行政院國家科學委員會專題研究計畫　成果報告

適用於分散式編譯器的自動匿名代理簽章法之設計

計畫類別：個別型計畫
計畫編號：NSC94-2213-E-032-022-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：淡江大學資訊工程學系

計畫主持人：黃心嘉

計畫參與人員：黃嘉濰　許德煜

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 95 年 10 月 3 日

行政院國家科學委員會補助專題研究計畫 ☑成果報告 □期中進度報告

適用於分散式編譯器的自動匿名代理簽章法之設計

計畫類別：☑ 個別型計畫　　□ 整合型計畫
計畫編號：NSC 94-2213-E-032-022
執行期間：　94 年 8 月 1 日至　95 年 7 月 31 日

計畫主持人：黃心嘉　淡江大學資工系　副教授
共同主持人：
計畫參與人員：黃嘉濰　　淡江大學資工系　研究生
　　許德煜　　淡江大學資工系　研究生

成果報告類型(依經費核定清單規定繳交)：☑精簡報告　□完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：淡江大學資工系

中　華　民　國　九十五 年八月三十一日

一、中英文摘要

中文摘要

為了避免檔案遭受未知電腦病毒的感染，許多藉由公正的編譯器製造者幫助的自動簽章方法已經被提出[9, 21, 46]。在這些所提出的方法中，對於驗證者而言，編譯器代理者們的公開金鑰必須被認證，而且驗證者必須將之記錄下來，這是一件麻煩的工作。如果驗證者只需要紀錄編譯器製造者的公開金鑰，那將會變得非常方便。為此本計畫提出一個匿名的代理自動簽章法。在新的方法中，因為編譯器的代理伺服器是匿名的，驗證者並不需儲存任何代理伺服器的公開金鑰，就可以驗證自動代理簽章，並且驗證者事前不只能驗證執行檔的來源，也可以偵測代理伺服器的不合法行為。我們所提出的方法可以採用於任何的離散對數型的簽章法，並且提供完善的裁判機制來偵測病毒感染的來源。

英文摘要

Many (proxy) automatic signature schemes are proposed [9, 21, 46] to guard against the (unknown) virus infection with the help of honest compiler makers. In these proposed schemes, the used compiler agents' public keys should be certificated and maintained by verifiers. If verifiers only keep the compiler makers' public key, it is more convenient. So this project proposes an anonymous proxy automatic signature scheme with compiler agents. In the new scheme, the compiler agents are anonymous and verifiers do not need to store compiler agents' public key to verify automatic proxy signatures. Moreover, verifiers can authenticate the source of received executable problems and detect compiler agents' deviation in advance. Our schemes are suitable for adopting any discrete logarithm based signature schemes. Our scheme has provides strong moderator's judgment to detect of virus infection sources.

關鍵字: Compilers, distributed system, computer virus, digital signature, proxy signature, automatic signature, anonymous.

二、前言與研究目的

在 Hwang 和 Li 的自動代理簽章法中[9]，消費者(或需求者)取得經過伺服器編譯的執行檔後，消費者(或需求者)必須使用伺服器與編譯器製造者的公開金鑰，方能驗證執行檔的自動代理簽章，以偵測執行檔是否被電腦病毒所感染修改。站在需求者(或消費者)的角度來看，需求者(或消費者)本身就必須知道使用哪一家編譯器製造者的編譯器，進行程式原始碼編譯成執行檔的動作；因此需求者(或消費者)知道編譯器製造者的身份是自然的，自然可以取得編譯器製造者的公開金鑰，以對編譯器製造者的編譯器所產生的執行檔及自動代理簽章，做驗證的動作。

但是伺服器是受編譯器製造者的代理委託的，兩者之間的編譯器代理關係，可能隨代理關係到期而改變，或是因為其它因素，編譯器製造者更動了伺服器。因此隨著時間演進，同一個編譯器製造者就會有許多不同時期的代理伺服器，可能是極端複雜，而且知道最清楚了，應該就是編譯器製造者本身了。反觀需求者與消費者就不甚清楚這些錯綜複雜的代理關係了。需求者與消費者最確定是可就是誰是編譯器製造者了。因此要求需求者和消費者必須也知道負責編譯原始碼的伺服器的公開金鑰，才能夠去做自動代理簽章的驗證，這

就會造成需求者與消費者的不方便與負擔。

所以如果可以把在伺服器端產生的自動代理簽章，改為匿名的自動代理簽章，那麼需求者就不需要知道是哪一台伺服器幫忙做編譯程式原始碼的工作，只要知道是被編譯器製造者合法授權的匿名伺服器即可；同時任何一位需求者(或消費者)對執行檔及所附加的代理簽章做驗證的時候，也不用知道匿名伺服器的公開金鑰，只需要知道代表編譯器製造者的公開金鑰就可以完成做驗證的動作，如此就不會受到代理授權伺服器的更動，而造成需求者(或消費者)驗證時的困擾。然而伺服器的匿名雖然可以達到需求者(或消費者)驗證的方便性，但是當使用者所拿到的執行檔及所附加的匿名代理簽章驗證發生錯誤的問題，代表執行檔受到電腦病毒感染時，仍須要有一套完整的判斷感染源流程，可以明確的指出感染源為伺服器、消費者，或是需求者撰寫的程式碼含有破壞性的病毒指令。因此在自動匿名代理簽章法中，當發生爭議時，必須能夠偵測出當初編譯可能感染病毒程式的匿名伺服器，所以必須加入完整的解除伺服器匿名的功能。

三、文獻探討

可匿名代理簽章的代理簽章法[4]的敘述

基於離散對數且利用委任書(warrant)方法的匿名代理簽章法描述如下:

此方法一共有四種成員：原始簽章者 U_O，代理簽章者 U_P，驗證者 V 和公正的第三方 TTP。首先說明的是公開系統參數和函數部分。p 和 q 是兩個大質數必且符合 p=2q+1，g 是一個在 Z_p^* 中且 order 為 q 的生成子，h()是一個賀序函數。每一個使用者 U_i 都有一個身份 ID_i，一把私密金鑰 $x_i \in Z_p^*$，合一把經過認證的公開金鑰 $y_i=g^{x_i} \bmod p$。在代理委任書裡面定義了原始簽章者 U_O，原始簽章者經過認證的公開金鑰 y_O，合法的授權期限和必要的代理細節描述。

演算法 $\mathbf{Auth_{APDW}(U_O, U_P, M_W, b, Proxy\text{-}Certificate)}$

輸入: 使用者 U_O 和 U_P 的身份以及委任書 M_W

輸出: 秘密值 b 和 Proxy-Certificate

(1) U_O 隨機在 Z_q^* 中選擇一個秘密值 b，接著計算 $y=y_P g^b \bmod p$

(2) U_O 使用離散對數型的千章法和私密金鑰 x_O 對 $h(M_W, y, h(bg^b \bmod p))$ 產生簽章 (r, s)。Proxy-Certificate=(M_W, y, $h(bg^b \bmod p)$, (r, s))。

(3) 原始簽章者透過安全通道傳送 Proxy-Certificate 和授權的秘密值 b 給代理簽章者 P。

演算法 $\mathbf{VerAuth_{APDW}(U_O, U_P, b', Proxy\text{-}Certificate)}$

輸入: U_O 和 U_P 的身份和秘密值 b'和 Proxy-Certificate

輸出: 由 VerCertAPDW(U_O, Proxy-Certificate)所回傳的布林值

(1) 代理簽章者計算代理公開金鑰 $y'=y_P g^b \bmod p$ 和 $h'=h(b'g^{b'} \bmod p)$。接著確認 $h(M_W, y, h(bg^b \bmod p))= h(M_W, y', h(b'g^{b'} \bmod p))$。

(2) 代理簽章者利用函式 $VerCert_{APDW}(U_O,$ Proxy-Certificate)確認 Proxy-Certificate 的正確性。

(3) 若函式 $VerCert_{APDW}(U_O,$ Proxy-Certificate)回傳 true，則代理簽章者計算代理秘密金鑰 $x=x_P+b$ 並且回傳 true，否則結束程序並回傳 false。

演算法 $\mathbf{VerCert_{APDW}(U_O,}$ Proxy-Certificate)

輸入: U_O 的身份和 Proxy-Certificate

輸出: 布林數。如果 Proxy-Certificate 被驗證通過則回傳 true 否則回傳 false

(1) 驗證者計算 $H=h(M_W, y, h(bg^b \bmod p))$

(2) 驗證者用公開金鑰 y_O，離散對數型的演算法及 H 去驗證簽章(r, s)的正確性，若驗證通過則回傳 true, 否則回傳 false。

演算法 $ID_{APDW}(TTP, U_O, y_p, b', Proxy\text{-}Certificate)$

輸入: 公正的第三方 TTP，U_O 的身份，經過認證的公開金鑰 y_P，由 U_O 提供的秘密值 b'和 Proxy-Certificate

輸出: 布林數。如果證明代理秘密金鑰 x 和代理公開金鑰的對應關係只會被知道 x_P 的人所產生則回傳 true，否則回傳 false。

(1) TTP 利用函式 $VerCert_{APDW}(U_O,$ Proxy-Certificate)驗證 Proxy-Certificate，若通過驗證則 TTP 確信代理公開金鑰被原始簽章者所認證。

(2) TTP 計算 $g^{b'} \bmod p$ 和 $y'= g^{b'}y_P \bmod p$ 並且檢查 $h(g^{b'} \bmod p)=h(g^b \bmod p)$和 $y'=y$ 兩個等式是否都成立，如果等式皆成立則回傳 true，否則回傳 false。

在新的匿名代理簽章法中共分為四個階段: 系統初始階段，代理授權階段，代理簽章的產生與驗證階段和代理簽章者的識別階段。

[系統初始階段]

系統的公開參數和函示如同上述，對於每一位使用者 U_i 都有一個身份 ID_i，一把私密金鑰 x_i 和對應的公開金鑰 $y_i=g^{x_i}$。M_W 則代表授權的委任書。

[代理授權階段]

原始簽章者首先執行 $Auth_{APDW}(U_O, U_P, M_W,$ b, Proxy-Certificate)產生秘密值 b 和 Proxy-Certificate 並送給代理簽章者。代理簽章者收到秘密值 b 和 Proxy-Certificate 後利用函式 $VerAuth_{APDW}(U_O, U_P,$ b', Proxy-Certificate)來做驗證。如果 $VerAuth_{APDW}(U_O, U_P,$ b', Proxy-Certificate)回傳 true 則代理簽章者計算代理私密金鑰 $x=x_P+b \bmod q$ 和對應的代理公開金鑰 $y=g^x \bmod p=y_Pg^b \bmod p$。

[代理簽章的產生與驗證階段]

首先代理簽章者利用代理秘密金鑰 x 及離散對數型的簽章法 $DLS_{(x, y)}(h(m||r||s))$對明文 m 產生代理簽章(R, S)。當驗證者收到(R, S)和 m 後利用 $VerAuth_{APDW}(U_O, U_P,$ b', Proxy-Certificate)來驗證 Proxy-Certificate 和代理公開金鑰 y 是否為原始簽章者所合法授權的，若正確則利用代理公開金鑰及 DLS 使用的簽章驗證演算法來驗證代理簽章(R, S)。

[代理簽章者的識別階段]

若匿名的代理簽章(R, S)有任何爭議，原始簽章者必須使用執行 $ID_{APDW}(TTP, U_O, y_p,$ b', Proxy-Certificate)來揭發匿名代理簽章者的身份並使公正的第三方信任只有知道秘密金鑰 x_P 的人才可產生代理秘密金鑰 x。

匿名代理簽章群的群體導向代理簽章法

基於離散對數且利用委任書(warrant)方法的匿名代理簽章法描述如下:

此方法一共有四種成員：原始簽章者 U_O，一代理簽章群 $G_P(U_1, U_2, \ldots, U_n)$，驗證者 V 和公正的第三方 TTP。首先說明的是公開系統參數和函數部分。p 和 q 是兩個大質數必且符合 $p=2q+1$，g 是一個在 Z_p^*中且 order 為 q 的生成子，h()是一個賀序函數。每一個使用者 U_i 都有一個身份 ID_i，一把私密金鑰 $x_i \in Z_p^*$，合一把經過認證的公開金鑰 $y_i=g^{x_i} \bmod p$。在代理委任書裡面定義了原始簽章者 U_O，原始簽章者經過認證的公開金鑰 y_O，合法的授權期限和必要的代理細節描述。

演算法 $Auth_{AMPDW}(U_O, G_P, M_W, \{b_1, b_2, \ldots, b_n\}, Proxy\text{-}Certificate)$

輸入: U_O 和 G_P 的身份和代理委任書(M_W)

輸出: 秘密值($b_1, b_2, \ldots, b_n$)

(1) 原始簽章者 U_O 為每一位代理簽章者 U_i 選擇一個秘密值 $b_i \in Z_q^*$ 並計算 $g^{b_i} \bmod p$ 和 $g^{\Sigma_{i=1}^{n} b_i} \bmod p$。接著 U_O 計算每一把獨立的代理公開金鑰 $y_{G_i}=y_i g^{b_i} \bmod p$ 和 $y_{G_P}=\Pi_{i=1}^{n} y_i g^{\Sigma_{i=1}^{n} b_i} \bmod p$。

(2) 原始簽章者 U_O 利用私密金鑰 x_O 和離散對數型的簽章法對 $h(M_W, y_{G_P}, \{(y_{G_1}, h(b_1 g^{b_1} \bmod p)), (y_{G_2}, h(b_2 g^{b_2} \bmod p)), \ldots, (y_{G_n}, h(b_n g^{b_n} \bmod p))\})$。Proxy-Certificate=$(M_W, y_{G_P}, \{(y_{G_1}, h(b_1 g^{b_1} \bmod p)), (y_{G_2}, h(b_2 g^{b_2} \bmod p)), \ldots, (y_{G_n}, h(b_n g^{b_n} \bmod p))\}, (r, s))$。

(3) 原始簽章者送 Proxy-Certificate 和授權的秘密值 b 透過安全的方法傳送給 G_P 中的每一位代理簽章者。

演算法 $VerAuth_{AMPDW}(U_O, G_P, \{b_1', b_2', \ldots, b_n'\}, Proxy\text{-}Certificate)$

輸入: U_O 和 G_P 的身份，秘密值**$\{b_1, b_2, \ldots, b_n\}$和 Proxy-Certificate**

輸出: 由函式 $VerCert_{AMPDW}(U_O, Proxy\text{-}Certificate)$所回傳的布林值

(1) 每一位在 G_P 中的代理簽章者 U_i 首先計算自己的公開金鑰 $y_{G_i}'=y_i g^{b_i} \bmod p$ 和 $h'=h(b_i' g^{b_i'} \bmod p)$，接著檢查 $h(M_W, y_{G_P}, \{(y_{G_1}, h(b_1 g^{b_1} \bmod p)), (y_{G_2}, h(b_2 g^{b_2} \bmod p)), \ldots, (y_{G_i}, h(b_i g^{b_i} \bmod p)), \ldots, (y_{G_n}, h(b_n g^{b_n} \bmod p))\}) = h(M_W, y_{G_P}, \{(y_{G_1}, h(b_1 g^{b_1} \bmod p)), (y_{G_2}, h(b_2 g^{b_2} \bmod p)), \ldots, (y_{G_i}', h(b_i' g^{b_i'} \bmod p)), \ldots, (y_{G_n}, h(b_n g^{b_n} \bmod p))\})$是否成立。

(2) G_P 中的代理簽章者 U_i 執行函式 $VerCert_{AMPDW}(U_O, Proxy\text{-}Certificate)$ 檢查 Proxy-Certificate。

(3) 若 $VerCert_{AMPDW}(U_O, Proxy\text{-}Certificate)$回傳 true 而且 $y_{G_P}=\Pi_{i=1}^{n} y_{G_i} \bmod p$，代理簽章者 U_i 計算代理秘密金鑰 $x_{G_i}=x_i+b_i \bmod q$。若所有的代理簽章者接回傳 true 給原始簽章者，則原始簽章者回傳 true，否則回傳 false。

演算法 $VerCert_{AMPDW}(U_O, Proxy\text{-}Certificate)$

輸入: U_O 的身份和 Proxy-Certificate

輸出: 如果 Proxy-Certificate 通故所有使用者 U_i 的驗證，則回傳布林數 true，否則回傳 false。

(1) 驗證者計算 $H= h(M_W, y_{G_P}, \{(y_{G_1}, h(b_1 g^{b_1} \bmod p)), (y_{G_2}, h(b_2 g^{b_2} \bmod p)), \ldots, (y_{G_n}, h(b_n g^{b_n} \bmod p))\})$

(2) 驗證者利用公開金鑰 y_O 和離散對數型的驗證式來驗證 Proxy-Certificate。如果簽章 (r, s) 正確且 $y_{G_P}=\Pi_{i=1}^{n} y_{G_i} \bmod$ 則 Proxy-Certificate 驗證通過且回傳布林值 true，否則回傳 false。

演算法 $ID_{AMPDW}(TTP, U_O, \{y_1, y_2, \ldots, y_n\}, G_P, M_W, \{b_1, b_2, \ldots, b_n\}, Proxy\text{-}Certificate)$

輸入: 公正的第三方(TTP)，U_O 的身份，每一位代理簽章者經過認證的公開金鑰$\{y_1, y_2, \ldots, y_n\}$，$U_O$ 所提供的秘密值$\{b_1, b_2, \ldots, b_n\}$和 Proxy-Certificate

輸出: 如果確認了秘密金鑰 x_i 和公開金鑰 y_i 只能被知道 x_{G_i} 的使用者知道則回傳 true，否則回傳 false。

(1) 公正的第三方首先用 $VerCert_{AMPDW}(U_O, Proxy\text{-}Certificate)$驗證 Proxy-Certificate，則

公正的第三方可以確認群代理公開金鑰 y_{G_P} 和個別的代理公開金鑰 y_{G_i} 是被原始簽章者所認證。

(2) 公正第三方計算每一個 $y_{G_i}'=y_i g^{b_i'} \bmod$ 和 $y_{G_P}'=\Pi_{i=1}{}^{n} y_i g^{\Sigma_{i=1}{}^{n} b_i'} \bmod$ 接著在檢查等式 $h(b_i g^{b_i} \bmod p)= h(b_i' g^{b_i'} \bmod p)$ 和 $y_{G_i}=y_{G_i}'$ 對於每一個 i=1, 2, …, n，如果兩等式皆成立則回傳 true，否則回傳 false。

在新的匿名代理簽章群的群體導向代理簽章法中共分為四個階段: 系統初始階段，代理授權階段，代理簽章的產生與驗證階段和代理簽章者的識別階段。

[系統初始階段]

如同上述匿名代理簽章的代理簽章法。

[代理授權階段]

原始簽章者 UO 先執行 $\text{Auth}_{\text{AMPDW}}(U_O, G_P, M_W, \{b_1, b_2, \ldots, b_n\}, \text{Proxy-Certificate})$ 來授權給一群代理簽章者。當代理簽章者收到 Proxy-Certificate 和秘密值 b_i 則利用 $\text{VerAuth}_{\text{AMPDW}}(U_O, G_P, \{b_1', b_2', \ldots, b_n'\}, \text{Proxy-Certificate})$ 來驗證 Proxy-Certificate 和秘密值 b_i 的正確性，若驗證通過則信任代理簽章的授權並產生個別的代理秘密金鑰 $x_{G_i}=x_i+b_i \bmod q$ 及對應的代理公開金鑰 $y_{G_i}= g^{x_{G_i}} \bmod p$。群代理公開金鑰為 $y_{G_P}=\Pi_{i=1}{}^{n} y_{G_i} \bmod p$。

[代理簽章的產生與驗證階段]

藉由在 G_P 中所有代理簽章者 U_i 的合作，每個代理簽章者提供他們的代理秘密金鑰 x_{G_i} 和一個離散對數型的簽章法(表示為 DLMS)，對於一個明文 m 產生一組簽章 $(R, S)=\text{DLMS}_{(\Sigma_{i=1}{}^{n} x_{Gi},\ \Sigma_{i=1}{}^{n} y_{Gi})}(h(m||r||s))$。當驗證者收到對明文 m 的簽章(R, S)，首先利用 $\text{VerCert}_{\text{AMPDW}}(U_O, \text{Proxy-Certificate})$ 驗證 Proxy-Certificate 的正確性，接著則利用對應的離散對數驗證式(表示為 DLMV)和群代理公開金鑰 y_{G_P} 驗證簽章的正確性 $\text{DLMV}_{\Pi_{i=1}{}^{n} y_{Gi}}((R, S), h(m||t||s))$。若量個驗證式皆通過則驗證者信任 Proxy-Certificate 及代理簽章(R, S)耶為合法且為原始簽章者 U_O 所授權。

[代理簽章者的識別階段]

若簽章(R, S)出現爭議時，原始簽章者必須執行 $\text{ID}_{\text{AMPDW}}(\text{TTP}, U_O, \{y_1, y_2, \ldots, y_n\}, G_P, M_W, \{b_1, b_2, \ldots, b_n\}, \text{Proxy-Certificate})$ 證明代理群 G_P 就是使用者 $\{U_1, U_2, \ldots, U_n\}$。

四、研究方法

本計畫預計採用 Hwang 和 Li [9]方法中的代理簽章法概念，從具有保護代理簽章者身分的代理簽章法下手，開始設計適用於分散式編譯器的自動匿名代理簽章法。目前具有保護代理簽章者身分的代理簽章法，計有 Chan [4]與 Shum 和 Wei [36] 兩個可以保護代理簽章隱私的代理簽章法，因此將從這兩個方法開始進行研究，研究如何保護代理簽章者隱私，進而設計分散式編譯器的自動匿名代理簽章法，尤其如何在能夠防止代理簽章者否認，又可以保護代理簽章者身份之間，取得安全考量上的平衡。

五、結果與討論

適用於分散式編譯器的自動匿名代理簽章法的敘述

本方法一共分為五個階段: 系統初始化階段、編譯器製造者-伺服器授權階段、伺服器-需求者執行階段、消費者驗證階段與仲裁者仲裁階段。

[***系統初始化階段***]

一個具公信力的系統(TTP)建立者負責下列系統參數與函數。

(1) p，q 為兩個公開的大質數，滿足 q=2q+1。

(2) g 為一個在 Z_p^* 中 order 為 q 的生成子。

(3) h 為一個安全的單向雜湊函數。

(4) 編譯器製造者、伺服器、需求者、消費者各選擇一個私密金鑰 $x_i \in Z_q^*$，並計算對應的公開金鑰 $y_i = g^{x_i} \bmod p$。

(5) 所有的公開金鑰都必須經過公信的公開金鑰認證中心認證。

(6) W 為編譯器製造者和伺服器的編譯器代理授權書，授權書內含雙方的公開金鑰和其他代理授權所需要使用到的資訊。

(7) C_R 代表由誠實的編譯器製造者所生產的編譯器。

(8) P 為需求者所撰寫的程式碼。

(9) E 代表由原始碼 P 所編譯成的執行檔。

[***編譯器製造者-伺服器授權階段***]

(1) 伺服器 u_S 首先利用任何一種離散對數的簽章法對明文摘要 $h(u_S||R_{SM})$產生簽章，然後傳送$(R_{SM}, u_S, (r_{SM}, s_{SM}))$給編譯器製造者 u_M，R_{SM} 代表伺服器向編譯器製造者提出代理編譯程式碼的需求。以下任何簽章皆可使用任意的離散對數型簽章法。

(2) 當編譯器製造者 u_M 收到$(R_{SM}, u_S, (r_{SM}, s_{SM}))$後執行以下步驟：

(2.1) 選擇一個隨機數 $b \in Z_q^*$。

(2.2) 計算 $Y = y_s \times g^b \bmod p$。

(2.3) 使用私密金鑰 xM 對明文摘要 $h(M_W, h(C_R), Y, h(bg^b \bmod p))$產生簽章$(r_{MS}, s_{MS})$ 。

(2.4) 把代理憑證$(M_W, h(C_R), Y, h(bg^b \bmod p), (r_{MS}, s_{MS}))$，編譯器 C_R，和秘密值 b 透過安全通道傳送給伺服器 u_S。

(3) 當伺服器 u_S 收到代理憑證$(M_W, h(C_R), Y, h(bg^b \bmod p), (r_{MS}, s_{MS}))$，編譯器 C_R，和秘密值 b 後執行以下步驟：

(3.1) 計算 $Y' = y_S \times g^{b'} \bmod p$。

(3.2) 利用等式 $h(bg^b \bmod p) = h(b'g^{b'} \bmod p)$來檢查秘密值 b。

(3.3) 利用 u_M 的公開金鑰 y_M 驗證簽章(r_{MS}, s_{MS})來檢查代理憑證$(M_W, h(C_R), Y, h(bg^b \bmod p), (r_{MS}, s_{MS}))$的正確性。

若代理憑證$(M_W, h(C_R), Y, h(bg^b \bmod p), (r_{MS}, s_{MS}))$正確，則計算代理私密金鑰為 $X = x_S + b \bmod q$，而對應的代理公開金鑰為 $Y = g^X \bmod p$。

[***伺服器-需求者執行階段***]

(1) 需求者 u_R 首先利用編譯器製造者 u_M 的公開金鑰 y_M 驗證簽章(r_{MS}, s_{MS})來檢查代理憑證$(M_W, h(C_R), Y, h(bg^b \bmod p), (r_{MS}, s_{MS}))$，若正確才執行以下步驟。

(2) 需求者 u_R 對 $h(P, (u_R||R_{RS}))$產生簽章(r_{RS}, s_{RS})，接著傳送$(P, (u_R||R_{RS}), (r_{RS}, s_{RS}))$給伺服器 u_S，R_{RS} 代表的是編譯原始碼的需求。

(3) 當伺服器 u_S 收到$(P, (u_R||R_{RS}), (r_{RS}, s_{RS}))$後驗證簽章$(r_{RS}, s_{RS})$，若驗證通過的話則伺服器通過需求者的要求且準備幫助需求者 u_R 編譯原始碼。

(4) 首先伺服器 u_S 要先檢查編譯器 C_R 的正確性，利用驗證代理憑證$(M_W, h(C_R), Y, h(bg^b \bmod p))$ 和 $h(C_R') = h(C_R)$來檢查。

(5) 若以上的驗證皆通過，伺服器 u_S 開始使用編譯器 C_R 編譯原始碼 P 產生執行檔 E。編譯器產生執行檔 E 後馬上使用代理私密金鑰 X 和任何一種離散對數型的簽章法對摘要 $h(u_R, E, h(C_R), h(P), (r_{MS}, s_{MS}))$產生代理自動簽章$(r_{SR}, s_{SR})$而不受任何的間斷。然後伺服器 u_S 傳送$(u_R, E, (r_{SR}, s_{SR}))$給需求者 u_R。

(6) 當需求者 u_R 收到$(u_R, E, (r_{SR}, s_{SR}))$後，首先檢查對摘要 $h(u_R, E, h(C_R), h(P), (r_{MS}, s_{MS}))$

的簽章(r_{SR}, s_{SR})，若簽章正確則需求者 u_R 可以準備傳送執行檔給消費者 u_C。

(7) 需求者 u_R 對摘要 h((r_{SR}, s_{SR}), E, h(C_R), h(P))產生簽章(r_{RC}, s_{RC})。

(8) 需求者 u_R 傳送(M_W, h(C_R), Y, h(bg^b mod p), (r_{MS}, s_{MS}))和((r_{SR}, s_{SR}), (r_{RC}, s_{RC}), u_R, E, h(P)) 給消費者 u_C。

[***消費者驗證階段***]

(1) 當消費者 u_C 第一次收到執行檔首先先利用需求者的公開金鑰 y_R 驗證對摘要 h((r_{SR}, s_{SR}), E, h(C_R), h(P))的簽章(r_{RC}, s_{RC})。

(2) 接著利用編譯器製造者 u_M 的公開金鑰 y_M 驗證代理憑證(M_W, h(C_R), Y, h(bg^b mod p), (r_{MS}, s_{MS}))。

(3) 利用代理公開金鑰 Y 驗證代理自動簽章(r_{SR}, s_{SR})。

當以上驗證通過後，則往後消費者要執行執行檔 E 前只需執行驗證代理自動簽章即可。

[***仲裁者仲裁階段***]

(1) 消費者 u_C 傳送下列資料給仲裁者：代理憑證(M_W, h(C_R), Y, h(bg^b mod p), (r_{MS}, s_{MS}))，代理自動簽章(r_{SR}, s_{SR})，u_R，E 和 h(P)。

(2) 仲裁者首先利用簽章(r_{MS}, s_{MS})和代理公開金鑰 Y 檢查代理憑證(M_W, h(C_R), Y, h(bg^b mod p), (r_{MS}, s_{MS}))，並且利用代理公開金鑰 Y 驗證代理自動簽章(r_{SR}, s_{SR})。若(r_{MS}, s_{MS})和(r_{SR}, s_{SR})兩個簽章皆正確，則仲裁者繼續下列步驟。

(3) 仲裁者向伺服器 u_S 取得原始碼 P'和簽章(r_{RS}, s_{RS})。仲裁者利用需求者的公開金鑰驗證簽章(r_{RS}, s_{RS})，若(r_{RS}, s_{RS})通過驗證則可以證明原始碼 P'是由需求者 u_R 所撰寫的，否則仲裁者就執行 ID_{APDW}(TTP, u_M, y_S, b, proxy certificate)來撤銷伺服器 u_S 的匿名能力並找出真正的伺服器 ID。

(4) 仲裁者從編譯器製造者 u_M 取得原始授權的編譯器 C_R'。

(5) 仲裁者使用編譯器 C_R'和原始碼 P'產生執行檔 E。

(6) 仲裁者驗證下列不等式 h(P)≠ h(P')，h(C_R)≠ h(C_R')，和 h(E)≠ h(E')，其中若有任何一個不等式成立，則仲裁者執行 ID_{APDW}(TTP, u_M, y_S, b, proxy certificate)來撤銷伺服器 u_S 的匿名能力並找出伺服器的 ID。

(7) 若 h(P')= h(P)，則仲裁者一行一行的檢查原始碼來確認需求者撰寫的原始碼中是否藏有惡意的病毒指令。

六、參考文獻

[1] Agnew, G. B., Mullin, R. C. and Vanstone, "Improved Digital Signature Scheme Based on Discrete Exponentiation," *Electronics Letters*, vol. 26, no. 14, pp. 1024-1025, 1990.

[2] Boyd, C., "Digital *Signature*," *Proceedings of Conference on Coding and Cryptography*, Cirencester, Dec. 1986, pp. 15-17.

[3] Boyd, C., "Comment: New Digital Signature Scheme Based on Discrete Logarithm," *Electronics Letters*, vol. 30, no. 6, pp. 480-480, 1994.

[4] Chan, C.–C., *Anonymous (Multi-)Proxy Signature Scheme with Undeniable Agent*, Master. Thesis, Tamkang University, Taiwan, R.O.C., Jun. 2004.

[5] ElGamal, T., "A public key cryptosystem and a signature scheme based on discrete logarithm," *IEEE Transactions on Information Theory*, Vol. 31, No. 4, pp. 469-1985, 1985.

[6] Harn, L., "New Digital Signature Scheme Based on Discrete Logarithm," *Electronics Letters*, Vol. 30, No. 5, pp. 396-398, 1994.

[7] Harn, L., "*Group*-oriented (t, n) threshold digital signature scheme and digital multisignature," *IEE Proceedings: Computers and Digital Techniques*, Vol. 141, No. 5, pp. 307-313, 1994.

[8] Harn, L. and Xu, Y., "Design of Generalised ElGamal Type Digital Signature Schemes Based on Discrete Logarithm," *Electronics Letters*, Vol. 30, No. 24, pp. 2025- 2026, 1994.

[9] Hwang, S.-J. and Li, E.-T., "A Proxy Automatic Signature Scheme Using a Compiler in Distributed Systems," *2004 Information Security Conference*, Taipei, Taiwan, R.O.C., Jun. 10-11, 2004, pp. 345- 352.

[10] Horster, P., Michels, M., and Petersen, H., "Authenticated Encryption Schemes with Low Communication Costs," *Electronics Letters*, Vol. 30, No. 15, pp. 1212-1212, 1994.

[11] Horster, P., Petersen, H., and Michels, M., "Meta-ElGamal Signature Scheme," *Proc. 2nd ACM Conf. on Computer and Comm. Security*, , pp. 96-107, Fairfax , May 1994.

[12] Horster, P., Petersen, H., and Michels, M., "Meta Message Recovery and Meta Blind Signature Schemes Based on the Discrete Logarithm Problem and Their Applications," *Preproceedings of Asiacrypt' 94*, Australia, Nov. 1994, pp. 185- 196.

[13] Hsu, Chien-Lung, Wu, Tzong-Sun, and Wu, Tzong-Chen, "New Nonrepudiable Threshold Proxy Signature Scheme with Known Signers," *Journal of Systems and Software*, Vol. 58, pp. 119-124, 2001.

[14] Hwang, Min-Shiang, Lin, Iuon-Chang, and LU, Eric Jui-Lin, "A Secure Nonrepudiable Threshold Proxy Scheme with Known Signers," *INFORMATICA*, Vol. 11, No. 2, pp. 137-144, 2000.

[15] Hwang, Min-Shiang, Lee, Cheng-Chi, and Hwang, Shin-Jai, "Cryptanalysis of the Hwang-Shi Prosy Signature Scheme," *Fundamental Informaticae*, Vol. 53, No. 3, pp. 131-134, 2002.

[16] Hwang, S. J., Chang, C. C. and Yang, W. P., "An Encryption/Signature Scheme with Low Message Expansion," *Journal of the Chinese Institute of Engineers*, Vol. 18, No. 4, pp. 591-595, 1995.

[17] Hwang, S.-J. and Chen, C.-C., "New multi-proxy multi signature schemes," *Applied Mathematics and Computation* Volume: 147, Issue: 1, pp. 57-67, 2004.

[18] Hwang, S. J. and Shi, Chi-Hwai, "Specifiable Proxy Signature Schemes," *1999 National Computer Symposium*, Taipei, Taiwan, R.O.C., Dec. 20-21, 1999, pp. C190-C197.

[19] Hwang, S. J. and Shi, Chi-Hwai, "A Simple Multi-Proxy Signature Scheme," *Communications of the CCISA*, Vol. 8, No. 1, pp. 88-92, Dec. 2001.

[20] Kim, S., Park, S., and Won, D., "Proxy Signatures, Revisited", *Proc. Int'l Conf. Information Security and Comm. Security (ICICS '97), Lecture Notes in Computer Science, 1334*, Berlin: Springer, 1997, pp. 223-232.

[21] Lin, W.–D. and Jan, J.–K., "An automatic signature scheme using a compiler in distributed systems," *IEICE Transactions on Communications*, Vol. E83-B, No. 5, pp. 935-941, 2000.

[22] Lee, W. B. and Chang, Chin-Chen, "Authenticated Encryption Scheme without Using a One Way Function," Electronics Letters, Vol. 31, No. 19, pp. 1656-1657, 1995.

[23] Li, Li-Hua, Tzeng, Shiang-Feng, and Hwang, Min-Shiang, "Generalization of proxy

signature-based on discrete logarithms," *Computers & Security*, Vol. 22, No. 3, pp. 245-255, 2003.

[24] Lyuu, Y.-H. and Wu, M.-L., "Cryptanalysis of and Improvement of the Hwang-Chen Multi-Proxy Multi-Signature Schemes," to appear in Applied Mathematics and Computation, 2004.

[25] MAMBO, M., USUDA, K., and OKAMOTO, E., "Proxy signatures: delegation of the power to sign message," IEICE Transactions Fundamentals, E79-A, No. 9, pp. 1338-1354, 1996.

[26] MAMBO, M., USUDA, K., and OKAMOTO, E., "Proxy Signatures for Delegation Signing Operation," Proc. 3rd ACM Conference on Computer and Communications Security, 1996, pp. 48-57.

[27] Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A., Handbook of Applied Cryptography, New York: CRC Press, 1997.

[28] National Institute of Standards and Technology (NIST): "A Proposal Federal Information Processing Standard for Digital Signature Standard (DSS)," Federal Register, Vol. 56, No. 169, Aug. 1991, pp. 42980-42982.

[29] Nyberg, K., "Comment: New Digital Signature Scheme Based on Discrete Logarithm," Electronics Letters, Vol. 30, No. 6, pp. 481-481, 1994.

[30] Nyberg, K. and Rueppel, R. A., "Message recovery for signature scheme based on the discrete logarithm problem," Design, Codes and Cryptography, Vol. 7, No. 1-2, pp.61-81, 1996.

[31] Okamoto, E., "Integrated security system and its application to anti-viral methods," Proc. 6th Virus and Security Conf., 1993.

[32] Piveteau, J. M., "New Signature Scheme with Message Recovery," Electronics Letters, Vol. 29, No. 25, pp. 2185-2185, 1993.

[33] Rabin, M.O., "Digitalized Signatures and Public Key Functions as Intractable as Factorization," MIT/LCS/TR-212, 1979.

[34] Rivest, R. L., Shamir, A. and Adleman, L., "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of ACM, Vol. 21, No. 2, pp. 120-126, 1978.

[35] Schnorr, C. P., "Efficient identification and signatures for smart cards," Advances in Cryptology-CRYPTO,89, LNCS 435, New York: Springer-Verlag, 1990, pp.239-252.

[36] Shum, K. and Wei, V. K., "A strong proxy signature scheme with proxy signer privacy protection," Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises 2002 (WETICE 02), Pittsburgh, PA, 2002, pp. 55–56.

[37] Stinson, Douglas R., Cryptography: Theory and Practice, New York: CRC Press, 1995.

[38] Sun, H. –M., "Design of Time-stamped Proxy Signatures with Traceable Receivers," IEE Proceedings Computers and Digital Techniques, Vol.147, and Issue: 6, 2000, pp. 462-466.

[39] Sun, Hung-Min, "Efficient nonrepudiable threshold proxy signature scheme with known signers," Computer Communications, Vol. 22, No. 8, p 717-722, 1999.

[40] Sun, Hung-Min, "On Proxy (Multi-) Signature Schemes", 2000 International Computer

Symposium, Chiayi, Taiwan, R.O.C., Dec. 6-8, 2000, pp. 65-72.

[41] Sun, Hung-Min and Chen, Biing-Jang, “Time-Stamp Proxy Signatures with Traceable Receivers," Proceedings of the Ninth National Conference on Information Security, Taiwan, 1999, pp. 247-253.

[42] Sun, Hung-Min and Hsieh, Bin-Tsan, “Remark on Two Nonrepudiable Proxy Signature Schemes,” Proceedings of the Ninth National Conference on Information Security, Taiwan, 1999, pp. 241-246.

[43] Sun, Hung-Min, Lee, N-Y and Hwang T., “Threshold Proxy Signatures,” IEE Proc.-Computers and Digital Techniques, Vol. 146, No. 5, pp. 259-263, 1999.

[44] Tseng, Y.–M., “Cryptanalysis and restriction of an automatic signature scheme in distributed systems,” *IEICE Transactions on Communications*, Vol. E86-B No. 5, pp. 1679-1681, 2003.

[45] Tzeng, Shiang-Feng, Hwang, Min-Shiang, and Yang, Cheng-Ying, “An improvement of nonrepudiable threshold proxy signature scheme with known signers,” *Computers & Security*, Vol. 23, Issue 2, pp. 174-178, 2004.

[46] Usuda, K., Mambo, M., Uyematsu, T., and Okamoto, E., “Proposal of an automatic signature scheme using a compiler,” *IEICE Transactions Fundamentals*, Vol. E79-A, No. 1, pp .94-101, 1996.

[47] Yen, S.-M., Hung, C.-P., and Lee, Y.-Y., “Remarks on Some Proxy Signature Schemes,” 2000 International Computer Symposium, Chiayi, Taiwan, R.O.C., Dec. 6-8, 2000, pp. 54-59.

[48] Yen, S. M. and Laih, C. S., “New Digital Signature Scheme Based on Discrete Logarithm,” Electronics Letters, Vol. 29, No. 12, pp. 1120-1121, 1993.

[49] Yi, L. Bai, G., and Xiao, G., “Proxy multi-signature scheme: A new type of proxy signature scheme,” Electronics Letters, Vol. 36, No. 6, pp.527-528, 2000.

[50] Zhang, K, “Threshold proxy signature schemes,” 1997 Information Security Workshop, Japan, Sep. 1997, pp. 191-197.

[51] 張真誠, 電腦密碼學與資訊安全, 台北: 松崗, 1989.

[52] 賴溪松, 韓亮, 張真誠, *近代密碼學及其應用*, 台北: 松崗, 1995.

七、計畫成果自評

本計畫的結果分別提出了一個適用於分散式編譯器的自動匿名代理簽章法，讓原始簽章者能授權給一群匿名的代理伺服器。在方法中，被授權代理伺服器是匿名的且代理伺服器亦無法知道其他代理伺服器的身份，另外和 Hwang 和 Li [9]的方法比較，在新的方法中，驗證者只需要保存編譯器製造者的公開金鑰來驗證自動簽章，並且代理編譯伺服器行為的愉悅可以被事先偵測。新的方法也擁有更完善的裁判機制，使的當有爭議發生時可以正確的發現病毒的感染援。因此達成本計畫的目標：適用於分散式編譯器的自動匿名代理簽章法之設計。