來分出隊伍。如圖 15 所示，在 PKG 產生時所假設的可信任莊家 c 個不同隊的 SFG/PKG 所取代 (圖中以雙線三角形代表)，他們合作參與 TC-DKG 的過程。假設被推舉出的使用者為 P (屬於第 l 階層、第 c 隊)，則在 TC-DKG 之後，他將得到第 l 階層、第 c 隊所使用的 PKG 同儕群組部分私鑰分享函式 $f_l^c(x)$ 和 PKG 同儕群組部分私鑰 s_l^c。

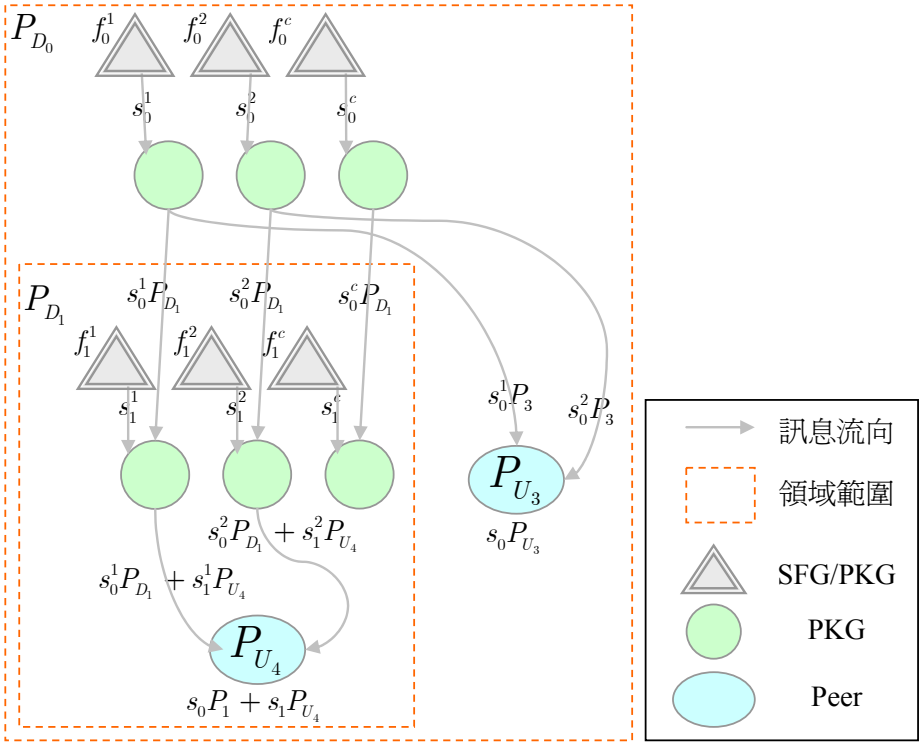

圖 15：HIBE 在沒有假設 TD 的金鑰抽取方式

Halo 中 SFG / PKG 共享秘密的方式詳述如下：

T 為各個同儕領域的公鑰 P_i，而 T' 是系統公開的參數，即為 HIBE 中的 P_0，Q_0 則為 root 同儕領域，執行完 TC-DKG 後即可得到，HIBE 系統中其餘的系統參數，如 Hash 函式 H_1、H_2、q、$E(F_q)$、$\hat{e}$ 都可以事前先決定，成為系統公開的資訊。

- Key Distribution (KD) 步驟: 所有 SFG / PKG 成為一個集合 S。

使用者 ID_i 選擇 2t+1 個數：$a_{ik} \in GF(q)$ 和 $b_{ik} \in GF(q)$，$0 \le k \le t$，利用他們產生兩個次數 (degree) 為 t 的多項式：$f_i(z) = \sum_0^t a_{ik} z^k$， $f_i'(z) = \sum_0^t a_{ik} z^k$。$ID_i$ 幫 $ID_j \in S$ 計算 $s_{ij} = f_i(c)$ 和 $s_{ij}' = f_i(c)$，其中 c 代表 P_j 所對應到的隊伍，利用秘密通道將 s_{ij} 和 s_{ij}' 傳送給 ID_j。

- Key Generation (KG) 步驟:

使用者 ID_i 計算並廣播 $A_{i0} = a_{i0} T$。並從 $P_j \in S$ 得到 $A_{j0} = a_{j0} T$，計算出此層的 Q 值 $Q_l = y = \sum_{j \in S}^{\oplus} A_{j0}$。

上述的分散式金鑰產生方式運作於全為誠實的同儕，若要增加強健性(robustness)，TC-DKG 也提供了方法，不僅能找出說謊的人，並能將他們對金鑰的影響移除，在過程中會定義一個$QUAL$集合，包括所有合法的分散金鑰參與者，根據文獻中的證明，所有的合法參與者會算出相同的$QUAL$集合，或利用Key Verification (KV)階段與 Key Check (KC)階段中，我們過程說明如下。

- Key Distribution (KD) 階段:

除了利用秘密通道將s_{ij}和$s_{ij}^{'}$傳送給P_j，且利用廣播通道公開關於手中多項式的資訊$P_{ik} = (a_{ik}T) \oplus (b_{ik}T^{'})$， $0 \le k \le t$。

- Key Verification (KV) 階段:

當使用者P_i收到從$P_j \in S$來的s_{ji}和$s_{ji}^{'}$，利用式子 (1) 驗證正確性：

$$(a_{ik}T) \oplus (b_{ik}T^{'}) = \sum_{k=0}^{t} (c)^k P_{jk} \tag{1}$$

其中其中c代表P_i所對應到的隊伍。

1) 若對於ID_j的s_{ji}或$s_{ji}^{'}$驗證 (1) 不通過，則廣播一個對ID_j的抱怨(complaint)。

2) 若ID_j收到從ID_i來的抱怨，則廣播滿足 (1) 的s_{ji}和$s_{ji}^{'}$。

- Key Check (KC) 階段:

若滿足下列兩種情況，使用者ID_j將被移出Q，而且P_i更新其秘密為 $\sum_{j \in Q} s_{ji}$

1) 收到多於$t+1$個來自不同使用者的抱怨ID_j

2) 收到ID_j廣播的辯駁s_{ji}或$s_{ji}^{'}$，但仍然不能滿足 (1)

- Key Generation (KG) phase: 和先前版本相同 ■

第 5 章 系統運作

在同儕網路系統中，同儕群組整合多個同儕，並提供特別的服務給組內的同儕，群組也能執行成員資格服務，增加守門員的角色，想加入群組的同儕必須滿足認證的要求，因此，同儕群組為同儕網路中，呈現服務的最小單位。*Halo* 為了要管理同儕，*Halo* 將他們安排成階層式的同儕群組。每個同儕群組被賦與可以提供或是訂閱某種特定的服務：產生使用者私鑰，或是產生系統部分祕密可視為一種服務，所以我們設計的 PKG 和 SFG/PKG 是分屬不同的同儕群組。而一個同儕群組若同時包含 PKG 群組和 SFG/PKG 群組，且這個同儕群組就成為了同儕領域。一個同儕領域擁有 *Halo* 最基本的管理元件，因為它擁有自治的能力：簽發私鑰給同儕，並能夠再提升同儕為 PKG 或是 SFG/PKG。一個網路節點需得到某個同儕領域簽發的私鑰，才能夠被允許加了該同儕領域。

如圖 16 所示，*Halo* 同儕領域包含三種同儕群組：SFG/PKG 群組、PKG 群組與一般同儕群組，此外，我們將 SFG/PKG 群組與 PKG 群組以同樣的方式分隊 (cohort) ，使每個群組保有同一份隊祕密 (cohort secret) ，如同 SFG/PKG 群組中，同隊的成員，都會擁有相同的部分祕密分享函式與部分祕密，而 PKG 群

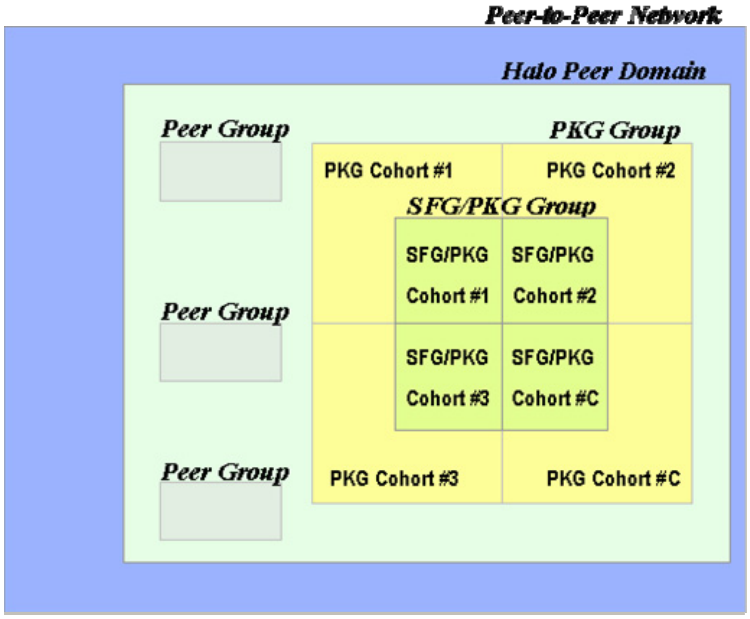

圖 16：Halo 系統同儕領域、同儕群組與分隊的設計

組中，同隊的成員，每個人會有相同的部分祕密，這兩個群組能夠藉由自動的招募，或是一般同儕自願提升的方式，維持足夠數量的人數，如此，一方面提高服務可利用性，也保護重要的祕密，以分散且門檻式的方式來避免單點失敗(single-point failure) 的問題。

圖 17 表示一個網路上的節點，若在部屬了 *Halo* 系統的同儕網路中，有哪些種類的身份轉換方面。一般的網路節點，能夠藉由執行同儕網路的程式模組，而進入同儕網路，如圖中菱形的點，進入同儕網路後，若此同儕需要 *Halo* 的服務支援，便需向 *Halo* 領域中，和自己身份對應的 *PKG* 認證並取得對應的 *Halo* 系統私鑰，即成為圖中紫色的點。過了一段時間，若同儕有意願成為領域的 *PKG*，則需向 *Halo* 領域中，和自己身份對應的 *PKG* 認證並取得對應的部分祕密，即成為圖中綠色的點，最後，使用者也能夠成為領域的 *SFG / PKG*，作法和成為 *PKG* 的過程相似，只是認證後取得的是對應於隊的部分祕密分享函式，即為圖中的雙線三角形。

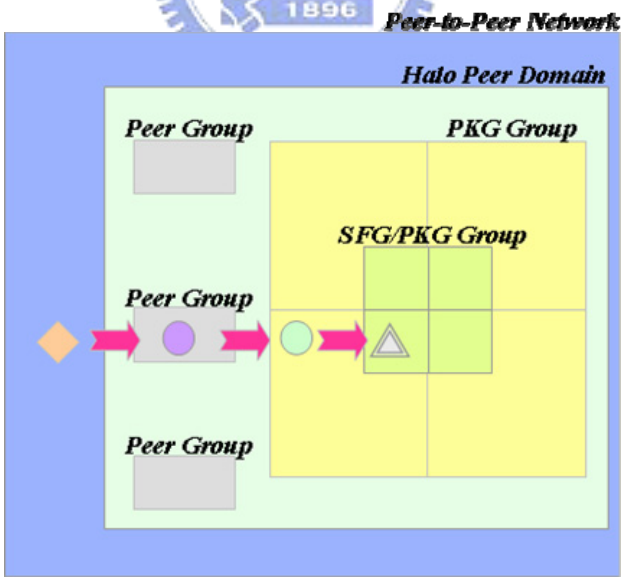

圖 17：Halo 系統中網路節點各種身份的演進

接下來的章節將會詳細的討論 *Halo* 系統初始化的步驟、新同儕領域的產生 (包含 *SFG / PKG* 群組及 *PKG* 群組) 、同儕身份在領域中變換的過程 (網路節點、同儕、*PKG* 、*SFG / PKG*) ，以及同儕進行跨領域的身份變換步驟。■

5.1 產生新同儕領域

在同儕網路中，同儕群組的產生，通常由同網路平台 (如：JXTA[11]) 預設的允許控制產生，通常是填寫表單或是鍵入密碼即可。但當某群同儕需要更高的安全等級並且使用制定好的允許政策 (admission policy) ，他們就需要一群權威 (authority) 擔任守門者並執行這個政策。最初在舊領域會有一群發起的同儕，他們向舊領域的 SFG/PKG 發出請求，並獲得新領域的名稱和對應於此名稱的領域私鑰。此外，他們也取得由舊領域 SFG/PKG 所簽發的允許政策，當完成這些步驟，這些同儕成為新領域的候選 SFG/PKG，他們利用§2.3.2.1 節提到的機制決定新領域共享的祕密函式與祕密，成為新領域第一個 SFG/PKG，也是 PKG。請參考圖 18：Halo 系統中產生新同儕領域的示意圖及下面的步驟。

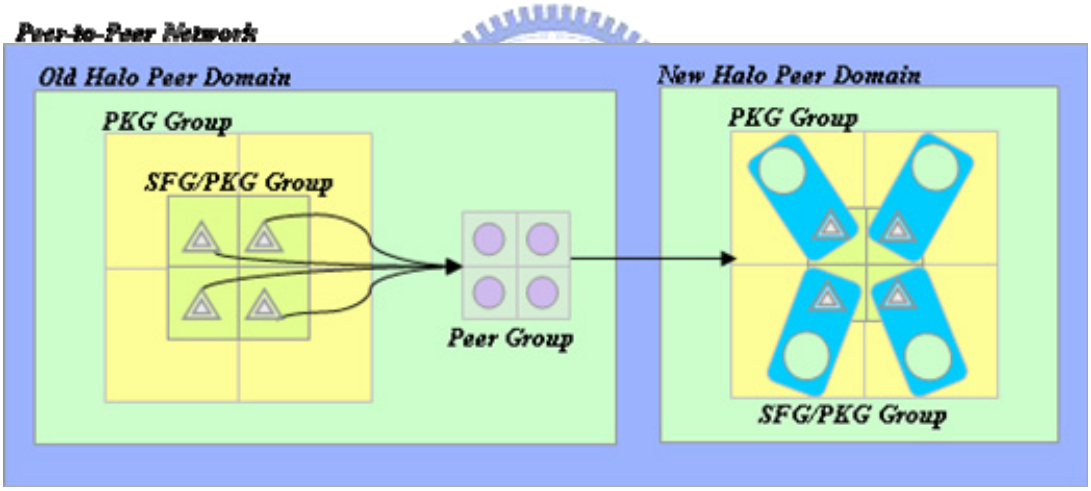

圖 18：Halo 系統中產生新同儕領域的示意圖

- 步驟一：一群舊領域同儕，其公鑰分別對應於舊領域的各隊，他們將組成新領域的 SFG/PKG 群組，我們稱他們為 SFG/PKG 候選群組。他們先決定新領域的名稱、新領域的允許政策和新領域的 RA_{new}，接著向舊領域的 RA_{old} 做認證，並取得舊領域 RA_{old} 認證通過的安全標記，其中包含新領域的名稱及新領域政策，其中新領域政策由舊領域的允許政策金鑰簽章保護。

- 步驟二：SFG/PKG 候群群組利用同儕網路同儕找尋服務 (peer discovery

service) ，在舊領域 SFG/PKG 群組中，每隊找一個 SFG/PKG ，共找 c 個 SFG/PKG ，新領域發起者向舊領域的 SFG/PKG 出示安全標記，並取得對應於該隊的身份識別金鑰對，和新領域的允許政策金鑰對。

- 步驟三：SFG/PKG 候選群組，假設每隊的代表人為 $Peer^c$ ，擔任 TC-DKG (§2.3.2.1) 中玩家的角色，決定第 c 個隊的部分祕密分享函式 f_l^c 以及部分祕密 s_l^c 。

- 步驟四：每一個新領域 SFG/PKG 群組中，隊的代表人 $Peer^c$ ，向舊領域和自己身份對應的 SFG/PKG 認證，取得該隊部分領域私鑰 S_l^c ，成為新領域該隊的第一個 SFG/PKG^c 也是 PKG^c 。新領域的 RA_{new} 向舊領域 t 個不同的 SFG/PKG 認證，並取得新領域中，對應於 RA 身份的金鑰對，至此，我們已完成新領域的創立。 ■

5.1.2 產生 Halo 中第一個同儕領域

在 *Halo* 系統初始的時候，沒有上層的 SFG/PKG 群組存在，所以第一個領域的允許政策需由候選 SFG/PKG 進行自簽領域政策保護，或是讓第一個同儕領域的允許政策包含系統中最小且最必要的項目，並公開之，則不需要自簽來保護。此外，每個候選 SFG/PKG 必需擔任 RA 的身份，進行認證並簽發金鑰的工作。請參考圖 19 及下面的步驟。

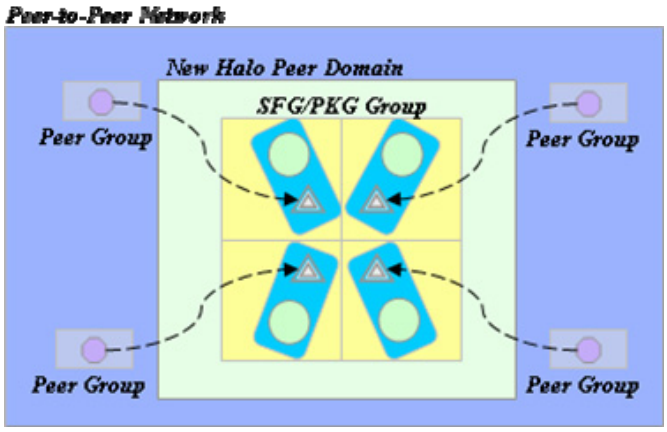

圖 19：Halo 中產生第一個同儕領域的示意圖

- 步驟一：一群同屬同儕群組的同儕，他們將組成初始領域的 SFG/PKG 群組，我們稱他們為 SFG/PKG 候選群組。他們先決定初始領域的允許政策，並依照他們選取的系統參數，決定分隊的方式與隊伍的數量。

- 步驟二：候選 SFG/PKG 中，每一隊派出代表人 $Peer^c$，擔任 TC-DKG (參考 4.3 節) 中玩家的角色，決定第 c 隊的部分祕密分享函式 f_l^c 以及部分祕密 s_l^c，成為新領域第一個 SFG/PKG^c 也是 PKG^c，至此，我們已完成 *Halo* 初始領域的創立。■

- 步驟三。初始領域 SFG/PKG 身兼初始領域 RA 的工作，彼此交換訊息，合作做出對應於 RA 的金鑰對，至此，我們已完成新領域的創立。■

5.2 產生分散式 SFG/PKGs 和 PKGs

經由系統初始化後，每個 SFG/PKG 群組及 PKG 群組中的隊伍，至少存在一個實體 (參與 5.1 節過程的同儕)，他們身兼 SFG/PKG 以及 PKG 的工作，他們接下來會去招募自己隊伍的 PKG，並給予用來產生下層同儕私鑰的部分祕密以及此層的領域政策。同儕可向新領域中，和自己同隊的新領域 PKG 認證，取得對應於身份識別的公鑰和對應的私鑰，並成為新領域的同儕。新領域的同儕可向新領域中，和自己同隊的 PKG 認證，取得和自己公鑰對應隊伍的部分祕密，成為該群組的 PKG。新領域的 PKG 可向新領域中，和自己隊伍相同的 SFG/PKG 認証，取得和自己公鑰對應到隊伍的部分祕密分享函式及部分祕密，成為該隊伍的 SFG/PKG，我們詳細的說明這些過程如下，並參考圖 20 及圖 21。

- 步驟一：同儕領域中的 PKG 也是候選 SFG/PKG，向領域 RA 做認證，RA 根據此領域的允許政策驗證，通過後，RA 給予該 SFG/PKG 安全標記，其中包含新的身份及 SFG/PKG 服務的資訊，此安全標記由 RA 簽章保護完整性。

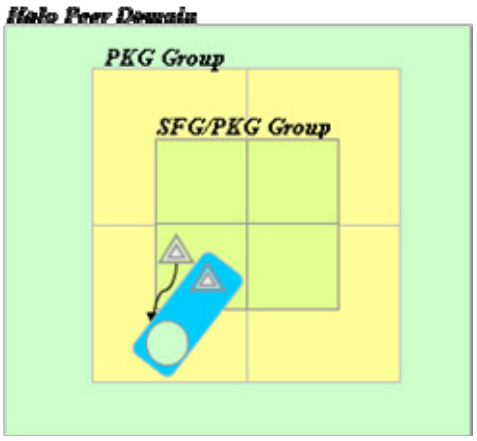

圖 20：Halo 中的 PKG 升格為 SFG/PKG 的示意圖

- 步驟二：候選 SFG/PKG 利用同儕網路同儕找尋服務，找尋領域中，和自己身份識 別對應的 SFG/PKG 認證：出示 RA 所給予的安全標記，並取得對應隊伍的部分祕密分享函式 f_l^c 及對應於 SFG/PKG 身份的金鑰對，成為該群組新的 SFG/PKG。■

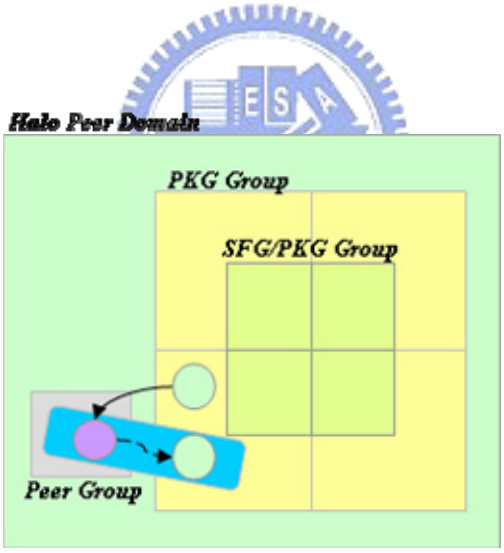

圖 21：Halo 系統中，同儕升格為 PKG 的示意圖

- 步驟一：同儕領域中的同儕，我們稱之為候選 PKG^c，向領域 RA 做認證，RA 根據此領域的允許政策驗證，通過後，RA 給予該候選 PKG 的安全標記，其中包含於 PKG 服務的身份，對應到的隊伍，此安全標記由 RA 簽章保護完整性。

- 步驟二：候選 PKG^c 利用同儕網路同儕找尋服務，找尋領域中，和自己同隊的 PKG 進行認證：PKG^c 出示 RA 所給予的安全標記，並取得對應隊伍的部分祕密 s_l^c，及對應 PKG 身份的金鑰對，成為該隊新的 PKG。 ■

5.3 同儕的認證及允許控制

5.3.1 家域的允許控制

同儕領域中的 RA 執行領域政策，當同儕要進入該同儕領域或是使用特別的服務，就需要符合這個領域政策。同儕向 RA 認證，通常透過本來親自出現或是出示相關的身份證明文件來證明自己。一旦通過，同儕所能執行的運算或是使用的服務將會寫進它的公鑰中並由 RA 簽章保護。當 PKG 看到此安全標示，先驗證其完整性，並根據安全標記上的記載，產生對應的私鑰。請參考圖 22 及下方的步驟。

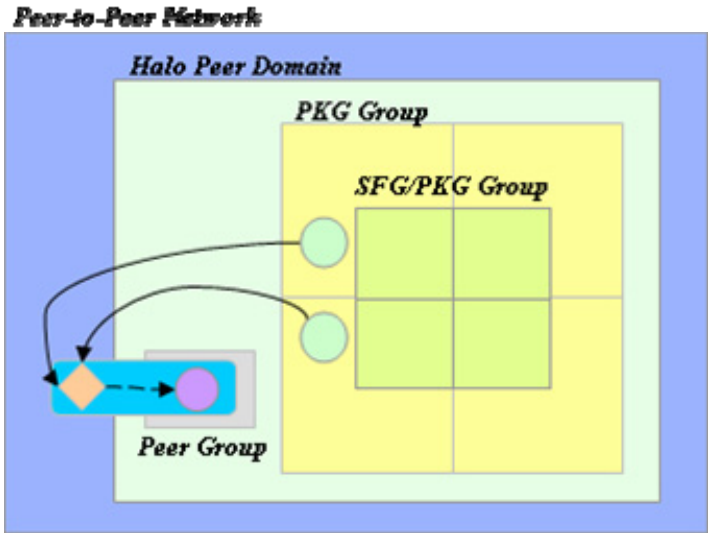

圖 22：網路節點進入 Halo 系統中的示意圖

- 步驟一：同儕網路中的網路節點，我們稱作候選同儕，向領域 RA 做認證，RA 根據此領域的允許政策驗證，通過後，RA 給予該候選同儕安全標記，其中包含新公鑰，其中包含使用 *Halo* 服務的資訊，並以 RA 簽章保護其完整性。

- 步驟二：候選同儕利用同儕網路同儕找尋服務 (peer discovery service) ，找尋領域中，和自己公鑰對應隊伍 (corresponding cohort) 的 PKG 群組。

- 步驟三：候選同儕向領域 PKG 出示 RA 所給予的安全標記，並取得對應隊伍

的部分使用者金鑰，若取得 t 種以上的部分祕密後，即可回復其私鑰 S_{peer} ，並進入該領域，成為該領域的同儕。■

5.3.2 外域的允許控制

當同儕擁有 *Halo* 系統中，家領域所發的私鑰，則此同儕能夠以此當作合法同儕的證據。外領域的 RA 利用 challenge-response 的機制認證此使用者，若能夠通過，則 RA 將同儕所能執行的運算，或是使用的服務將會寫進它的身份識別並以簽章保護。當 PKG 看到此安全標示，先驗證其完整性，並根據記載產生對應的私鑰。此外，我們也限制拿著外領域身份的同儕，無法成為該外領域 SFG / PKG 或是 PKG 的群組或是拿著外領域的身份再去換取其他領域的身份，以管理的策略，增加系統的安全性。請參考圖 23 及下方的步驟。

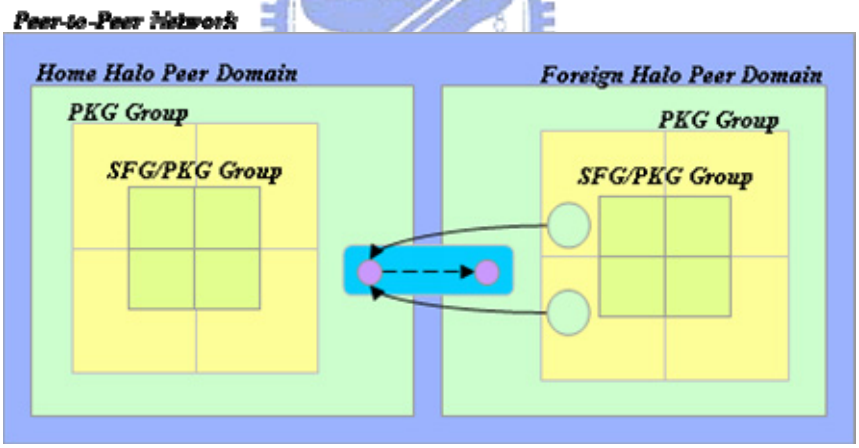

圖 23：Halo 系統中，家域中的領域進入外域的示意圖

- 步驟一：同儕領域中的同儕，我們也稱是外領域候選同儕，向外領域 RA 做認證，RA 根據外領域的允許政策驗證，驗證方式通常假設使用者在家領域有身份識別，它此導推出對應的公鑰，challenge-response，即可證明同儕是否為某領域中，合法的 *Halo* 系統使用者。

- 步驟二：驗證通過後，RA 給予該外領域候選同儕安全標記，其中包含對應

到新領域的身份識別，能夠使用外領域服務的資訊，此安全標記由外領域RA簽章保護完整性。

- 步驟三：候選外領域同儕利用同儕網路同儕找尋服務 (peer discovery service) ，找尋外領域中，和自己外領域公鑰對應隊伍的PKG群組。

- 步驟四：外領域候選同儕向外領域PKG出示外領域RA所給予的安全標記，並取得對應隊伍的部分使用者金鑰，若取得t種以上的之部分祕密後，即可回復其私鑰，並進入該領域，成為外領域的同儕。■

由此可知，在 *Halo* 系統中的使用者，會因為使用或是支援不同的服務，而有多個不一樣的公鑰及對應於隊伍的部分私鑰，像是SFG/PKG即有身份為SFG/PKG、PKG和一般使用者三種身份，所以也會有三組公私鑰對，若有更多的服務支援或是使用權，則會產生更多的公鑰及對應的私鑰。若要得知該同儕是否擁有其宣稱的服務，可以用 challenge-response 的方式確認。

此外，在系統的組成方面，*Halo* 系統所需管理同儕個數 (包含SFG/PKG和PKG) ，其充分條件為：每個領域至少需要和隊伍個數一樣多的SFG/PKG，所以若系統中有k個同儕領域，每個領域有c個隊伍，則最少需要$k \cdot c$個管狂同儕。但為了增加系統的可利用性，避免管理同儕消失，所以每個隊伍中的管理同儕將會產生化身，我們建議一個同儕領域中，至少要有一個專職PKG和兩個SFG/PKG，因SFG/PKG同時也能擔任PKG的工作，所以整體來看，我們在領域中會有三個PKG和二個SFG/PKG。此外，若要考量使用者找尋管理同儕的效率，則管理同儕數量則需要更多，若同儕搜尋服務以 Chord 為基礎，則我們必需要放置$\log n$ (n為同儕領域的大小) 個PKG來進行允許控制，若以大小為$10K$的同儕領域，則只需要$4 \cdot 3 \cdot c$個管理同儕，c值大約是6 ± 2，所以需要不到1%的同儕，是非常符合系統運作的需求。 ■

第 6 章 系統比較

和我們系統出發點相似，用來解決在同儕網路中金鑰系統的部署問題，就屬 Threshold RSA (TRSA) ，此節我們將比較我們的設計和他們的相同處和相異處，並指出我們的設計優於 TRSA，更適合在同儕網路中部署我們的金鑰系統，最後詳細列出 *Halo* 中各項動作所需要的運算複雜度以及通訊複雜度。

6.1 分散式 RSA 設計

門檻式的 RSA (其中組成元件為門檻式憑證權威 (threshold certificate authority, TCA) 設計能讓簽章能夠分散式的被一群人所簽署，而簽署的人大於門檻就能產生合法的憑證，少於門檻的人數則無法產生。這個設計也常被稱作 t 取 n 的門檻式簽章系統，n 為系統中的成員，而 t 是產生合法憑證最低所需的人數。目前的設計致力於將標準的簽章：如 RSA、DSS 等，加入門檻簽署的性質，這樣的好處是，經過這個設計的步驟後，產生的是標準的憑證，而驗證方式和原來的 RSA 設計一樣，跟門檻設計時，多少人簽署、簽章被產生的演算法都沒有關係。通常這樣的設計將簽章金鑰 (signing key) 拆成多個等份，讓每個擁有部分簽章金鑰的人產生部分憑證，當使用者蒐集門檻個數個部分憑證，即可藉由系統設計的演算法，回復完整的憑證。最新的設計在[23]被提出，他們延伸 Shoup[19]的設計，並應用於 ad-hoc 網路中。我們將簡要的介紹 TRSA，並詳細的說明如何利用 TRSA 達到同儕網路中管理階層的需求：

- 系統金鑰分享：可信任的莊家選定 RSA 公開參數 N, e，並計算出私鑰為 d，接著使用 Shamir 的設計，向 TCA 群組分享的部分祕密

- 部分簽章計算及回復完整憑證：TCA 利用部分祕密進行部分簽章，主要用到找最大公因數法、Lagrange 內插法以及廣展的尤拉演算法 (extended Euler's

algorithm) ，因為我們要比較的是系統金鑰如何部屬，所以我們將抽像化這兩個步驟，以黑盒子的方式來描述。

接下來將說明 TRSA 用於同儕網路中產生新階層的方式：

- 步驟一：舊領域 TD_{old} 選定新領域的 TD_{new} ，TD_{new} 利用 RSA 公開參數 N ，並計算出公私鑰對為 (e_1, d_1) ，將私鑰 d 拿給舊領域的 TCA 認證並取得其憑證 $Cer(d_1)$ 。

- 步驟二：TD_{new} 接著使用 Shamir 的設計，依隊的方式在 TCA 群組中分享的部分祕密，第 c 隊得到 d_1^c ，並給予完整祕密的簽章 $Cer(d_1)$ 。

- 步驟三：拿到祕密的 TCA，則可以幫使用者簽發部分公鑰憑證，使用者取得門檻個數的公鑰憑證，即可回復完整的公鑰憑證。

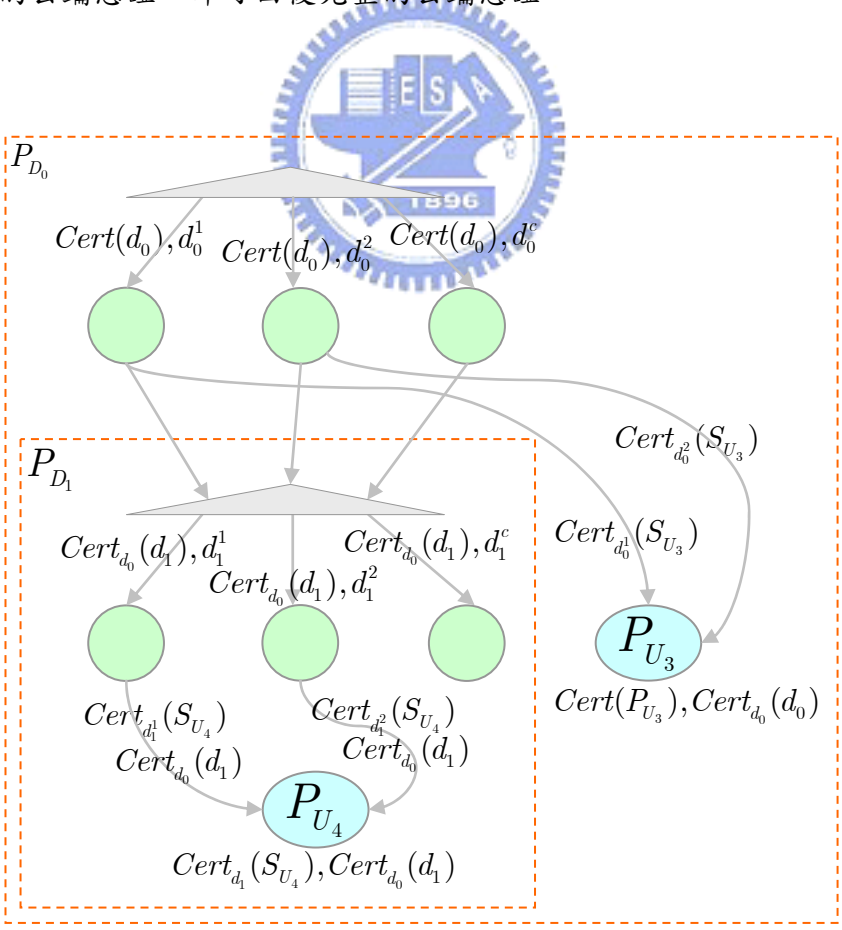

圖 24：TRSA 用於多階層系統的示意圖

6.2 分散式 RSA 設計與 Halo 系統的比較

根據§6.1 **錯誤! 找不到參照來源。**節的討論，我們可以了解 TRSA 在每個階段假設有可信任的莊家存在，在我們討論 *Halo* 系統機制時 (請參考 4.2 的討論) ，我們亦先假設每個領域中有可信任莊家的存在，利用它來分派部分系統金鑰。在分析之下，這個過渡版本的 *Halo* 系統和 TRSA 使用的是類似的技術，用來拆開系統的金鑰，並分派 TCA 部分金鑰，所以在系統初始步驟和計算上，複雜度是接近的。

但在金鑰的使用上，TRSA 使用部分金鑰產生部分公鑰憑證，而 *Halo* 系統則是產生同儕的部分金鑰，雖然同儕的憑證也能夠由同儕網路提供的搜尋服務找到，但由於下列三個原因，讓我們想要避免使用憑證：

- 在 PKI 中，憑證的產生已經非常複雜，若要在同儕網路中，甚至是分散式的產生憑證，困難度更是增加許多。
- 在我們機會合作的前提下，需求者通常是找服務，而非找人，也就是做一種能力的搜尋：我們在同儕網路中會和某人合作，合作的同儕剛好路過，而且我也需要這個服務，所以你就成了有能力的同儕，所以我邀請你一起合作。
- 在同儕網路中，分散式的資料庫通常不是用來搜尋特別的同儕，而是提供服務的地方，就像下載影片時，是用片名來做搜尋，而不是大盤商的名字來找。

因為 *Halo* 系統為一身份基礎金鑰系統，所以同儕能夠使用的服務是寫於其公鑰中，對方可以藉由 challenge-response 的方式，知道同儕是否屬於某個領域，或是真的具備某種服務，因而除去憑證的需求。

最終我們利用更強的方式取代原先需假設的可信任的莊家 (請參考 4.3 討論) ，但同樣的方式卻無法套用於 TRSA，原因是在於 RSA 公鑰系統設計中所用的金鑰對(e,d)，需要有一個很緊密的數學關係存在 (在$\phi(N)$中互為反元素) ，在同

儕間分散式的選取過程中，是很困難達成這樣的關係，但是在 *Halo* 的設計中，因為屬於身份基礎公鑰系統，設計中所用的金鑰對(P_i, sP_i)，只要讓 s 能夠在橢圓曲線的 order 範圍內，即為一個合法的金鑰對。所以在 *Halo* 系統能夠進一步的將權威拆開，分散式的決定產生系統的祕密，而 TRSA 則無法做到，這也是 *Halo* 較 TRSA 更適合同儕網路的最大原因。

下表將 TRSA 與 *Halo* 的比較總結於下表：

	TRSA	*Halo* (THIBE)
系統初始	需要可信任莊家分發部分系統金鑰，讓 CA 共享系統金鑰	無需可信任莊家，同儕間合作決定系統金鑰且共享此金鑰
金鑰抽取		同儕需向 t 不同隊伍的 PKG 認證，取得 t 種不同的部分金鑰
憑證簽發	同儕需先後向 t 個 CA 認證，並接受 CA 簽於金鑰的部分憑證	
階層擴充	需要可信任莊家分發部分階層金鑰，讓 CA 共享階層金鑰	無需可信任莊家，同儕間合作決定階層金鑰且共享此金鑰
權威加入	同儕需先後向 t 個 CA 認證，並接受 CA 的給予的部分金鑰 (部分憑證的產生方法會改變)	PKG 需向一個和自己同隊的 SFG / PKG 認證，並取得該隊的祕密；同儕需向一個和自己同隊的 PKG，並取得該隊的祕密
THIBE 的優點	◆ 需公鑰憑證證明公鑰之有效性，去除憑證管理的工作：*Halo* 系統中的成員，能夠藉由推導出的公鑰加密文件，做 challenge- response 來得知金鑰的有效性。 ◆ 使用 *Halo* 服務的同儕能夠藉由單一的演算法運算 (如解密或是簽章驗證) 來得知整個信任鍊 (chain-of-trust) 的真偽，而不需要像 TRSA 一層層往上驗證。	

最大優勢	◆ 在 *Halo* 系統中的把關者（*SFG* / *PKG* 或 *PKG*）能夠以分散且門檻的方式來實現，如此大大的增加系統金鑰的安全性； ◆ 以隊的方式管理群組金鑰，也避免階層的部分祕密，因同儕的離開而消失，如此增加系統的可利用性以及避免單點失敗的情況。

表格 5：Halo 與 TRSA 的比較表

6.3 Halo 系統運作複雜度分析

我們分析的是 PKI 在管理上的複雜性，因為實現系統時，選用的不同系統與函式庫，所以複性會有很大的差異，所以我們以同儕網路中基本的運算來計算，以下將先介紹同儕網路中的基本運算：*Halo* 系統主要的動作分成五大類：(一) 向 RA 註冊 (registration) 。 (二) 向 *SFG* / *PKG* 或 *PKG* 認證 (authentication) 。 (三) 向同儕認證 (authenti- cation) 。 (四) 搜尋資源 (resource discovery) 。 (五) 分散式金鑰產生 (distributed key generation, DKG) 。

◆ 向 RA 註冊：通常是一個實體的過程 (physical process) ，經由手動設定 (manual setup) 來完成：申請者必需現身於 RA 櫃台，並提供足夠的合法身份識別資料，如身份證、學生證...等，此外，RA 也確認申請者是否有權使用某些服務，申請者也需提供相關的資料，如會員證明、繳費收據等，RA 確認申請者身份及被授與的服務，確認無誤後，將實體身份識別資料轉換成電子身份識別，並以自己的私鑰簽章保護，寫於安全標記中。

◆ 向 *SFG* / *PKG* 或 *PKG* 認證：因為 RA 所簽發的電子識別，唯一對應 *Halo* 系統的一把公鑰，*PKG* 能夠藉由安全標記上，簽發者的簽章是否有效，確認申請者的電子身份識別及公鑰，通過後，*SFG* / *PKG* 或 *PKG* 產生對應公鑰的私鑰。

◆ 向同儕認證：當要確認同儕是否為本人，且具備某種服務，驗証同儕能利用 challenge- response 來確認被驗證同儕的身份，驗證者先推導出可能的公鑰，*Halo*

加密法加密亂數給被驗證的同儕，若被驗證的同儕具備這樣的身份，則他必有對應的私鑰，所以能夠解開密文，並回傳 response。

- 搜尋資源:在同儕網路中，資源的搜尋通常以 distributed hashed table (DHT) 來達成，如 CAN，Chord…等，有些甚至加入有限的漫遊 (limited random walk) ，一方面保留 DHT 的快速搜尋，一方面加強遇到失敗的 DHT 搜尋的對策。

- 分散式金鑰產生：請參考§2.3.2.1 的分析。

6.3.2 通訊複雜度

由下表得知，在產生新的領域時的複雜度最高，即為 DKG 運作時所需的溝通管道，但因為領域的產生不會經常發生，而且之後的動作，包含管理同儕的自我維護，或是家域、外域的同儕允許控制，複雜度都是相當的低，如 SFG/PKG 和 PKG 產生化身的過程，僅需傳遞必要的祕密，而允許控制則是和系統最初設捕的門檻有關，通常門檻值在5附近。所以也是相當的便宜。

Halo 系統動作	參與交易的實體	完成動作所需通道
產生新的同儕領域	c 個同儕 (全部是誠實的)	$2c$ 個廣播通道 $c(c-1)$ 個安全單播通道
	c 個同儕 (其中有說謊的)	$c(t+3)$ 個廣播通道 $c(c-1)$ 個安全單播通道
擴增分散式 SFG/PKG	1 個同隊本域 SFG/PKG 與 SFG/PKG 候選人	2 個單播通道
擴增分散式 PKG	1 個同隊本域 PKG 與 PKG 候選人	2 個單播通道

本域允許控制	t 個本域 PKG 與本域同儕候選人	$2t$ 個單播通道
外域允許控制	t 個外域 PKG 與外域同儕候選人	$2t$ 個單播通道

表格 6：Halo 中各試動作所需的通道複雜度

6.3.2 運算複雜度

由上面的表格得知，在產生新的領域時有兩種情形，若能保證參與領域產生的同儕階誠實，則複雜度可以降低許多。和分析溝通管道的討論相同，雖然領域產生的複雜度較高，但因為領域的產生不會經常發生，而且之後的動作，包含管理同儕的自我維護，或是家域、外域的同儕允許控制，複雜度都是相當的低，如 SFG/PKG 和 PKG 產生化身的過程，僅需安全傳遞必要的祕密，而允許控制和系統最初設定的門檻值有關，需要 t 個 PKG 產生對應於使用者身份識別的金鑰。

Halo 系統動作	參與交易的實體	動作所需運算量	章節
產生新的 同儕領域	c 個同儕 (全部是誠實的)	TC-DKG (=KD+KG)	§2.3.2.1
	c 個同儕 (其中有說謊的)	TC-DKG (=KD+KV+KC+KG)	
擴增分散式 SFG/PKG	1 個同隊本域 SFG/PKG	1 HIBE 加密	§2.2.2
	SFG/PKG 候選人	1 HIBE 解密	
擴增分散式 PKG	1 個同隊本域 PKG	1 HIBE 加密	
本域允許控制	PKG 候選人	1 HIBE 解密	

	t 個本域 PKG 本域同儕候選人	t HIBE 金鑰抽取	§2.2.2
外域允許控制	t 個外域 PKG 外域同儕候選人	t HIBE 金鑰抽取	

表格 7：Halo 系統各項動作所需運算量

第 7 章 實作議題

7.1 密碼學方面的考量

7.1.1 密碼系統參數選擇

根據美國國家標準局 (National Institute for Standards and Technology, NIST) , 1024 位元長度的參數的 RSA 和 Diffie-Hellman，在 2010 年前已相當的足夠。NIST 也提供了不同密碼系統建議的金鑰長度。請參考下表。所以就金鑰長度來說，ECC 密碼系統在相同的安全階級下，有著較短的金鑰和系統參數長度。此外，就如同其他的公開金鑰密碼 (Public-key Cryptography) 系統，加、解密相對於對稱式加密系統 (Symmetric-key Cryptography) ，依舊很費時費功，所以我們也採用混合式的方法：公開金鑰系統讓金鑰管理和電子簽章有效率，而對稱式金鑰系統讓大量的資料能夠快速的加並保有其完整性。而且在 session 中，若更多的訊息需要安全的交換，我們能夠利用交換加密過或簽章過的 ephemeral key (參考 §4.1) ，如此能夠達到簡單的 Diffie-Hellman 金鑰協議。若需要更高的安全，簽後 (Sign-then-encrypt) 加密或是簽密 (Signcrypt) ephemeral key 也能夠加進安全服務中，讓金鑰協議更加安全。

Elliptic Curve Key Size	160 bit	224 bit	256 bit	384 bit
RSA Key Size	1024 bit	2048 bit	3072 bit	7680 bit

圖 25：NIST 建議 ECC 與 RSA 金鑰長度的換算

7.1.2 橢圓曲線及雙線性對選擇

在 Pairing-based 的密碼系統中，如何找到一個橢圓曲線的子群，其 embedding degree 大到能夠抵擋 Fery-Ruck attack，但小到能夠讓 Tate pairing 能夠有效率的運算。在一般的橢圓曲線中，embedding degree 通常很大。到目前唯一能夠找到

擁有合理大小的 embedding degree 的曲線，是來自超橢圓曲線 (supersingular curves) 。但因為這些曲線是建構在較小的 characteristic，讓他們很容易受到離散對數 (discrete logarithm) 演算法的攻擊。所以在實作上，雖然有些許攻擊，但此攻擊仍很難成功，所以我們仍傾向在 *Halo* 中，使用超橢圓曲線，讓系統能夠在足夠的安全理論支持下，有效率的運作。一些常用的超橢圓曲線及其 pairing 如下表所示：

k	Supersingular Elliptic Curve
2	$E: y^2 = x^3 + a$ over $\mathrm{F}_p, where\ p \equiv 2 \pmod 3$ $Distortion\ map\ (x,y) \mapsto (\zeta_3 x, y), where\ \zeta_3^3 = 1$
2	$E: y^2 = x^3 + x$ over $\mathrm{F}_p, where\ p \equiv 3 \pmod 4$ $Distortion\ map\ (x,y) \mapsto (-x, iy), where\ i^2 = -1$
3	$E: y^2 = x^3 + a$ over $F_{p^2}, where\ p \equiv 5 \pmod 6$ $a \in F_{p^2}, a \notin F_p\ is\ a\ square\ which\ is\ not\ a\ cube$ $Distortion\ map\ (x,y) \mapsto (x^p / \gamma a^{(p-2)/3}, y^p / a^{(p-1)/2}), where\ \gamma^3 = a, \gamma \in F_{p^6}$
4	$E_i: y^2 + y = x^3 + x + a_i$ over $F_2, where\ a_1 = 0\ and\ a_2 = 1$ $Distortion\ map\ (x,y) \mapsto (u^2 x + s^2, y + u^2 s x + s),$ $where\ u \in F_{2^2}\ and\ s \in F_{2^4}\ satisfy\ u^2 + u + 1 = 0\ and\ s^2 + (u+1)s + 1 = 0$
6	$E_i: y^2 = x^3 - x + a_i$ over $F_3, where\ a_1 = 0\ and\ a_2 = -1$ $Distortion\ map\ (x,y) \mapsto (\alpha - x, iy),$ $where\ i \in F_{3^2}\ and\ \alpha \in F_{3^3}\ satisfy\ i^2 = -1\ and\ \alpha^3 - \alpha - a_i = 0$

圖 26：常用的超橢圓曲線及其上的 pairing 運算

7.1.3 HIBE 軟體實作結果

我們實作了 Gentry-Silverberg 的階層式身份基礎金鑰系統，測試跑在桌上型電腦，配備為 AMD Sempron 1.64 GHz 和 1.0G RAM。實作項目包含直接版本 HIBE/HIBS 和雙重版本 HIBE/HIBS，並使用 elliptic curve arithmetic API[7]. 執行

的時間以 millisecond (ms) 計算。

圖 27 中，下三角中的數值代表執行直接版本 HIBE 加密所需的時間，而上三角袋表執行直接版本 HIBE 解密所需的時間，第一行及第一列分別袋表加密者及解密者所在的階層，例如顯示 708/2328 的格子，表示位於第一層的加密者需要 708 ms 加密文件給解密者，解密者需要 2328 ms 還原回明文。以此類推，圖圖 29 顯示 HIBS 簽章及驗證所需的時間，顯示 108/2657 的格子，表示俛於第二層的簽章者需要 108 ms 準備密文，而驗證者需要 2658 ms 驗證是密文是否通過。雙重版本的 HIBE 及 HIBS 顯示於圖 29：雙重版本 HIBE 實作效能測試　圖 30：雙重版本 HIBS 實作效能測試，其中我們固定解密者及驗證者在第四個階層，如第一列的數字代表共同的祖先，如圖三中，顯示 2421/1596 的格子代表加密者位於第三層，需要用 2421 ms 加密文件給位於第四層的解密者，兩者的共同祖先位於第二層。我們可以藉由比較圖 27 和圖 29，了解雙重版本帶來的好處，原先階層三送到階層四的時間為 3458/889，但若他們共同的組先在階層二，則時間變成 2421/1595，能夠讓雙方平均分攤計算時間，且雙方合計所需的時間總合，也較直接版本來的少，符合了使用雙重版的的情境：加密解密者間的距離若小於解密者到 root 的距離。

non-dual HIBE q=160 bit		Receiver Level			
		1	2	3	4
Sender Level	1	1203 / 594	2328 / 708	2865 / 781	3459 / 897
	2	1203 / 610	2348 / 712	2875 / 789	3455 / 880
	3	1203 / 609	2312 / 704	2875 / 781	3458 / 889
	4	1204 / 610	2345 / 705	2874 / 788	3451 / 889

non-dual HIBS q=160 bit		Receiver Level			
		1	2	3	4
Sender Level	1	2054 / 115	2031 / 109	2047 / 109	2036 / 110
	2	2657 / 109	2657 / 110	2656 / 114	2672 / 109
	3	3328 / 109	3382 / 109	3311 / 111	3312 / 113
	4	4015 / 111	4046 / 105	4060 / 114	4031 / 108

圖 27：直接版本 HIBE 實作校能測試　圖 28：直接版本 HIBS 實作校能測試

dual HIBS q=160 bit		Receiver Level = 4 Common Ancestor Level 1	2	3
Sender Level	2	1671 / 750	N/A	N/A
	3	2234 / 1359	1672 / 749	N/A
	4	2875 / 1971	2312 / 1371	1766 / 745

dual HIBE q=160 bit		Receiver Level = 4 Common Ancestor Level 1	2	3
Sender Level	2	2375 / 1593	N/A	N/A
	3	2360 / 2182	2421 / 1595	N/A
	4	2287 / 2734	1782 / 2156	1203 / 1579

圖 29：雙重版本 HIBE 實作效能測試　圖 30：雙重版本 HIBS 實作效能測試

Pairing 運算在不同金鑰長度的密碼系統下所需的時間，顯示如下：

Elliptic Curve Key Size	160 bit	224 bit	256 bit	384 bit
RSA Key Size	1024 bit	2048 bit	3072 bit	7680 bit
Pairing Operation	609 ms	1282 ms	1703 bit	5100 bit

表格 8：Pairing 在不同金鑰長度密碼系統的效能測試

7.2　同儕網路部屬方面的考量

我們將以 JXTA 平台，探討同儕網路環境實際部署的問題，包括如何將我們的設計整合進去，我們設計中的運作對應到平台中的運算，以及身份命名的對應關係，都在接下來的章節一一討論：

7.2.1　補強 JXTA 安全支援

因為在同儕網路中沒有指定的伺服器擔任信任權威 (trust authority) ，JXTA 讓每個同儕產生自己的金鑰和對應的憑證 (certificate) ，也就是說，每圈同儕充當自己的權威。這個安全機制稱作自簽憑證 (self-sign certificate) 。它只能保證金鑰

對的有效性 (是否成對) ，但是沒有提供任何關於擁有者的資訊。所以讓有心人取得他人的金鑰來使用，系統是無法察覺的。為了要保證安全的通訊，我們需要公開金鑰系統所提供基礎的安全服務：加密、簽章和金鑰協議 (key agreement) 。我們設計的系統 *Halo* 能夠強化 JXTA 在安全通訊上的弱點。我們將延伸 JXTA 成員服務 (membership service) ：需要重新定義其中的認證器 (authenticator) 、安全憑證 (secure credential) 來支授我們設計的階層式身份基礎金鑰系統。

7.2.2 JXTA 通訊上的機制

JXTA 中的管道 (pipe) 是同儕間虛擬的通道，雖然我們把同儕間的通訊想成單一的連線，但其中的防火牆 (firewall) 或是閘道 (gateway) 等障礙物，讓同儕間無法直接連線，所以管道就是為了解決這個問題的虛擬通道，利用閘道點 (gateway peer) 來支援接繼通訊 (relayed communication) 。

同儕和同儕間實際上連線通訊時，極有可能牽涉其他的同儕，而管道這個抽象化概念，把其間的複雜性隱藏，所以 JXTA 協定支援了許多不同型式的管道如單向非同步 (uni-directional synchronous) 、單播可信賴安全管道 (uni-cast reliable secure pipe) ，當然也提供了不同的定址方式如點對點 (point-to-point) 和擴散 (propagate) ，其中擴散方式能夠達到廣播通訊的校果。

在 *Halo* 系統的設計中，在新的同儕領域產生時，最初的幾個同儕 (候選 SFG/PKG) 會需要用到擴散的方式交換彼此的資訊，而同儕領域產生後，PKG 和同儕領域中的同儕則可以藉由單向同步，甚至是單播可信賴的安全管道來進行安全的交易。

7.2.3 JXTA 中的身份與密碼學中身份的連結

在 JXTA 中，通告 (advertisement) 是用來描述有關同儕、同儕群組和服務的資

訊。同儕群組通告 (請參考表格 9) 包含 JXTA 同儕群組 ID (GID) ，用來唯一參考到一個同儕群組。JXTA 模組規格 ID (Module Specification ID, MSID) ，用來唯一參考到同儕群組的機制、名稱、敘述 (Desc) 和群組服務 (Svc) 。為了將我們的設計整合進去，我們在群組敘述中，加入了一群參數，例如 GID, MSID, Hash 函數和密碼系統中所需的系統參數。這邊的 GID 和先前提到的 JXTA GID 不同。他是將 JXTA GID, JXTA MSID 連接 (concatenate) 起來表示我們系統中的同儕群組 ID。同儕群組的公鑰則是由群組 ID 的 Hash 值取得。請見表格 9。

同儕通告 (請參考表格 10) 包含了 JXTA 同儕 ID (PID) ，用來唯一參考到唯一的同儕 (GID, 名稱, 敘述和其他服務的資訊) 。就如同我們修改 JXTA 同儕群組通告一樣，我們在敘述中加入了一群參數來整合我們的系統。我們系統中的 PID 將改為 JXTA PID, JXTA GID 和 JXTA MCID 連接，這個 ID 將會唯一指到同儕群組所提供的服務。它意指同儕能夠在特定的群組中存取特定的服務。而同儕使用的公鑰，則是由 PID 的 hash 值得到。

```
<!DOCTYPE jxta:PGA>
<jxta:PGA xml:space="default" xmlns:jxta="http://jxta.org">
     <GID>
     urn:jxta:uuid-425A5C703CD5454F9C03938A0D65BD
     </GID>
     <MSID>
     urn:jxta:uuid-DEADBEEFDEAFBABAFEEDBABE01ACEF
     </MSID>
     <Name>          Peer Group (Gossip) </Name>
     <Desc>
          <GID>      Peer Group ID                 </GID>
          <MSID>     Module Specification ID       </MSID>
          <G1>       Additive Group (G1)           </G1>
          <G2>       Multiplicative Group (G2)     </G2>
          <e>        Pairing (e)                   </e>
          <P0>       Additive Group Generator (P0) </P0>
          <Hash1>    Bits-to-G1 Hash Function      </Hash1>
          <Hash2>    G2-to-Bits Hash Function      </Hash2>
          <Sig>           Parameter Set Signature  </Sig>
     </Desc>
     <Svc>
          <MCID>
     urn:jxta:uuid-4CD1574ABA614A5FA242B613D8BAA3
          </MCID>
          <Parm type="jxta:GossipServiceConfigAdv"
               xml:space="preserve"
               xmlns:jxta="http://jxta.org">
               <gossip> test </gossip>
          </Parm>
     </Svc>
</jxta:PGA>
```

表格 9：同儕群組廣告的範例

```
<!DOCTYPE jxta:PA>
<jxta:PA xml:space="default" xmlns:jxta="http://jxta.org">
     <PID>
     urn:jxta:uuid-59616261646162614A787461503250
     </PID>
     <GID>
     urn:jxta:uuid-425A5C703CD5454F9C03938A0D65BD
     </GID>
     <Name> Peer Name (test) </Name>
     <Desc>
          <PID>     Peer ID                              </PID>
          <MCID>    Module0 Component ID                 </MCID>
          <G1>      Additive Group (G1)                  </G1>
          <G2>      Multiplicative Group (G2)            </G2>
          <e>       Pairing (e)                          </e>
          <P0>      Additive Group Generator (P0)        </P0>
          <Hash1>   Bits-to-G1 Hash Function             </Hash1>
          <Hash2>   G2-to-Bits Hash Function             </Hash2>
          <Sig>          Parameter Set Signature         </Sig>
</Desc>
     <Svc>
          <MCID>
     urn:jxta:uuid-DEADBEEFDEAFBABAFEEDBABE0805
          </MCID>
          <Parm>
               <jxta:RA xml:space="preserve"
                       xmlns:jxta="http://jxta.org">
               <DstPID>
     urn:jxta:uuid-59616261646162614A787461503250
               </DstPID>
          </Parm>
     </Svc>
</jxta:PA>
```

表格 10：同儕廣告的範例

7.2.4 JXTA 服務搜尋的機制

在 *Halo* 系統的設計中，使用者會先向同儕領域的 *PKG* 認證，以成為領域的成員，而之後又能藉由 *PKG* 的幫助，成為該同儕領域的 *PKG*，最後藉由同儕領域 *SFG / PKGs* 的認證而成為該同儕領域的 *SFG / PKG*。其中的關鍵就是如何找到能夠認證的人，如何確定此人有權力能夠認證。在 JXTA 中，每個同儕，甚至是同儕群組都有描述自身的通告 (詳見§7.2.3)。在 JXTA 中，會有一群名叫 rendezvous 的同儕群組，這些點能夠儲存大量週邊同儕的資訊，所以藉由這些點來維維整個同儕網路的狀態和儲存的資訊，所以在 *Halo* 系統中的使用者，能夠藉由向這些點要求所需同儕的通告，並進行接下來的交易。

第 8 章 結語

我們提出了 *Halo* 系統，一個階層式的身份基礎公開金鑰系統 (HIDPKI) 。它不但支援遞迴式的方式建立階層式 IBE 架構，並且提供無需固定伺服器的方式產生 HIBE 參數和使用者私鑰，此外，我們也提出了系統管理方面的策略，較傳統 TRSA 更有效率，且更適合同儕網路的場景。

8.1 研究成果

一、在同儕網路中，我們使用身份基礎金鑰系統，避免了憑證簽發與維護的困擾。我們也加強了系統金鑰的安全性，以分散且門檻式的方法分散保存，一方面避免單一同儕擁有完整秘密，使該點需要比平常高出許多的安全保護，一方面也增強抵擋惡意權威的能力，在少於門檻數量的同儕被破解或是共謀，是無法得知任何有關系統秘密的資訊。

二、在同儕網路中，我們使用階層式的金鑰系統，並設計其上的管理架構，不僅支援網路多層次的需求，也加入了管理服務階層的需求，讓同儕網路中各項服務能夠獨立管理與授權，分工合作，加強同儕網路系統機會性合作的優勢。

三、在同儕網路中，我們引進分隊與晉升的機制，讓同隊的管理同儕擁有相同的部分秘密，而且同儕能夠經認證而變成管理同儕新的化身，一方面分攤同隊同儕的工作量，增加同儕的可取得性 (availability) ，另一方面也避免單點失敗，同一份資料有多個同儕來維護。

四、同儕躍升為新權威的過程、網路節點進入 *Halo* 系統的過程，*Halo* 系統初始化及新的同儕領域的產生的動作，我們分析這些運算與通訊的複雜度，並與 TRSA 做比較，說明 Halo 系統在同儕網路中，較 TRSA 更具競爭力。

五、Halo 系統管理同儕領域的複雜度很低，所有的管理同儕(SFG / PKG 和 PKG)總數不到全體同儕的 1%，且增加層數，管理同儕以線性增加(約是和隊伍數量一致)，是一個非常實際且易擴充的設計。

8.2 未來方向

- 密碼學方面：

一、 尋找安全性和效率最佳的取捨 (trade-off) 的身份基礎金鑰系統設捕。

二、 加速橢圓曲線上加法群純量積及 pairing 運算的速度。

三、 擴充必要的安全機制，像是更安全的金鑰協議、群組金鑰等。

- 同儕網路系統方面：

一、在 JXTA 平台上設計 *Halo* 所需的機制，如命名方式，通訊方式...等。

二、在 JXTA 平台上實現 *Halo* 的原型，設計成可附加 (add-on) 的服務模組。

三、利用身份基礎的金鑰系統中身份與服務的連結，考量 Halo 系統整合能力基礎 (capability-based) 的允許控制的可行性。

參考文獻

[1] A. Shamir, *Identity-based cryptosystems and signature schemes*, in Advances in Cryptology – Crypto 1984, Lecture Notes in Computer Science 196 (1984)

[2] B. Lehane, L. Doyle, *Shared RSA Key Generation in a Mobile Ad Hoc Network. IEEE Military Communications Conference (Milcom)* , 2003.

[3] C. Gentry, A. Silverberg. *"Hierarchical ID-Based Cryptography"*. ASIACRYPT 2002, Springer-Verlag, LNCS #2501 (2002) .

[4] C. Tang, A. Chronopoulos, "*An Efficient Distributed Key Generation Protocol for Secure Communications with Casual Ordering*". IEEE Proceedings of the 2005 International Conference on Parallel and Distributed Systems (ICPADS'05)

[5] D. Boneh, M. Franklin, "*Identity Based Encryption from Weil Pairing*", Advances in Cryptology-ASIACRYPTO'01, pp.514-532.

[6] D. Boneh, M. Franklin. "*Efficient Generation of Shared RSA Key*s". Crypto'97, pp. 425-439, LCNS-1294, 1997.

[7] Elliptic Curve Arithmetic API, provided by Computer Security and Cryptography Group, Dept. of CS, NUIM, http://www.crypto.cs.nuim.ie///software/

[8] F. Dabek, M. Frans Kaashoek, D. Karger, R. Morris, I. Stoica. "*Wide-area Cooperative Storage with CFS*". 18th ACM Symposium on Operating Systems Principles (SOSP'01) , Lake Louis, Banff, Alberta, Canada, 2001.

[9] J. Horwitz and Ben Lynn, *Toward Hierarchical Identity-Based Encryption*, Proceedings of Eurocrypt 2002, LNCS 2332, pp. 466-481, 2002

[10] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R.

Gummadi, S. Rhea, H. Wetherspoon, W. Weimer, C. Wells, B. Zhao. "*OceanStore: Architecture for Global-Scale Persistent Storage*". ACM ASPLOS 2000, Cambridge, MA, USA.

[11] JXTA™ Community Projects https://jxta.dev.java.net/

[12] L. Owens, A. Duffy and T. Dowling, *An Identity Based Encryption System*, International Conference on the Principles and Practice of Programming in Java, 2004

[13] M. Castro, P. Druschel, A.M. Kermarrec, A. Nandi, A. Rowstron and A. Singh. "SplitStream: High-bandwidth Content Distribution in Cooperative Environment". 20th ACM Symposium on Operating Systems Principles (SOSP'03) , Lake Bolton, New York, 2003.

[14] P.A. Fouque, J. Stern. "*Fully Distributed Threshold RSA under Standard Assumptions*". Asiacrypt'01, 2001.

[15] S.S.M. Chow, T.H. Yuen, L.C.K. Hui, S.M. Yiu. "*Signcryption in Hierarchical Identity Based Cryptosystem*". 20th IFIP International Information Security Conference, 2004.

[16] SA Baset, H Schulzrinne. "*An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol*". Arxiv preprint CS.NI/0412017, 2004.

[17] T. Nadeem, S. Dashtinezhad, C. Y. Liao, L. Iftode. "*TrafficView: Traffic Data Dissemination using Car-to- Car Communication*". IEEE Int'l Conference on Mobile Data Management (MDM'04) and Mobile Computing and Communications Review 8 (3) :6-19, 2004.

[18] T. P. Pedersen. "*A Threshold Cryptosystem without a Trusted Party*". EUROCRYPT, 1991.

[19] V. Shoup. "*Practical Threshold Signatures*". Eurocrypt'00, LNCS 1807, pp. 207-220, 2000.

[20] X.Y. Zhang, J.C. Liu, Bo Li, T. S. Yum. "*CoolStreaming/ DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming"*. IEEE INFOCOM 2005, Miami, FL, 2005.

[21] Y. H. Guo, J. K. Zao, W. H. Peng, L. S. Huang, C. M. Lin, F. P. Kuo, "*Trickle: Resilient Real-Time Video Multicasting for Dynamic Peers with Limited or Asymmetric Network Connectivity*". IEEE International Symposium on Multimedia (ISM'06) , San Diego, U.S.A., December 11-13, 2006.

[22] Clifford Cocks, *An Identity Based Encryption Scheme Based on Quadratic Residues*, Proceedings of the 8th IMA International Conference on Cryptography and Coding, 2001

[23] R. Gennaro. S. Jarecki, H. Krawczyk, T. Rabin, "*Secure Distributed Key Generation for Discrete-Log Based Cryptosystems*". Proceeding Eurocrypt, 1999

[24] N. Saxena, G, Tsudik, Jeong Hyun Yi, "*Threshold cryptography in P2P and MANETs: The case of access control*". ScienceDirect Computer Networks 51 (2007) 3632-3649

[25] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, *Recommendation for Key Management*- Part 1 :General (Revised) , NIST Special Publication 800-57, May, 2006 *http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part1.pdf*

[26] Ya-Chu Yang, Chien-Ming Cheng, Pei-Yun Lin, and Shiao-Li Tsao, "*A Real-Time Road Traffic Information System based on a Peer-to-Peer Approach*," IEEE Symposium on Computers and Communications 2008

[27] A. J. Menezes, *Elliptic Curve Public Key Crytosystems*, Kluwer Academic

Publishers, 1993.

[28] J. H. Silverman, *The Arithmetic of Elliptic Curves, Graduate Texts in Mathematics*, 106, Springer-Verlag, 1986

[29] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press.

[30] V. Miller, "*Short Programs for Functions on Curves*", Unpublished Manuscript, 1986