國立交通大學

資訊學院　資訊學程

碩 士 論 文

利用階層化基於身份公開金鑰密碼系統保護隨意自組網安全資料交換

Secure Ad-Hoc Transactions Protected By Hierarchical Identity-Based Public Key Cryptography

指導教授：邵家健　教授

研 究 生：李　仲　平

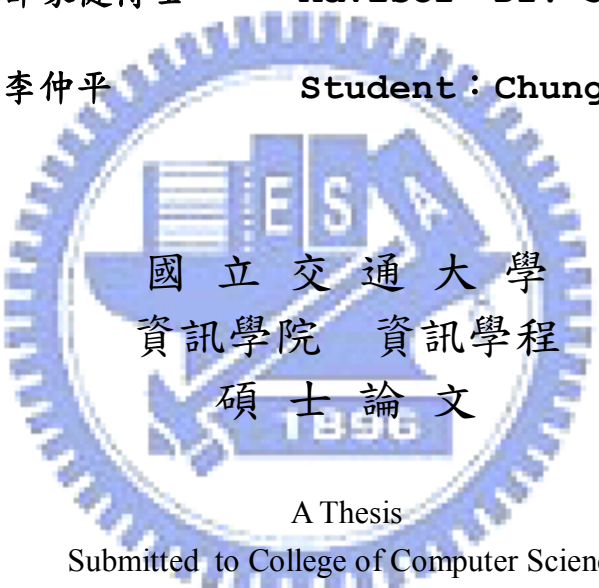
中 華 民 國 九 十 八 年 一 月

利用階層化基於身份公開金鑰密碼系統保護隨意自組網安全資料交換

Secure Ad-Hoc Transactions Protected By Hierarchical Identity-Based Public Key Cryptography

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

研 究 生：李仲平　　Student：Chung-Ping Lee



國 立 交 通 大 學

資訊學院　資訊學程

碩 士 論 文

A Thesis

Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

Jan 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年一月

利用階層化基於身份公開金鑰密碼系統保護隨意自組網安全資料交換

學生：李仲平　　指導教授：邵家健博士

國立交通大學

資訊學院　資訊學程碩士班

摘　　要

基於身份公開金鑰基礎建設(IDPKI)系統技術簡化了對存放公鑰憑證之目錄服務主機的依賴所造成之不便與龐大的憑證管理成本，而階層化基於身份公開金鑰基礎建設(HIDPKI)系統技術則可進一步適用在複雜可彈性成長網路中分攤單一私鑰產生中心(PKG)之負載瓶頸，也符合人類社會之運作架構。但若想要應用 HIDPKI 系統技術，則昔日之安全服務與協定就需做些修改，本論文即利用 HIDPKI 系統技術中階層化基於身份密碼系統(HIBC)技術進行雙向身分鑑別認證(Mutual Authentication)、雙方密鑰協商(Key Agreement)、安全資料交換(Secure Data Transaction)方案之研究；並將此技術使用在 SSL／TLS 協定與 IEEE 802.1X、802.11i 協定上，並嘗試對協定弱點提出強化之想法。

Secure Ad-Hoc Transactions Protected By Hierarchical Identity-Based Public Key Cryptography

student : Chung-Ping Lee Advisors : Dr. John Kar-kin Zao

Degree Program of Computer Science

National Chiao Tung University

ABSTRACT

ID-based PKI Technology can reduce the huge cost of certificate management and reduce the dependence of directory server for public key & certificate search. Hierarchical ID-based PKI Technology can not only share the heavy load bottleneck of single PKG in a scalable network environment but also compatible with hierarchical structure of human organization. When we want use HIDPKI technology, the previous security services & protocols must be modified. In this thesis we propose the HIBC technology in HIDPKI System at the Mutual Authentication, Key Agreement, Secure Data Transaction Mechanisms; application for SSL/TLS, IEEE 802.1X, 802.11i Protocols, and try to improve the weakness of protocol.

誌　　謝

首先要特別感謝指導老師邵家健教授於母親生病住院時的關心幫忙，也感謝邵老師於學生研究低潮時的加油打氣與指導，讓學生在研究過程有些不同嘗試與學習機會。也感謝陳榮傑教授許多年前的培訓課程中精采講授與解惑指導。同時要感謝陳榮傑教授、邵家健教授、胡鈞祥博士三位委員能百忙撥戎蒞臨口試指導，並提出寶貴的修改建議。在此也要感謝兩位老師共同指導而實力甚強的輔國學長能幾次抽空幫忙解惑，讓在下能夠從一個較陌生的研究領域能較快進入狀況。

其實幾年的學習過程中要感謝的人還很多，一些也曾默默幫助的老師與貴人，請恕無法一一詳列，但也將長存感恩獻上祝禱。

遺憾地是未能與母親分享這份遲來的喜悅，雖然畢業延後了許多年，但慶幸能陪伴您走過最後幾年。孩子必將竭盡努力，盼讓生命留下點有用足跡，讓您天靈稍感欣慰。

目　　錄

圖 目 錄

符 號 說 明

IBC 基於身份密碼系統(Identity based Cryptosystem)

HIBC 階層化基於身份密碼系統(Hierarchical Identity based Cryptosystem)

PKI 公開金鑰基礎建設(Public Key Infrastructure)

CA 憑證授權中心(Certificate Authority)

PKG 私鑰產生中心(Private Key Generator)

P 生成元素(Generator)

$P_{pub/i}$ 第 i 層 PKG 之公鑰(Public Key of PKG at level i)

$S_{/i}$ 第 i 層 PKG 之私鑰(Private Key of PKG at level i)

s 安全點(Secret Point)

$\hat{e}()$ 雙線性對運算(Bilinear Map Operation)

$\hat{t}()$ Tate Pairing

G_1 加法循環群(Additive Cyclic Group)

G_2 乘法循環群(Multiplicative Cyclic Group)

Z_q^* 乘法群(Multiplicative Group)

$P_{pubU/i}$ 第 i 層使用者 U 之公鑰(Public Key of user U at level i)

$S_{U/i}$ 第 i 層使用者 U 之私鑰(Private Key of user U at level i)

N_X 代表通訊主體 X 身分之隨機亂數(Nonce of Entity X)

H 雜湊函數(Hash Function)

MAC 訊息認證碼(Message Authentication Code)

EKP 會談密鑰協商所交換之臨時可公開金鑰參數(Ephemeral Key Parameter)

K_{XY} 通訊主體 X 與 Y 共享之會談密鑰(Shared Session Key between Entity X & Y)

HIBE Hierarchical ID-based Encryption

$HIBE^{-1}$ Hierarchical ID-based Decryption

HIBS Hierarchical ID-based Signature

$HIBS^{-1}$ Hierarchical ID-based Verification

HIBES Hierarchical ID-based Signcryption

$HIBES^{-1}$ Hierarchical ID-based DeSigncryption

E RijnDael Encryption

D RijnDael Decryption

1 Introduction

1.1 Problem

在開放之網路環境中通訊節點間要安全通訊，至少要有下列之安全服務：通訊主體身分鑑別認證(Entity Authentication)、資料安全保密(Data Confidentiality)、資料完整真確(Data Integrity)、存取控制(Access Control)、有時還會要求有不可否認(Non-Repudiation)等服務。

而傳統網路環境中之安全機制多係利用公開金鑰基礎建設(Public Key Infrastructure; ＰＫＩ)的憑證授權中心(Certificate Authority; CA)來扮演可信任第三者(Trust Third Party; TTP)進行產生與配發及管理維護通訊主體之公鑰與憑證，藉由 CA 簽章之公鑰憑證達到通訊主體身分鑑別認證。但因金鑰憑證之產生與管理成本甚高，且在開放環境中通訊前不易找到對方之公鑰，故傳統公開金鑰基礎建設機制至今一直無法全面實行。故開始有所謂基於身分公開金鑰基礎建設(Identity-based Public Key Infrastructure; IDPKI)的系統技術解決此問題。

雖然 IDPKI 技術的產生相對傳統 PKI 架構，已簡化存放公鑰憑證之目錄服務主機及憑證的依賴不便及產生與管理成本，但若遇到較龐大或複雜網路時，原本單一之 PKG 將會成為負載瓶頸，

因而產生了階層化基於身份公開金鑰基礎建設(Hierarchical Identity-based Public Key Infrastructure; HIDPKI)系統技術。HIDPKI 系統之階層化 PKG 架構不但適合在複雜可彈性成長網路中分攤單一 PKG 之負載瓶頸，也符合人類社會之運作架構。但若想要應用 HIDPKI 系統技術，因計算量很大且加密與簽章之位元數較少，故昔日之安全服務與安全協定就需做些修改。

1.2 Approach

本論文即利用 HIDPKI 系統技術中階層化基於身份密碼系統(Hierarchical Identity based Cryptosystem; HIBC)技術進行雙向身分鑑別認證(Mutual Authentication)、雙方密鑰協商(Key Agreement)、安全資料交換(Secure Data Transaction)方案之研究;並用 HIBC 技術在 SSL／TLS 協定與 IEEE 802.1X、802.11i 協定上，並嘗試針對協定弱點提出部份強化之想法。

1.3 Outline

本論文共分五章，除了本章前言外，第 2 章先介紹本論文之主要背景技術(IBC/HIBC)觀念，第 3 章討論雙向身份鑑別認證與雙方會談密鑰協商以及安全資料交換之研究方案，第 4 章討論利用 HIBC 技術在 SSL／TLS 及 IEEE 802.1X、802.11i 協定之應用研究與弱點改善強化，最後第 5 章做一結論。

2 Technology

本章將先介紹無階層化之基於身份密碼系統(Identity-based Cryptosystem; IBC)的基本原理與技術，接者再介紹階層化之基於身份密碼系統(Hierarchical Identity-based Cryptosystem; HIBC)技術。主要係節錄或翻譯幾個代表性方案[5,6,7,8,9]。

2.1 Identity-based Cryptosystem (IBC)

基於身份密碼系統技術的主要精神是用外顯易得能代表身分之任意字串(Identity String)來當作通訊主體之公鑰，產生此想法之原始動機，是想簡化對存放公鑰憑證之目錄服務主機及憑證的依賴造成不便與管理成本。其中基於身份加密(Identity-based Encryption; IBE)技術的觀念是 Shamir 在 1984 年第一次提出的，即利用可代表身分之外顯易得資訊來當作公鑰，同時也可用此產生臨時使用之短命公鑰(Ephemeral Public Key)。例如，若 Alice 要將訊息安全地送到 Bob 的信箱 bob@yahoo.com，則 Alice 只需直接用"bob@yahoo.com"或再經公開週知運算後當做公鑰來加密訊息。最近還研究出具前向安全性功能之加密機制(Forward-Secure Encryption Scheme)。很快地幾個基於身份簽章技術(Identity-based Signature; IBS)也被研究出來。[2,5]

但因當時這些技術之計算量還太複雜，且安全性也尚未經過形式化分析證明(Formal Analysis & Proof)，還不能實際應用。直到 2001 年，Boneh & Franklin 利用雙線性對技術(Bilinear Pairing)提出第一個安全且實用之基於身份加密技術後，一些利用雙線性對技術的基於身份加密、基於身份簽章及基於身份密鑰協商(Identity-based Key Agreement)技術之研究相繼產生。

故本節將先介紹基於身份密碼系統中的幾種機制技術－基於身份加密(ID-based Encryption; IBE)、與基於身份簽章(ID-based Signature; IBS)、以及基於身份密鑰協商(ID-based Key Agreement; IBKA)技術，但限於篇幅，每種機制只介紹一種方案。

由於三種機制的前兩階段都均是 Setup 與 Extract，故先抽出在此說明。基於 IBC 技術之安全系統中至少會有一個可離線之可信任第三者私鑰產生中心(Private Key Generator; PKG)，類似傳統 PKI 系統之 CA，但已簡化甚多。同樣會產生 PKG 系統自己的公鑰與私鑰對，也會接受使用者申請，為其產生對應之私鑰。而 Setup 階段就是 PKG 先產生系統自己之私密參數與私鑰及公鑰與系統參數。至於 Extract 階段就是使用者將代表其身分之

字串 (Identity)經過運算產生公鑰後，提交向 PKG 申請使用者之私鑰，PKG 驗證身份後再用系統私密參數產生對應使用者身分字串之私鑰。由圖 2.1-1 可大致了解第二階段運作機制。

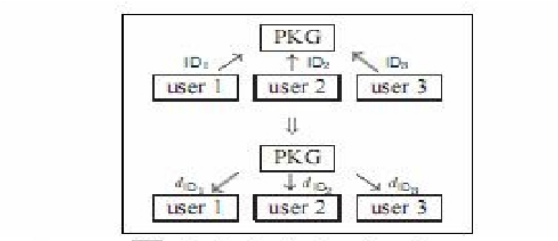


圖 2.1-1 Private Key Request & Extraction at PKG

2.1.1 Identity-based Encryption (IBE)

本小節將介紹 Boneh & Franklin 之 IBE 方案[2,5]，共分 Setup, Extract, Encryption, Decryption 4 個階段，下面逐一介紹這 4 個階段。

Setup 階段：

1) PKG 產生基於質數階數 q 之加法循環群 G_1 及乘法循環群 G_2，

也產生其雙線性對 $\hat{e}: G_1 \times G_1 \rightarrow G_2$。

2) 隨機挑選生成元素 $P \in G_1$，並挑選隨機數 $s \in Z_q^*$，

算出系統公鑰 $P_{pub}=sP$，s 為 PKG 系統主要私鑰(MasterKey)。

3) 挑選四個雜湊函數，

$H_1:\{0,1\}^* \rightarrow G_1^*$, $H_2:G_2 \rightarrow \{0,1\}^n$,

$H_3:\{0,1\}^n \times \{0,1\}^n$, $H_4:\{0,1\}^n \rightarrow \{0,1\}^n$.

明文空間 $M=\{0,1\}^n$, 密文空間 $C=G_1^* \times \{0,1\}^n \times \{0,1\}^n$,

公開系統參數 $<G_1, G_2, \hat{e}, q, n, P, P_{pub}, H_1, H_2, H_3, H_4>$

Extract 階段：

身分字串 $ID \in \{0,1\}^n$，已知使用者U身份字串 ID，

1) 使用者U算出U之公鑰 $P_{ID}=H_1(ID) \in G_1^*$後向 PKG 申請，

2) PKG 算出使用者私鑰 $S_{ID}=sP_{ID}$，再將 S_{ID} 安全地交給U。

Encryption 階段：

若明文為M，並已知 P_{ID}， 則

1) 挑選隨機數 σ，計算 $r=H_3(\sigma, M)$， $g=\hat{e}(P_{pub}, P_{ID})$，

2) 組成密文 $C=<rP, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma)>$ 後送給收方。

Decryption 階段：

若將密文 $C=<rP, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma)>$視為$<U, V, W>$則

1) 收方計算 $g'=\hat{e}(U, S_{ID})$， $\sigma=V \oplus H_2(g')$， $M=W \oplus H_4(\sigma)$，

2) 計算 $r=H_3(\sigma, M)$，若 $U \neq rP$ 則丟棄C，否則解密得M。

2.1.2 Identity-based Signature (IBS)

本小節將介紹 Hess 之 IBS 方案[4,6]，共分 Setup, Extract, Sign, Verify 4個階段，下面逐一介紹這4個階段。

Setup 階段：

1) PKG 產生基於質數階數 q 之加法循環群 G_1 及乘法循環群 G_2，

也產生其雙線性對 $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$。

2)隨機挑選生成元素 $P \in G_1$，並挑選隨機數 $s \in Z_q^*$，

算出系統公鑰 $P_{pub}=sP$，s 為 PKG 系統主要私鑰(MasterKey)。

3)挑選二個雜湊函數，

$H_1:\{0,1\}^* \rightarrow G_1^*$,　　$h:\{0,1\}^n \times G_2 \rightarrow Z_q^*$,

公開系統參數 $<G_1, G_2, \dot{e}, P, P_{pub}, H_1, h>$

Extract 階段：

身分字串 $ID \in \{0,1\}^n$，已知使用者U身份字串 ID，

1)使用者U算出U之公鑰 $P_{ID}=H_1(ID) \in G_1^*$後向 PKG 申請，

2)PKG 算出使用者私鑰 $S_{ID}=sP_{ID}$，再將 S_{ID}安全地交給U。

Sign 階段：

若U欲用私鑰 S_{ID}為明文M簽章時，則

1)簽章者先挑選隨機數 k 及任意數 p_1，

2)計算 $r=\dot{e}(p_1, P)$，　$v=h(M, r)$，　$U=vS_{ID}+kp_1$，

3)組成簽章 $\sigma=<U, v>$ 後送給收方。

Verify 階段：

若收方要驗證簽章 $\sigma=<U, v>$，則

計算 $r=\dot{e}(U, P) \cdot \dot{e}(P_{ID}, -P_{pub})^v$，若 $v=h(M, r)$則接受此簽章。

2.1.3 Identity-based Authenticated Key Agreement (IBAKA)

本小節將介紹 McCullagh 及 Barreto 之 IBAKA 方案[4,7]，而上述二人之研究有三個方案，分別解決三個問題，本章之目的係觀念介紹，基於篇幅考量只介紹第二方案。本方案共分 Setup, Extract, Key Agreement 3個階段，下面將介紹這3個階段。

Setup 階段：

1) PKG 產生基於質數階 q 之加法循環群 G_0, G_1 及乘法循環群 G_2，也產生其雙線性對 $\hat{t}: G_0 \times G_1 \rightarrow G_2$。

2) 隨機挑選生成元素 $P \in G_0$，$Q \in G_1$，並挑選隨機數 $s \in Z_q^*$，算出系統公鑰 $P_{pub}=sP$，s 為 PKG 系統主要私鑰(MasterKey)。

3) 挑選一個雜湊函數 $H_1: \{0,1\}^* \rightarrow Z_q^*$，公開系統參數 $<G_1, G_2, \hat{t}, P, P_{pub}, H_1>$

Extract 階段：

已知使用者A之身份字串為 ID_A，使用者B之身份字串為 ID_B。

PKG 算出A之公鑰 $P_A=(a+s)P$，此處 $a= H_1(ID_A)$

算出A之私鑰 $S_A=(a+s)^{-1}Q$，將 S_A 安全地交給A。

PKG 算出B之公鑰 $P_B=(b+s)P$，此處 $b= H_1(ID_B)$

算出B之私鑰 $S_B=(b+s)^{-1}Q$，將 S_B 安全地交給B。

Key Agreement 階段：

若欲用協商雙方共享之會談密鑰時，則

使用者A先挑選隨機數 χ_a，計算 $T_A=\chi_a P_B$，並送 T_A 給B。

使用者B先挑選隨機數 χ_b，計算 $T_B=\chi_b P_A$，並送 T_B 給A。

使用者A計算 $K_{AB}=\hat{t}(T_B, S_A)^{\chi_a}$，

使用者B計算 $K_{BA}=\hat{t}(T_A, S_B)^{\chi_b}$，

最後雙方計算出共享之會談密鑰 $K_{AB}=K_{BA}=\hat{t}(P, P)^{\chi_a\chi_b}=K$。

2.2 Hierarchical Identity-based Cryptosystem (HIBC)

IBC 技術的產生雖對傳統 PKI 架構已簡化方便許多，但若遇到較龐大或複雜網路時，PKG 就變成負載瓶頸失敗單點危機。不但是計算量龐大複雜，同時也需驗證使用者身份，並產生使用者私鑰後，建立安全通道送回給使用者，因而產生 HIBC 技術。

HIBC 與 IBC 之差別是，IBC 系統只有一個 PKG，但 HIBC 因可將驗證身份與產生私鑰與發送工作分散授權給較低層之 PKG，能階層化多個 PKG 分工，在不同階層之 ID 與私鑰會層層相關。若 Root PKG 階層代號為0，下一層為1，餘此類推。而關於 PKG 之 ID 關係請參考圖 2.2-1[2]，若定義第 i 層之 PKG 之 ID(i)被表示為<$ID_1, ID_2, .. ID_i$>，係由上層 ID 加上本層 ID 不斷串接而成。至於 PKG 之私鑰關係則請參考圖 2.2-2[1]，例

如第 i 層之私鑰 $S_i=S_{i-1}+s_{i-1}P_i$。其實 HIBC 中之 Root PKG 在 Setup 階段是與 IBC 相同，但下層 PKG 之 Setup 階段產生金鑰時，會如前述受上層 PKG 影響。[1,2,8]

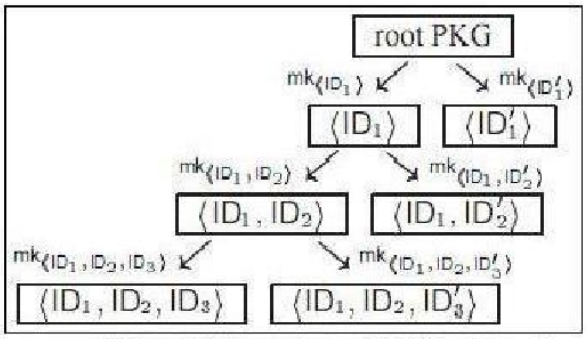


圖2.2-1 Hierarchy of PKG's IDs [2]

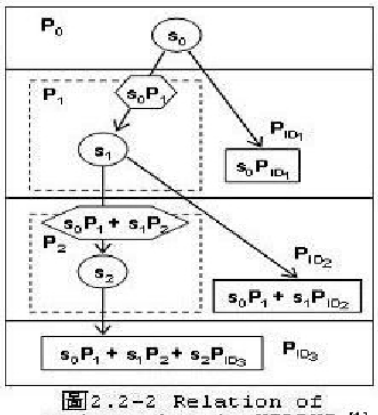


圖2.2-2 Relation of private key in HIDPKI [1]

由於階層化基於身份密碼系統較為複雜，原盼清楚報告，雖節錄僅有的幾個代表性方案，但相關論文甚少又受限所學，恐多疏漏，還懇請大家原諒。接下來我們將介紹階層化基於身份加密(HIBE)、階層化基於身份簽章(HIBS)、及階層化基於身份簽密(HIBES)技術。

2.2.1 Hierarchical ID-based Encryption (HIBE)

本小節介紹 Gentry & Silverberg 之 HIBE 方案[1,3,8]，此方案共分 Root PKG Setup, Lower Level PKG Setup, Extract, Encryption, Decryption 5 階段，將逐一介紹。

Root PKG Setup 階段：

1) PKG 產生基於質數階 q 之加法循環群 G_1 及乘法循環群 G_2，

也產生其雙線性對 $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$。

2)隨機挑選生成元素 $P \in G_1$，並挑選隨機數 $s_0 \in Z_p^*$，

算出系統公鑰 $P_{pub0}=s_0 P_0$，s_0 為 PKG_0 系統主私鑰(MasterKey)。

3)挑選四個雜湊函數，

$H_1:\{0,1\}^* \rightarrow G_1^*$，　　$H_2:G_2^* \rightarrow \{0,1\}^*$，

$H_3:\{0,1\}^n \times \{0,1\}^n \rightarrow Z_q^*$，　$H_4:\{0,1\}^n \rightarrow \{0,1\}^n$.

明文空間 $M=\{0,1\}^n$，密文空間 $C=G_1^j \times \{0,1\}^n \times \{0,1\}^n$，

公開系統參數 $<G_1, G_2, \hat{e}, q, n, P_0, P_{pub0}, H_1, H_2, H_3, H_4>$

Lower Level PKG Setup 階段：

第 i-1 層 PKG 挑選隨機數 s_{i-1}，並算公鑰 $P_{pubi-1}=s_{i-1} P_0$。

Extract 階段：

1)計算第 i 層使用者 $A_{/i}$ 之 $P_{A/i}=H_1(ID_1, ID_2, .., ID_{i-1}, ID_{A/i})$，

計算第 j 層使用者 $B_{/j}$ 之 $P_{B/j}=H_1(ID_1, ID_2, .., ID_{j-1}, ID_{B/j})$，

2)計算第 i 層 $A_{/i}$ 之安全點(Secret Point) $S_{A/i}=S_{i-1}P_{A/i}^{S_{i-1}}$，

計算第 j 層 $B_{/j}$ 之安全點(Secret Point) $S_{B/j}=S_{i-1}P_{B/j}^{S_{i-1}}$，

3)算 $A_{/i}$ 公鑰 $P_{pubA/i}=s_i P_0$，私鑰 $S_{A/i}$ 為 $<S_i, P_{pub0}, P_{pub1}, .., P_{pubi-1}, P_{pubA/i}>$。

算 $B_{/i}$ 公鑰 $P_{pubB/i}=s_i P_0$，私鑰 $S_{B/j}$ 為 $<S_i, P_{pub0}, P_{pub1}, .., P_{pubj-1}, P_{pubB/j}>$。

Encryption 階段：

若使用者 $A_{/i}$ 欲將明文 M 安全地送給另一使用者 $B_{/j}$，則

1)算出 $B_{/j}$ 之公鑰 $P_{B/j}=H_1(ID_1, ID_2, .., ID_{j-1}, ID_{B/j})$，

2)挑選隨機數 σ， 計算 $r=H_3(\sigma,M)$， $g=\hat{e}(P_{pub0}, P_1)\in G_2$，

3)送出密文 $C=<rP_0,rP_2,..,rP_{j-1},rP_{B/j},\sigma\oplus H_2(g^r),M\oplus H_4(\sigma)>$。

Decryption 階段：

若將密文 $C=<rP_0,rP_2,..,rP_{j-1},rP_{B/j},\sigma\oplus H_2(g^r),M\oplus H_4(\sigma)>$ 視為 $<U_0,U_2,..,U_j,V,W>$，收方 $U_{/j}$ 私鑰為 $<S_j,P_{pub0},P_{pub1},..,P_{pubB/j}>$，若 $<U_0,U_2,..,U_j>\in G_1^j$ 則丟棄 C，否則繼續計算。

1)收方計算 $g'=\hat{e}(U_0,S_{B/j})/\prod_{i=2}^{j}\hat{e}(P_{pubi-1},U_i)'$

2)計算 $\sigma=V\oplus H_2(g')$， $M=W\oplus H_4(\sigma)$，

3)計算 $r=H_3(\sigma,M)$，若 $U_0\neq rP_0$ 則丟棄 C，否則解密得 M。

2.2.2 Hierarchical ID-based Signature (HIBS)

本小節介紹 Gentry & Silverberg 之 HIBS 方案[1,8]，此方案共分 Root PKG Setup, Lower Level PKG Setup, Extract, Sign, Verify 5個階段，下面介紹這5個階段。

Root PKG Setup 階段：

1) PKG 產生基於質數階 q 之加法循環群 G_1 及乘法循環群 G_2，也產生其雙線性對 $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$。

2)隨機挑選生成元素 $P\in G_1$，並挑選隨機數 $s_0\in Z_p^*$，算出系統公鑰 $P_{pub0}=s_0P_0$，s_0 為 PKG_0 系統主私鑰(MasterKey)。

3)挑選二個雜湊函數，

$H_1:\{0,1\}^* \rightarrow G_1$,　　$H_3:\{0,1\}^* \rightarrow G_1$,

簽章空間 $S=G_1^{i+1}$，公開系統參數$<G_1, G_2, \dot{e}, P_0, P_{pub0}, H_1, H_3>$

Lower Level PKG Setup 階段：

第 i-1 層 PKG 挑選隨機數 S_{i-1}，並算公鑰 $P_{pubi-1}=S_{i-1}P_0$。

Extract 階段：

1)計算第 i 層使用者 $U_{/i}$ 之 $P_{U/i}=H_1(ID_1, ID_2, .., ID_{i-1}, ID_{U/i})$，

2)計算第 i 層 $U_{/i}$ 之安全點(Secret Point) $S_{U/i}=S_{i-1}+S_{i-1}P_{U/i}$，

3)算 $U_{/i}$ 公鑰 $P_{pubU/i}=S_iP_0$，私鑰 $S_{U/i}$ 為$<S_i, P_{pub0}, P_{pub1}, .., P_{pubi-1}, P_{U/i}>$。

Sign 階段：

若第 i 層使用者 $U_{/i}$ 欲用私鑰 $S_{U/i}$ 為明文M簽章時，則

1)計算 $U_{/i}$ 之 $P_M=H_3(ID_1, ID_2, .., ID_{i-1}, ID_{U/i})$，

2)計算簽章 $Sig(ID_1, ID_2, .., ID_{i-1}, ID_{U/i}, M)=S_{U/i}+S_iP_M$，

3)將簽章 $Sig(ID_1, ID_2, .., ID_{i-1}, ID_{U/i}, M)$，與 $P_{pubU/i}=S_iP_0$ 給收方。

Verify 階段：

使 $(ID_1, ID_2, .., ID_{U/i}, M)$ 之簽章$<Sig, P_{pub1}, .., P_{pubi-1}, P_{pubU/i}>\in S$，

若收方要驗證簽章 $Sig(ID_U\text{-}Tuple, M)$，則收方需確認

$\dot{e}(P_o, Sig)=\dot{e}(P_{pub0}, P_1)\dot{e}(P_{pubi}, P_M)\prod_{i=2}^{i}\dot{e}(P_{pubi-1}, P_i)$

2.2.3 Hierarchical ID-based Signcryption (HIBES)

本小節將介紹 Chow, Yuen, Hui 及 Yiu 等四人之 HIBES 方案[1,9]，而上述四人之研究有兩個方案分別解決兩個問題，本小節目的係觀念介紹，基於篇幅考量，只介紹提供有效前向安全簽密(Forward-secure Signcryption)並能公開驗證與密文鑑別(Public Verifiability & Public Ciphertext Authenticity)的第一方案。此方案共分 Root PKG Setup, Lower Level PKG Setup, Extract, Sign, Encryption, Decryption, Verify 7個階段，下面就來介紹這7個階段。

Root PKG Setup 階段：

1) PKG 產生基於質數階 q 之加法循環群 G_1 及乘法循環群 G_2，

也產生其雙線性對 $\hat{e}: G_1 \times G_1 \rightarrow G_2$。

2) 隨機挑選生成元素 $P \in G_1$，並挑選隨機數 $s_0 \in Z_p^*$，

算出系統公鑰 $P_{pub0}=s_0 P_0$，s_0 為 PKG_0 系統主私鑰(MasterKey)。

3) 挑選三個雜湊函數，

$H_1:\{0,1\}^* \rightarrow G_1$, $H_1:\{0,1\}^* \rightarrow G_1$, $H_3:G_2 \rightarrow \{0,1\}^{k_0+k_1+n}$,

此處 k_0 係 G_1 元素之位元數，

k_1 係第 i 層身份字串最大位元數，

n 係用來簽密之訊息最大位元數。

公開系統參數$<G_1, G_2, \hat{e}, q, P_0, P_{pub0}, H_1, H_2, H_3>$

Lower Level PKG Setup 階段：

第 i-1 層 PKG_{i-1} 挑選隨機數 s_{i-1}，並算出公鑰 $P_{pubi-1}=s_{i-1} P_0$。

Extract 階段：

1)計算第 i 層使用者 $A_{/i}$ 之 $P_{A/i}=H_1(ID_1, ID_2, .., ID_{i-1}, ID_{A/i})$，

計算第 j 層使用者 $B_{/j}$ 之 $P_{B/j}=H_1(ID_1, ID_2, .., ID_{j-1}, ID_{B/j})$，

2)計算第 i 層 $A_{/i}$ 之安全點 (Secret Point) $S_{A/i}=S_{i-1}P_{A/i}^{S_{i-1}}$，

計算第 j 層 $B_{/j}$ 之安全點 (Secret Point) $S_{B/j}=S_{i-1}P_{B/j}^{S_{i-1}}$，

3)算 $A_{/i}$ 之公鑰 $P_{pubA/i}=P_0^{S_i}$，$B_{/j}$ 之公鑰 $P_{pubB/j}=P_0^{S_j}$。

Sign 階段：

若第 i 層使用者 $A_{/i}$ 欲用私鑰 $S_{A/i}$ 為明文M簽章時，則

1)計算 $P_{MA/i}=H_2(M)$，$P_{MA/i}$

2)挑選隨機數 $r\in Z_q^*$，計算 $\sigma=S_{A/i}P_{MA/i}^{r}$，

3)組成簽章 $<\sigma, P_{pub1}, P_{pub2}, \cdots, P_{pubi-1}, P_{MA/i}=P_0^{r}>$，

以及 r 當做加密所需之暫時資料(Ephemeral Data)。

Encryption 階段：

若欲用第 j 層使用者 $B_{/j}$ 公鑰為 $A_{/i}$ 之簽章加密(簽密)，則

1)計算第 j 層使用者 $B_{/j}$ 之 $P_{B/j}=H_1(ID_1, ID_2, .., ID_{j-1}, ID_{B/j})$，

2)組成密文 $C=<P_{B2}^{r}, .., P_{B/j}^{r}, (M|\sigma|A_{/i})\oplus H_3(\overset{\circ}{g}^{r}), P_{pub1}, P_{pub2}, .., P_{pubi-1}, P_{MA/i}>$。

此處 $\overset{\circ}{g}=\hat{e}(P_{pub0}, P_{B1})\in G_1$。

Decryption 階段：

$B_{/j}$ 之私鑰 $S_{B/j}=\prod_{i=1}^{j} P_{U/i}^{S'_{i-1}}$，$A_{/i}$ 之公鑰 $P'_{A/i}=P_0^{S'_i}$，

此處 $1\leq i\leq j-1$。

1) 若密文 $C=<P_{B/2}^{r},..,P_{B/j}^{r},(M|\sigma|A_{/i})\oplus H_3(\hat{g}^{r}),P_{pub1},P_{pub2},..,P_{pubi-1},P_{A/i}>$

視為 $<U_2,..,U_{/j},V,P_{pub1},P_{pub2},..,P_{pubj-1},P_{A/i}>$

則計算 $V\oplus H_3(\hat{e}(P_{MAi},S_{B/j})/\prod_{i=2}^{j}\hat{e}(P'_{pubi-1},U_{/i})=M|\sigma|A_{/i}$

(當 j=1 時 $\prod_{i=2}^{j}\hat{e}(P'_{pubi-1},U_{/i})$ 定義為 G_1 之 Identity 元素)

2) 回傳 $<M,\sigma,A_{/i},P_{pub1},P_{pub2},..,P_{pubj-1},P_{A/i}>$。

Verify 階段：

若任何人要驗證 $A_{/i}$ 之簽章 $<r,\sigma,P_{pub1},P_{pub2},..,P_{pubj-1},P_{A/i}>$，則

1) 計算 $P_M=H_2(M)$，

2) 計算 $P_{A/i}=H_1(ID_1,ID_2,..,ID_{i-1},ID_{A/i})$，此處 $1\leqq i\leqq j$。

3) 若 $\hat{e}(P_0,\sigma)/\prod_{i=2}^{j}\hat{e}(P_{pubi-1},P_{A/i})=\hat{e}(P_{pub0},P_1)\hat{e}(P_{A/i},P_M)$

(當 j=1 時 $\prod_{i=2}^{j}\hat{e}(P_{pubi-1},P_{A/i})$ 定義為 G_1 之 Identity 元素)

則驗證正確。

3 Mechanism

本章主要是利用階層化基於身份密碼系統(HIBC)技術，來對網路節點進行雙向身分鑑別認證(Mutual Authentication)、雙方密鑰協商(Key Agreement)、及安全訊息交換(Secure Data Transaction)方案之研究。

3.1 HIBC-based Mutual Authentication

通訊主體身分鑑別認證(Entity Authentication)機制有許多方式：依據被鑑別主體之相關訊息來分有所是(What you are)、所有(have)、所知(know)；由被鑑別主體個數方向來分有單向(Unilateral)、雙向(Mutual)；若由所採用密碼算法來分有對稱密鑰(Symmetric Secret Key)系統、非對稱公鑰(Asymmetric Public Key)系統；而由有否可信任第三者認證主機介入來分有基於認證主機(Server based)、無認證主機(Server-less)等多種身分鑑別認證方式。

本節討論的是利用主體所知、雙向、非對稱公鑰、無主機之HIBC技術來提出幾種網路節點間之雙向身分鑑別認證(Mutual Authentication)研究方案，分別是利用階層化基於身份身份加密(HIBE)技術、以及利用階層化基於身份簽章(HIBS)技術之雙向身分鑑別認證方案。

3.1.1 HIBE based Mutual Authentication

非對稱密碼系統特性是送方用收方公開在外之公鑰(Public Key)加密訊息，收方再用自己私藏保護之私鑰(Private Key)解密，只有持有私鑰之特定收方才能解密。

若 Alice 與 Bob 想利用 HIBE 進行雙向身分鑑別認證時，

1) Alice 先將代表 Alice 身分之 A 及臨時產生之隨機亂數 N_A (Nonce)利用 Bob 之公鑰加密($HIBE_B$)，附帶將 A,B,N_A經過雜湊運算(Hash)後一起送給 Bob(Challenge)。
2) Bob 收到 Alice 之 Challenge 後，用自己之私鑰解密($HIBE_B^{-1}$)得到代表 Alice 身分之 A 及隨機亂數 N_A，也用相同雜湊運算驗證 N_A雜湊值是否完整未被竄改，有否中間人假冒或重送之 N_A；同樣臨時產生代表 Bob 身分之隨機亂數 N_B及 B，再利用 Alice 之公鑰加密($HIBE_A$)，也將 B,A,N_A, N_B經由雜湊運算後一起送回給 Alice(Response & Challenge)。
3) Alice 收到 Bob 之 Response & Challenge 後，同樣用自己之私鑰解密($HIBE_A^{-1}$)得到代表 Bob 身分之 B 與 N_B，也用相同雜湊運算同理驗證 B,N_B及 Bob 之回應值 A,N_A；再將 B,N_B連同 A,N_A經由雜湊運算後送回給 Bob 當做回應(Response)。

4)Bob 收到 Alice 之回應後，將自己送出之 B 與 N_B，連同 Alice 送來之 A 與 N_A 經過相同雜湊運算以驗證 Alice 回應之 B 與 N_B 是否正確。藉此達到雙方身份鑑別認證機制。

圖 3.1-1 即利用 HIBE 技術來實現雙向身分鑑別之流程。

A－＞B： $HIBE_B(A,N_A)$， $H(A,B,N_A)$

B－＞A： $HIBE_A(B,N_B)$， $H(B,A,N_A,N_B)$

A－＞B： $H(A,B,N_B,N_A)$

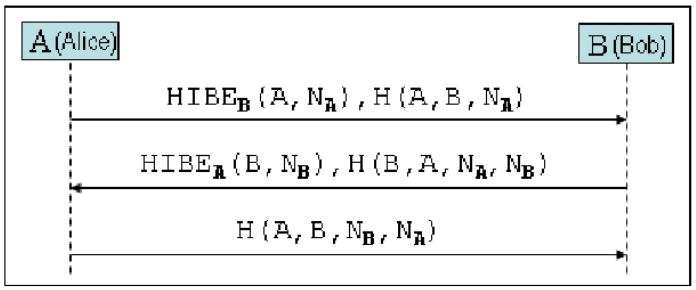


圖 3.1-1 HIBE based Mutual Authentication

註：若 (X,N_X) 長度超過 160 Bits，則改成 $X,HIBE_Y(N_X)$。

3.1.2 HIBS based Mutual Authentication

非對稱式密碼系統的另一個特性，是如果送方用自己之私鑰(Private Key)對訊息簽章，則任意的收方可用送方公開在外之公鑰(Public Key)驗證送方身分。因只擁有私鑰之特定簽章者才能簽章，而藉此達到通訊主體之身分鑑別認證。

若 Alice 與 Bob 想利用 HIBS 進行雙向身分鑑別認證時，則

1)Alice 先產生隨機亂數 N_A(Nonce)，並將代表 Alice 及 Bob 身分之 A,B 及 N_A 利用雜湊運算後一起送給 Bob(Challenge)。

2)Bob 收到 Alice 之訊息後，利用相同雜湊運算驗證 A,B 及 N_A 雜湊值是否完整未被竄改；同樣臨時產生代表 Bob 身分之隨機亂數 N_B，用自己之私鑰對 (N_A,A) 簽章 $(HIBS_B)$ (當 Response)，再將 N_B(Challenge)及 (B,A,N_B) 之雜湊運算值一起送給 Alice。

3)Alice 收到 Bob 之訊息後，用 Bob 之公鑰驗證 $(HIBS_B^{-1})$ Bob 身分及 N_A 回應值，同時得到 Bob 送來之 N_B，也用相同雜湊運算驗證 B,A,N_B 雜湊值是否完整未被竄改；再用自己之私鑰對 Bob 送來之 N_A 及 B 簽章 $(HIBS_A)$，附帶將 (A,B,N_B) 之雜湊值一起送給 Bob(當 Response)。

4)Bob 收到 Alice 之訊息後，用 Alice 之公鑰驗證 $(HIBS_A^{-1})$ Alice 身分及 N_B 回應值，也用相同雜湊運算驗證 A,B,N_B 雜湊值是否完整未被竄改。藉此達到雙方身份鑑別認證機制。

圖 3.1-2 即利用 HIBS 技術來實現雙向身分鑑別之流程。

A－＞B： N_A, $H(A,B,N_A)$

B－＞A： $HIBS_B(N_A,A)$, N_B, $H(B,A,N_B)$

A－＞B： $HIBS_A(N_B,B)$, $H(A,B,N_B)$

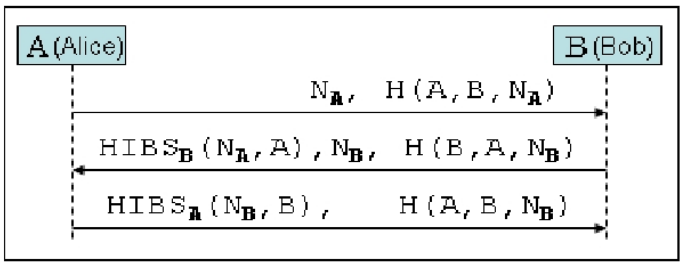


圖 3.1-2 HIBS based Mutual Authentication

註：若 (X,N_X) 長度超過 160 Bits，則改成 $X,HIBS_X(N_X)$。

3.2 Hierarchical ID-based Authenticated Key Agreement

資料安全通訊過程中如資料加解密或含資料驗證等所需密鑰之產生協定可分兩大類：一為密鑰傳送(Key Transport)協定、另為密鑰協商(Key Agreement)協定，而密鑰協商協定又可依參與主體數分為雙方(Two-Party)、多方(Multi-Party)兩類。

本節將提出兩種通訊主體間雙方密鑰協商研究方案，分別是利用階層化基於身份加密(HIBE)技術、與階層化基於身份簽章(HIBS)技術之密鑰協商方案。但因研讀 HIBC 技術之時間很短，白天又要上班，一時還想不出較佳之密鑰協商方案，故懇請容許使用密鑰協商參數(Ephemeral Key Parameter; EKP)代表雙方交換之協商參數繼續報告。

而密鑰協商參數(EKP)之交換機制，容以橢圓曲線密碼系統之會談密鑰協商為例簡要說明。若密鑰協商之雙方為A與B，則

1) A與B先分別挑選隨機亂數 r_A, r_B，然後雙方分別計算己方之密鑰協商參數，$EKP_A=r_AP$，$EKP_B=r_BP$，再送給通訊對方。

2) A與B分別收到對方送來之 EKP_B 與 EKP_A 後，再分別計算出雙方之會談密鑰 $K_{AB}=r_AEKP_B=r_Ar_BP$，$K_{BA}=r_BEKP_A=r_Br_AP$。

3.2.1 HIBE based Authenticated Key Agreement

邵老師說我們可參考 SKEME 方案來改[16]，但後因發現原方案之第一訊息未對 EKP_A 進行訊息驗證，容易遭受偽造攻擊。故增加 $H_{N_A}(EKP_A, A, B)$，讓 Bob 可確認 EKP_A 之真確完整性。若 Alice 與 Bob 想利用 HIBE 進行雙向身分鑑別與雙方密鑰協商時，則

1) Alice 先將代表 Alice 身分之 A 與臨時產生之協商參數 EKP_A 及 N_A，利用 Bob 之公鑰對 (A, N_A) 加密 $(HIBE_B)$，附帶將 EKP_A 及將 (EKP_A, A, B) 利用 N_A 當 Key 之雜湊運算值一起送給 Bob。

2) Bob 收到 Alice 之訊息後，利用自己之私鑰解密 $(HIBE_B^{-1})$ 後得到代表 Alice 之 A, N_A 及 EKP_A，也臨時產生協商參數 EKP_B 及代表 Bob 身分之 B, N_B，利用 Alice 之公鑰對 (B, N_B) 加密 $(HIBE_A)$，附帶將 EKP_B 連同利用代表雙方身分之隨機亂數雜湊值 K_N 當 Key 對 (EKP_A, EKP_B, B, A) 雜湊運算後一起送給 Alice。

3) Alice 收到 Bob 之訊息後，同樣用自己之私鑰解密 $(HIBE_A^{-1})$ 得到 Bob 送回之 B 與 N_B 及 EKP_B，也用 K_N 當 Key 以相同雜湊運算驗證收到之 EKP_A 與 N_A 及 B, N_B 以檢驗 Bob 有否假冒重送；

再同樣利用 K_N 當 Key 對 (EKP_B, EKP_A, A, B) 雜湊運算後一起送給 Bob。鑑別 Bob 身分真確後，即可算出共享會談密鑰 K_{AB}。

4) Bob 收到 Alice 之訊息後，也利用 K_N 當 Key 對 (EKP_B, EKP_A, A, B) 相同雜湊運算驗證 Alice 送回之 EKP_B 與 N_B 及 A, N_A 以檢驗 Alice 有否假冒重送是否正確。若 Alice 身份鑑別真確後，就可算出共享之會談密鑰 K_{BA}。

參考 SKEME 之利用 HIBE 技術實現雙向身分鑑別流程如下。

$A -> B$ ：$HIBE_B(A, N_A)$， EKP_A， $MAC_{N_A}(EKP_A, A, B)$

$B -> A$ ：$HIBE_A(B, N_B)$， EKP_B， $MAC_{K_N}(EKP_A, EKP_B, B, A)$

$A -> B$ ：$MAC_{K_N}(EKP_B, EKP_A, A, B)$

$K_{AB} = F(r_a, N_A, EKP_B)$ $K_{BA} = F(r_b, N_B, EKP_A)$

適當設計 EKP_i 使得最後結果 $K_{AB} = K_{BA} = K$

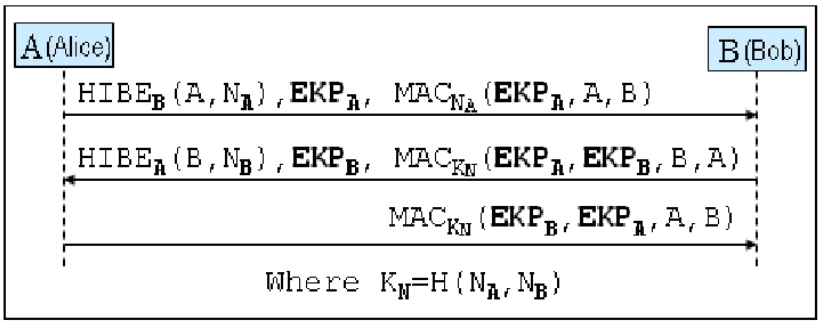


圖 3.2-1 HIBE based Authenticated Key Agreement

3.2.2 HIBS based Authenticated Key Agreement

若 Alice 與 Bob 想利用 HIBS 進行雙方密鑰協商時，則

1) Alice 先臨時產生 N_A 與密鑰協商參數 EKP_A，再用自己之私鑰對 (B, EKP_A) 簽章 $(HIBS_A)$，再將 A 及 N_A（當作 Challenge）一起送給 Bob。

2) Bob 收到 Alice 之訊息後，利用 Alice 之公鑰驗證 $(HIBS_A^{-1})$ Alice 身分並得到 Alice 之 EKP_A 與 N_A；同樣臨時產生代表 Bob 之 N_B 及協商參數 EKP_B，用收到之 EKP_A 與 EKP_B 算出雙方共享之會談密鑰 K_{BA}，再用自己之私鑰對 (A, EKP_B) 簽章 $(HIBS_B)$，附帶將 B, N_B（當 Challenge），並利用 K_{BA} 當 Key 對 (B, A, N_A, N_B) 雜湊運算後一起送給 Alice，雜湊值內含 N_A（當作 Response）。

3) Alice 收到 Bob 之訊息後，同樣用 Bob 之公鑰驗證 $(HIBS_B^{-1})$ 驗證 Bob 身分，得到 Bob 送來之 N_B 及 B 與 EKP_B，也由收到之 EKP_B 與自己之 EKP_A 算出雙方共享之會談密鑰 K_{AB}（$K_{AB}=K_{BA}$），也用相同運算驗證 (B, A, N_A, N_B) 雜湊值是否完整未被竄改，並驗證 Bob 之回應 N_A 是否正確；再用 K_{AB} 當 Key 對 (A, B, N_B, N_A) 雜湊運算後送回給 Bob，雜湊值內含 N_B（當作 Response）。

4) Bob 收到 Alice 之訊息後，利用相同運算驗證 (A, B, N_B, N_A) 雜湊值是否完整未被竄改，並驗證 Alice 之回應 N_B 是否正確。

下面即利用 HIBS 技術來實現雙方密鑰協商之流程。

A －＞ B ： $HIBS_A(B, EKP_A), A, N_A$

B －＞ A ： $HIBS_B(A, EKP_B), B, N_B,\ H_{K_{BA}}(B, A, N_A, N_B)$

A －＞ B ： $H_{K_{AB}}(A, B, N_B, N_A)$

$K_{AB} = F(r_a, N_A, EKP_B)$ $K_{BA} = F(r_b, N_B, EKP_A)$

適當設計 EKP_i 使得最後結果 $K_{AB} = K_{BA} = K$

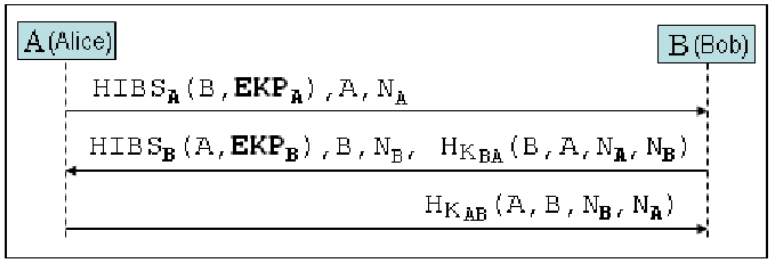


圖 3.2-2 HIBS based Authenticated Key Agreement

註：HIBS 方案安全性稍弱，只能在受限範圍內構思，原本曾設計更安全之 HIBES 方案。但邵老師說計算量會超過而作罷容略。

3.3 HIBC-based Secure Message Exchange

本節將提出利用 HIBC 技術對網路節點間安全資料交換之研究方案。由於非對稱之公鑰密碼算法雖然對於加密解密與簽章驗證能提供較安全之機制，但其運算量卻遠比傳統對稱密碼算法多，故一般是用混合式密碼算法將非對稱之公鑰密碼算法用於簽章驗證達到通訊主體身分與資料來源之鑑別認證(Authentication)保護或對共享密鑰協商之協商參數加密解密；而真正要傳送大量資料時之加解密運算勢必要改用傳統對稱密碼算法(如較快之 AES Rijndael 算法)以達到資料保密性(Data Confidentiality)；或再利用對訊息之雜湊運算進行資料完整性(Data Integrity)保護。

資料通訊又分為非連接導向(Connection-less)與連接導向(Connection Orient)通訊。連接導向(Connection Orient)通訊係指傳輸資料前會先建立雙方之連線或會談(Connection or Session)，在未中斷連線或會談前，所有資料都屬於同一連線會談(Session ID)。而非連接導向(Connection-less)之通訊係指傳輸資料前不需先建立雙方之連線或會談即可送出，每個資料都是獨立資料。當連接導向之通訊雙方進行身分鑑別認證並進行密鑰協商後，就可利用協商後之會談密鑰(Session Key)進行後續整個會談(Session)之安全資料交換了。但若是非連接導向之通訊時，則不需先密鑰協商，而可利用簽密算法對加密訊息之對稱密鑰運算得簽密值，再連同用密鑰加密後之訊息一起送給收方。

邵老師說我們只須研究非連接導向部分，故本節只討論利用HIBC技術對非連接導向之安全資料交換方案，因為非連接導向(Connection-less)通訊沒有連線編號與會談編號問題，若也不考慮封包切割組合順序問題，就相對簡單許多。方案如下：

假設A要以非連接導向方式安全地送訊息給B，則如下處理：

1) A將加密訊息M所需之密鑰K_m利用階層化簽密(HIBES)算法簽密後，再用對稱密鑰加密訊息，若希有資料完整性，則再將

對密鑰之簽密值($HIBES_{BA}(K_m)$)、初始向量(IV_m)、加密後訊息($E_{Km}(M_A)$)合併經由雜湊運算後一起送給B。

2) B收到A安全處理後之訊息串後，先用自己私鑰解密並用B之公鑰驗證B身分，解簽密得到密鑰K_m，配合IV_m與K_m對密文$E_{Km}(M_A)$解密得到訊息M_A，再用相同雜湊運算驗證訊息M_A,初始向量(IV_m)，密鑰K_m之資料完整性。

若希望更安全，則也將密鑰分成兩部份，一為加解密用(Data Encryption Key; EK)，一為資料驗證碼(Message Authentication Code; MAC)用之(Integrity Key; IK)。故前方案也可改成：

1) A先將K_m經過運算分解成EK與IK，將將加密訊息M所需之密鑰K_m利用階層化簽密(HIBES)算法簽密後，再用對稱密鑰EK加密訊息M_A，若希有資料完整性，則再將對密鑰之簽密值($HIBES_{BA}(K_m)$)、初始向量(IV_m)、加密後訊息($E_{Km}(M_A)$)合併經由利用IK當Key之雜湊運算後一起送給B。

2) B收到A安全處理後之訊息串後，先用自己私鑰解密並用B之公鑰驗證B身分，解簽密得到密鑰K_m，也將K_m經過相同運算分解成EK與IK，配合IV_m與EK對密文$E_{EK}(M_A)$解密得到訊息M_A，再利用IK當Key之相同雜湊運算驗證訊息M_A,初始向量(IV_m)，密鑰K_m之資料完整性。

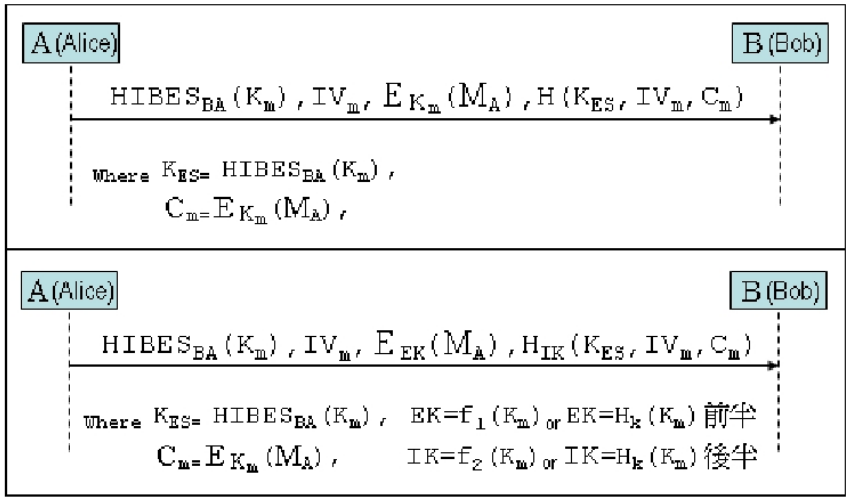


圖 3.3-1(a)(b) Hierarchical ID-based Secure Data Transaction

4 Application

本章介紹利用 HIBC 技術在 SSL／TLS 安全協定及無線網路中 IEEE 802.1X, 802.11i 安全協定之部份應用，並嘗試針對協定弱點提出部分強化之想法。

4.1 Hierarchical Identity-based TLS

在開始討論利用 HIBC 技術在 SSL／TLS 協定之部份應用前，為了讓大家對將 SSL／TLS 協定有一大致了解，故將先介紹 SSL／TLS 主要內容。

4.1.1 Introduction of SSL／TLS Protocol [11]

SSL(Secure Socket Layer)／TLS(Transport Layer Security)是一種為 TCP/IP 網路傳輸層(Transport Layer)上之應用層服務協定提供安全通訊之協議。SSL 最先是由 Netscape 公司設計出來，改進到第三版(SSL v3)後，網際網路標準制定單位(Internet Engineering Task Force; IETF)將 SSL v3 協定標準納入改名為 TLS v1 協定標準。

SSL／TLS 協定堆疊(Protocol Stack)分為上方之交握層及下方之記錄層兩層，係由後述五個子協定組成，SSL／TLS 協定組彼此關係，及與 TCP/IP 協定堆疊關係請參考圖 4.1-1。

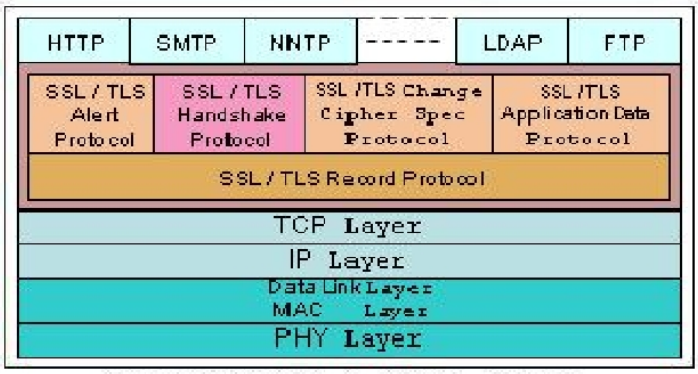


圖 4.1-1 SSL / TLS Protocol Stack in TCP / IP

1) 紀錄協定(Record Protocol)：

負責根據交握協定協商之參數，進行資料之切割組合、壓縮解壓縮、資料驗證(Message Authentication)與加解密運算、計算密鑰組(Key Block)，提供安全連線之私密性(Privacy)與可靠性(Reliability)等功能。紀錄協定上方有下面四個被服務之客戶協定。

2) 交握協定(Handshake Protocol)：

負責協商雙方都支援之密碼組(Cipher Suite)、壓縮法等演算參數，提供通訊主體身分鑑別及雙方會談密鑰協商，及協商過程資料安全保護等功能。

3) 變更加密規格協定(Change Cipher Spec Protocol)：

負責變更加密規格狀態。

4) 警示協定(Alert Protocol)：

負責提供異常警示訊息。

5)應用資料協定(Application Data Protocol)

負責將由應用層服務協定收到之資料送給記錄層切割壓縮加密，或是反向處理後送回給應用層協定。

SSL／TLS 協定堆疊中之交握協定(Handshake Protocol)負責提供通訊主體間雙向身分鑑別認證與會談密鑰協商與計算，這部分正是本論文研究重點，故將針對這部份稍作進一步說明。交握協定中有兩個通訊主體，各為 Server 與 Client，以下將分別說明圖 4.1-2 標示之四個階段。

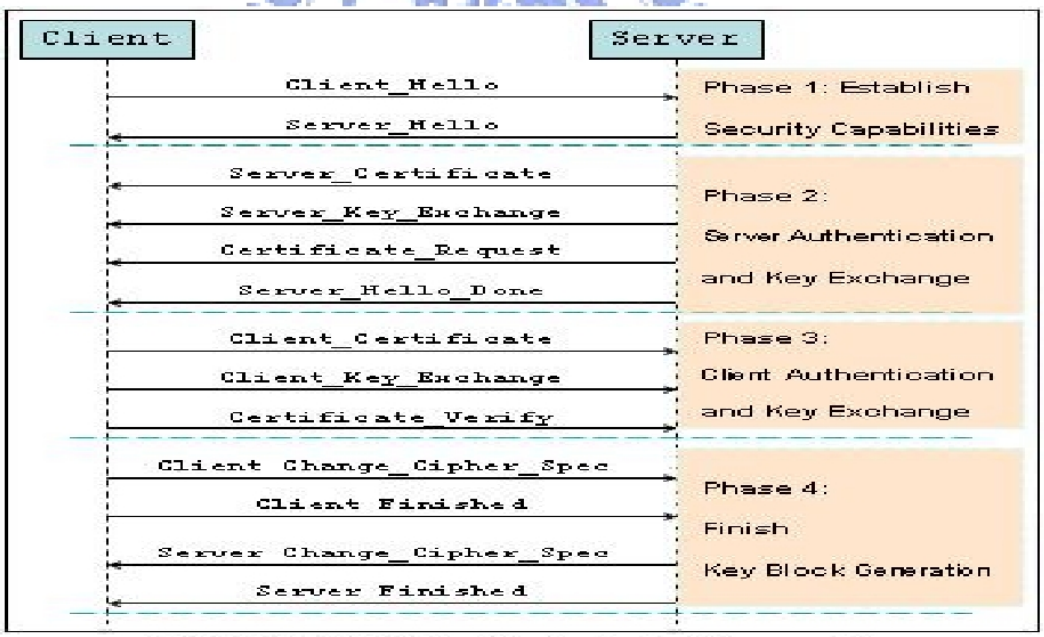


圖 4.1-2 SSL/TLS Handshake Protocol Message Flow

Phase 1：

主要是雙方協商出後續安全通訊能力資訊。內含 SSL／TLS 版本、會談編號(Session ID)、密碼組算法(Cipher Suite)、壓縮

算法(Compression Method)、以及代表身分之隨機亂數(Nonce)。

1) Client_Hello：Client 首先送 Client_Hello 給 Server，告知 Client 能提供之一些安全能力組合及壓縮方法等。

2) Server_Hello：Server 收到 Client_Hello 後，在自己能匹配之安全能力中選擇一組，用 Server_Hello 回報給 Client，後續就用此組安全能力建立安全通訊。

Phase 2：

主要是 Server 提出代表自己身份的憑證或鑑別資訊，以及計算會談密鑰協商之臨時公開協商值等參數。

3) Server_Certificate：Server 送 Server_Certificate 給 Client，內含 Server 經由 CA 簽章之憑證及公鑰。

4) Server_Key_Exchange：Server 送 Server_Key_Exchange 給 Client，內含依據 Hello 協商後之密鑰協商方案臨時公開協商值。目前支援五種密鑰交換方式，我們將討論最安全之 Ephemeral Diffie-Hellman 方式，經由雙方共同交換臨時公開協商值，並計算產生出共享之會談密鑰。

5) Certificate_Request：若想要 Client 也提供憑證，則 Server 送 Certificate_Request 向 Client 提出要求。

6) Server_Hello_Done：Server 送 Server_Hello_Done 給 Client，告知 Server 端身分鑑別及密鑰協商訊息已送完。

Phase 3：

主要是 Client 提出代表自己身份的憑證或鑑別資訊，以及計算會談密鑰協商之臨時公開協商值等參數。

7) Client_Certificate:Client 送 Client_Certificate 給 Server，內含 Client 經由 CA 簽章之憑證及公鑰。

8) Client_Key_Exchange：Client 送 Client_Key_Exchange 給 Server，內含依據 Hello 協商後之密鑰協商方案臨時公開協商值。當雙方完成身分鑑別與密鑰協商後，就可準備算出雙方共享之 Pre-master Secret (PMS)，再算出 Master Secret，接者刪除 RAM 中 PMS。

9) Certificate_Verify：當 Client 用憑證來當簽名時，才送此訊息。內含對 Client 產生之 Master Secret 雜湊值簽名，供 Server 驗證 Client 之憑證是否有效。

Phase 4：

主要是 Client 與 Server 雙方變更加密規格狀態，並驗證身份鑑別與密鑰協商過程是否成功，準備產生密鑰組。

10) Change_Cipher_Spec：Client 與 Server 互送此訊息，將加密規格 Pending 狀態改成 Current 狀態。

11) Finished：Client 與 Server 互送 Finished 訊息，以驗證前面協商過程之身分鑑別與密鑰協商內容資訊之雜湊值

驗證資料完整真確性(Data Integrity)，同時產生後續所需之密鑰組(Key Block)。

以上是標準中 13 個訊息的簡要說明，看起來有點複雜，其實只有 4 個方向之訊息交換，請參考圖 4.1-3。

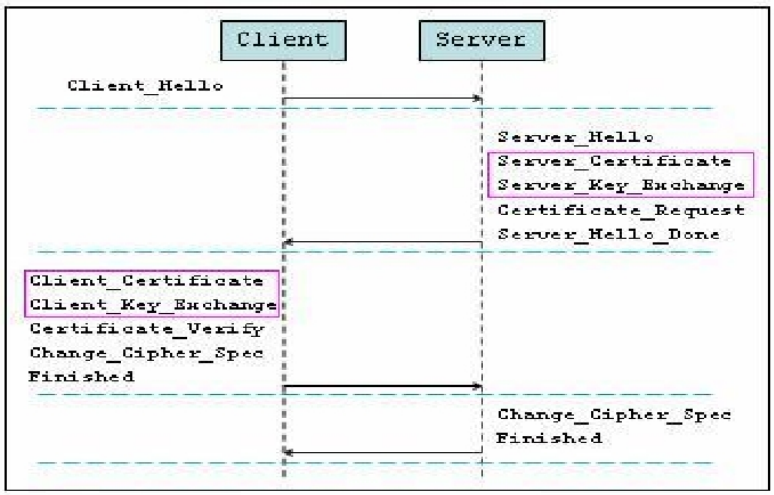


圖 4.1-3 SSL / TLS Handshake Protocol 4-way Message Flow

4.1.2 Hierarchical Identity-based TLS

若要利用 HIBC 技術在 SSL／TLS 協定上實現雙向身分鑑別與密鑰協商，則主要是圖 4.1-3 中深紫色框內訊息稍作部分修改。

由於原本 TLS 是使用 PKI X.509 憑證，但在 HIDPKI 架構，則利用經由 PKG 產生包含 PKG 私密參數之用戶私鑰來簽章，也可具有憑證功能。而爲了同時達到雙向身分鑑別認證，與密鑰協商功能，我們利用 Key_Exchange 訊息來進行雙向身份鑑別，同時交換會談密鑰協商所需之協商參數值(Ephemeral Key Parameter; EKP)。而在雙方 Key_Exchange 之交換訊息中，我們的想法是：

1) Server 送完協商之 Hello 訊息後，也立即臨時產生協商參數 EKP_S 及 N_S，利用自己之私鑰對前面雙方交換之 Hello 訊息及 DH 協商相關參數之雜湊值 $H(Hello_C, Hello_S, DHParam_S)$ 連同協商參數 EKP_S 一起簽章 $(HIBS_S)$ 後，附帶將 S, N_S 及 DH 協商相關參數 $(DHParam_S)$ 一起送給 Client。

2) Client 收到 Server_Key_Exchange 與 Server_Hello_Done 訊息後，利用 Server 之公鑰驗證 $(HIBS_S^{-1})$ Server 身分，得到 Server 之 EKP_S 與 DH 協商相關參數，並用相同雜湊運算驗證 $(Hello_C, Hello_S, DHParam_S)$ 雜湊值是否完整未被竄改；若驗證 Server 身份無誤後，則立即先臨時產生協商參數 EKP_C 與代表 Client 身分之 C 與 N_C，並將收到之 EKP_S 與 EKP_C 算出雙方共享之會談密鑰 K_{CS} (由 Pre-Master Secret 算出 Master Secret)，再用私鑰對前面交換之訊息及 DH 協商相關參數之訊息驗證碼 MAC_{KCS} (Server_Key_Exchange Message|$DHParam_C$) 連同協商參數 EKP_C 一起簽章 $(HIBS_C)$ 後，附帶將 C, N_C 及 DH 協商相關參數 $(DHParam_C)$ 一起送給 Server。

3) Server 收到訊息後，同樣用 Client 之公鑰驗證 $(HIBS_B^{-1})$ 驗證 Client 身分，得到 Client 送來之 EKP_C，並用相同運算驗證 MAC_{KCS} (Server_Key_Exchange Message|$DHParam_C$) 是否完整未被竄改；若驗證 Client 身份無誤，就由收到之 EKP_C 與自

己之 EKP_S 算出雙方共享之會談密鑰 K_{SC}($K_{SC}=K_{CS}$)，接者再將 K_{AB} 與 (S,C,N_C,N_S) 串併，連同前面之交換訊息與 Server 之代表字串及 K_{AB} 經 SHA 雜湊運算值，一起再用 SHA 雜湊運算後送回給 Server 驗證。藉此確認前面協商過程是否完整正確。

下面即雙方交換之 Key_Exchange 訊息：

Server_Key_Exchange：

$HIBS_S(H(Hello_C|Hello_S|DHParam_s),EKP_S),S,N_S,DHParam_s$

Client_Key_Exchange：

$HIBS_C(MAC_{KCS}(Server\_Key\_Exchange\ Nessage|DHParam_C),EKP_C),C,N_C,DHParam_C$

Server_之 Finish：

$SHA(K_{SC}+(S,C,N_C,N_S)+Pad2+SHA(Handshake\_Message+Sender+K_{SC}+Pad1))$

4.2 IEEE 802.1X & 802.11i with Hierarchical Identity-based Crypto-protection

在討論利用 HIBC 技術在 IEEE 802.1X、802.11i 協定之應用前，也將先簡要介紹 IEEE 802.1X、802.11i 會用到之 EAP、EAP-TLS 協定，再介紹 IEEE 802.1X、802.11i 協定，再來討論利用 HIBC 技術在 IEEE 802.1X、802.11i 之應用，最後對 IEEE 802.1X、802.11i 協定中之弱點加以強化。

4.2.1 Introduction of EAP & EAP-TLS Protocol

1) Extensible Authentication Protocol(EAP) [12]

可擴展認證協議是一個認證框架 (Authentication Framework)，可支援多種身分鑑別認證方法，是執行在資料鏈階層(Data Link Layer)上之協定堆疊。

可擴展認證協議為了對上層能提供多種身分鑑別認證方法的協商彈性，對下支援多種資料鏈結層網路之接續方式，而設計了一個多工模組(EAP Multiplexing Model)架構，此一模組內包含下列4種元件，請參考圖 4.2-1。

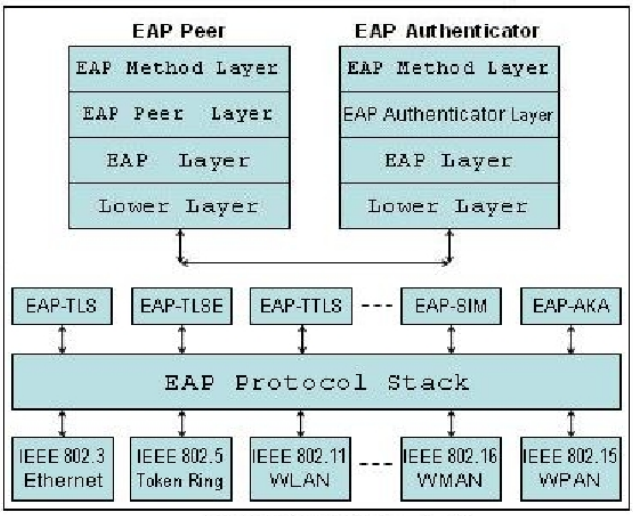


圖4.2-1 EAP Multiplexing Model

◦EAP Method Layer：

負責實現身分鑑別認證演算法，也負責與EAP Peer & Authenticator Layer間EAP Message之送收，若未支

援資料之切割組合，則交由其上層之身份鑑別認證協定處理。

◦EAP Peer & Authenticator Layer：

EAP 協定可允許通訊主體之角色為 Peer 或 Authenticator 甚或兩者同時存在，而此層則負責不同角色之功能處理。

◦EAP Layer：

負責與 Lower Layer 間之 EAP Packet 送收，實現了重複偵測與重傳機制，同時負責對與 EAP Peer & Authenticator Layer 之間 EAP Message 之送收。

◦Lower Layer：

負責送收 Peer 與 Authenticator 間之 EAP Frame，下層可支援多種資料鏈結層(Data Link Layer)協定。

而若有 Authentication Server 與 Authenticator 同時存在時，一般會將 Authenticator 當作 Pass-Through 方式運作，請參考圖 4.2-2。

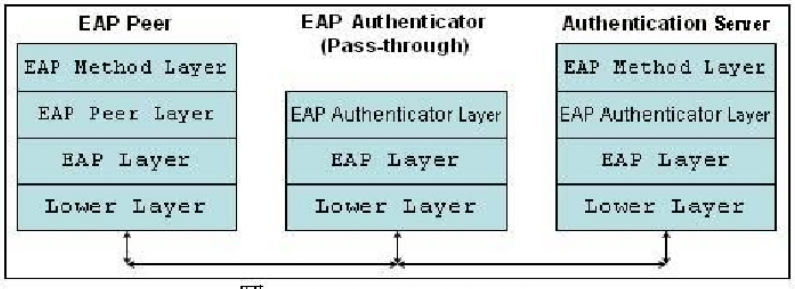


圖 4.2-2 Pass-through Authenticator

若是端點對端點方式時，則參考圖 4.2-3 方式運作。

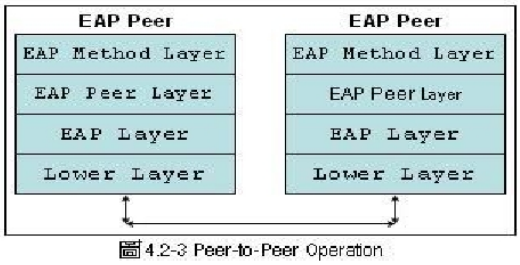


圖 4.2-3 Peer-to-Peer Operation

至於 EAP 協定內通訊主體間之訊息交換，請容到 EAP-TLS 一併討論，其實只是訊息外面包上不同協定不同作用之詢息頭。

EAP 協定不僅可以用於無線區域網，還可以用於有線網路，由圖 4.2-1 可見其應用之擴展性。最近，WPA 和 WPA2 標準已經採納 5 種 EAP 方法作為認證機制。現在大約有 40 種不同的 EAP 身份鑑別認證方法。IETF 的 RFC 中定義的方法包括：EAP-MD5, EAP-OTP, EAP-GTC, EAP-TLS, EAP-SIM,和 EAP-AKA, 還包括一些廠商提供的方法和新的建議。無線網路中常用的方法包括 EAP-TLS、EAP-TTLS、EAP-SIM、EAP-AKA、和 LEAP 等。同樣對下層之不同資料鏈結層網路也有相當大之彈性支援。

2) EAP-TLS Protocol [13]

EAP-TLS 協定是結合了 EAP 及 TLS 兩種協定功能所形成之一種 Client/ Server 架構之協定，因具有較安全之雙向身分鑑別認證與會談密鑰協商功能，故 IEEE 802.1X 已將此協定納入。雖然目前也有一些不同的雙向身分鑑別認證或 EAP-TLS 變形，但本論文只是一種研究的學習，故未多列入討論。

因原本之 EAP-TLS 協定採用 PKI 之憑證來與 Radius 認證伺服主機間進行身分鑑別認證，但使用者端之憑證取得與驗證在認證時的不便與負載造成它的致命傷。此時正好是 IDPKI 可表現的時候，此部分將於後面討論。

我們先討論 EAP-TLS 協定中身份鑑別認證流程之訊息往來，由圖 4.2-4 看出 EAP-TLS 協定在身份鑑別認證訊息交換過程，主要還是 TLS 之訊息，只是外面再包上 EAP 及 AAA 之詢息頭。

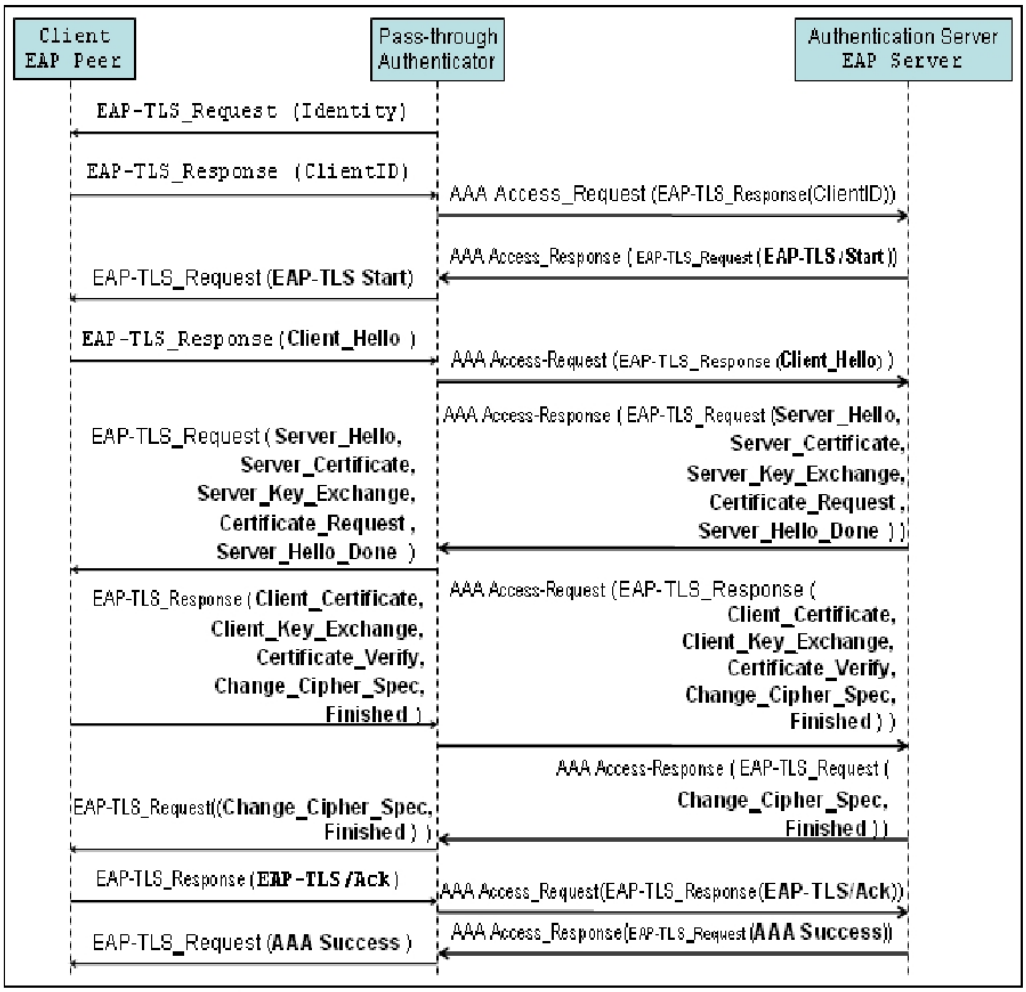


圖 4.2-4 EAP-TLS Authentication Message Flow

4.2.2 Introduction of IEEE 802.1X and 802.11i Protocol

1) IEEE 802.1X Protocol [14]

IEEE 802.1X 協定又稱為基於埠口之網路存取控制協定(Port based Network Access Protocol)，是 IEEE 制定使用者接入網路的身分鑑別認證標準。不但可用於無線網路，也可用於有線網路。在無線區域網路中引用 IEEE 802.1X 協定時，包含三個主體角色，分別是 Supplicant(即 Mobile Node)、Authenticator(即 Access Point)、Authentication Server(即 RADIUS 或 DIAMETER Server)。主要是利用 EAP 之身分鑑別認證框架及後端認證主機(Authentication Server)間，進行身分鑑別認證並產生共享會談密鑰。

若是使用 EAP-TLS 協定來進行身分鑑別認證，則只需將 4.2.1 小節中 EAP-TLS 協定之三主體名稱改為 802.1X 之三個主體名稱即可，而其身份鑑別認證流程完全相同。只是在最前面會先有 Mobile Node 掃描(Probe)尋找訊號最佳可用基地台，再進行 802.11 原版身份鑑別，接者 Mobile Node 會與 AP 進行 Association 預連線，並發出 EAPoL Start 訊息，後續就和 EAP-TLS 訊息完全相同。當 EAP-TLS 協定成功後，Mobile

Node 與 Authentication Server 間就會產生共享之 Master Session Key(MSK)。

而在身分鑑別認證過程中，是經由 Unauthorized Port 進行認證詢息之交換，待通過身分鑑別認證，並協商產生共享之 MSK 後，即可經由 Authorized Port 進行安全通訊。

2) IEEE 802.11i Protocol [15]

IEEE 802.11i 是為了強化 802.11 的安全加密功能而制定的修正案，同時也為了讓無線使用者確認基地台收到認證主機之 MSK 與認證主機直接送給自己之 MSK 是否正確，同時再與基地台間利用四步交握協商產生雙方共享之密鑰組，若需要還可再協商產生群組密鑰。WLAN 引用兩種協定後之訊息交換詳見圖 4.2-5。

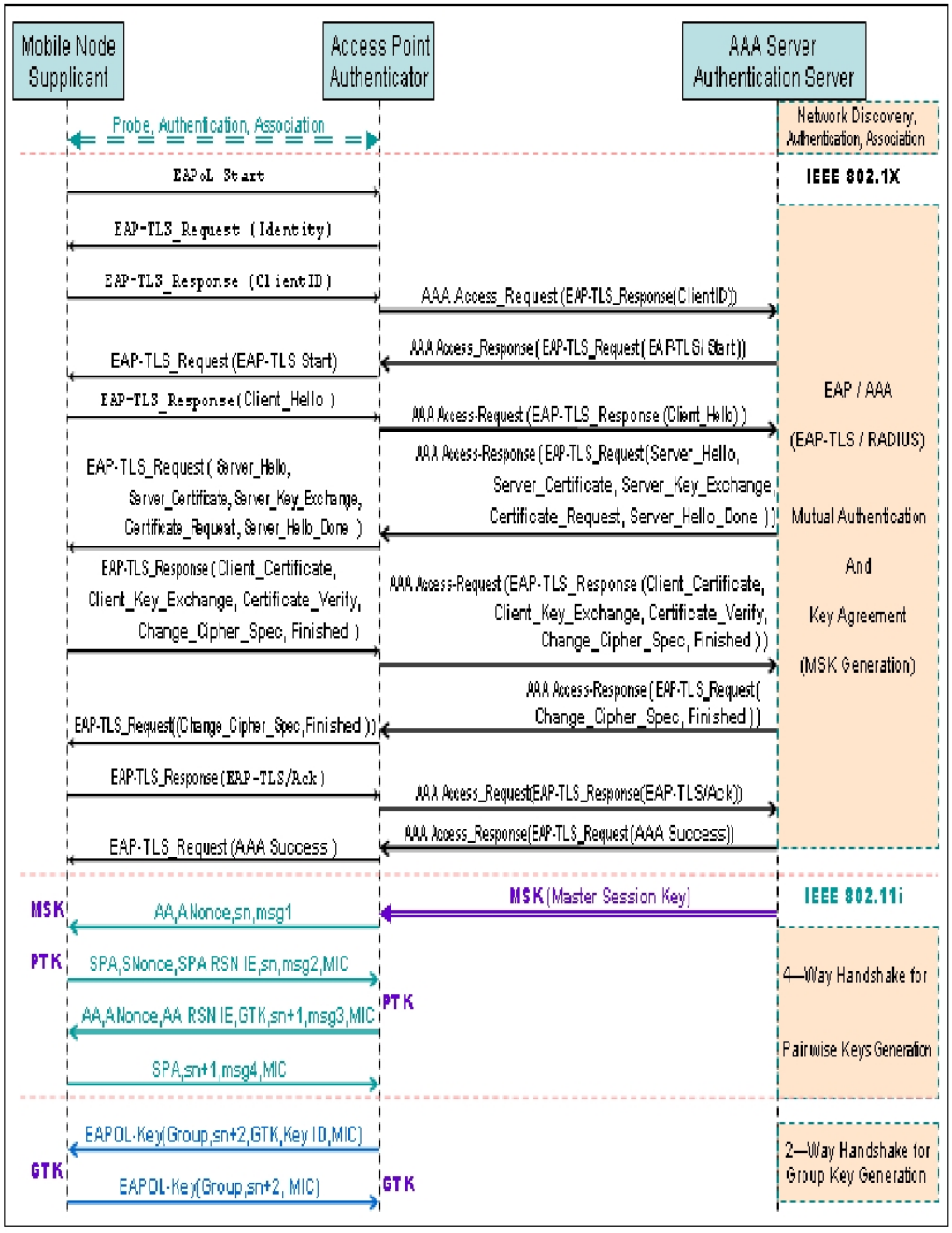


圖 4.2-5 IEEE 802.1X and IEEE 802.11i Message Flow

由前圖可看出，無限區網引用 IEEE 802.1X 及 802.11i 後，其身份鑑別認証與連線上網過程主要分為下列幾步驟。

1) IEEE 802.11 尋找可用網路，原版 802.11 認證，網路預連。

2) IEEE 802.1X MN 與 AAA 間雙向認證與密鑰協商。

3) IEEE 802.11i 四步交握雙方會談密鑰協商。

4) IEEE 802.11i 兩步交握群組密鑰協商。

5) 開始利用前面產生之密碼安全算法進行安全通訊。

4.2.3 Using HIBC Technology in IEEE 802.1X

若利用 HIBC 技術應用在 IEEE 802.1X 中 EAP-TLS 之雙向身份鑑別認證與雙方密鑰協商過程，則與前面 TLS 方式相同。也在 Key_Exchange 訊息中，同時交換密鑰協商所需之 EKP 時，同時達到雙方身份鑑別認證功能，交換訊息請參考第 4.1.2 小節。

Server_Key_Exchange：

$HIBS_S(H(Hello_C|Hello_S|DHParam_S),EKP_S),S,N_S,DHParam_S$

Client_Key_Exchange：

$HIBS_C(MAC_{KCS}(Server\_Key\_Exchange\ Nessage|DHParam_C),EKP_C),C,N_C,DHParam_C$

Server_之 Finish：

$SHA(K_{SC}+(S,C,N_C,N_S)+Pad2+\ SHA(Handshake\_Message+Sender+K_{SC}+Pad1))$

4.2.4 Improvement of IEEE 802.1X & 802.11i

經過仔細分析 IEEE 802.1X 及 802.11i 之訊息交換過程，發現進行 IEEE 802.1X 過程中，基地台(AP)是以穿透通過(Pass-through)方式代為雙方傳送訊息，當完成 MN 與 AAA 間之協商後就產生了雙方共享之 Master Session Key(MSK)。但此 MSK 僅在 MN 與 AAA 間使用，AP 與 MN 間也須保密，故 AAA 也將 MSK 送給 AP，讓 AP 與 MN 間以四向交握(4-Way Handshake)協商出雙方共享之密鑰組。但發現在 802.11i 協定中四向交握的第一個由 AP 送給 MN 之訊息並未經過身份鑑別，故若有惡意之 AP 假冒，不斷送不同之 ANonce 訊息，MN 收到後新的 ANonce 訊息後，為能協商出雙方共享之 Pairwise Transient Key (PTK)，就必須再算新的一組 SNonce，而讓資源很快就被耗盡。

故為了讓四向交握中 AP 送給 MN 的第一個訊息具有身分鑑別功能，我們利用只有真確身分之 AP 會拿到 MSK，故可用將 MSK 放入訊息驗證碼(MAC)中運算，只有收到相同 MSK 值者能算出相同之 MAC 值，來鑑別 AP 身分。經改善後之訊息如下：

$$AA, ANonce, sn, msg1, \mathbf{MAC}_{MSK}(AA, ANonce, sn, msg1)$$

其實也還發現有其他弱點，但受限時間，只做此部份之問題解決之研究學習。

5 Conclusion

5.1 Accomplishment

本論文首先對 IBC 及 HIBC 技術做一介紹，接者利用 HIBC 技術進行雙向身分鑑別認證(Mutual Authentication)、雙方密鑰協商(Key Agreement)、及安全資料交換方案之研究。再利用 HIBC 技術在 SSL/TLS 協定、及 IEEE 802.1X、IEEE 802.11i 協定之部份應用研究，並針對協定弱點提出強化想法。

5.2 Future Work

HIDPKI 技術在未來之研究方向，可能在跨管理區域之行動通訊節點間之密鑰協商方案會是一個研究熱點。可惜白天上班，最近工作又較忙，研究時間非常緊湊，故原本想嘗試研究較複雜基於 HIDPKI 技術之密鑰協商方案，一時間尚未想出較佳方案，且原也想嘗試學習安全協定形式化分析安全性證明及系統模擬的部分，也只能留下遺憾了，相信大家必會有許多讓人讚嘆之精彩研究成果。

References

[1] Fu-Kuo Tseng, John K. Zao, Yung-Hsiang Liu, Fang-Po Kuo, "Halo: A Hierarchical Identity-Based Public Key Infrastructure for Peer-to-Peer Opportunistic Collaboration", National Chiao Tung University, March 2008.

[2] Martin Gagne. "Identity-Based Encryption: a Survey", RSA Laboratories Cryptobytes, Spring 2003.

[3] Ratna Dutta, Rana Barua, Palash Sarkar. "Pairing-Based Cryptographic Protocols : A Survey", http://eprint.iacr.org/2004/064, Jun 2004.

[4] J Baek, J Newmarch, R Safavi-Naini, W Susilo. " A Survey on ID-Based Cryptographic Primitives ", http://eprint.iacr.org/2005/094, Jun 2005.

[5] D. Boneh, M. Franklin. "Identity Based Encryption from Weil Pairing", SIAM Journal of Computing, 32, 586-615, 2003.

[6] F. Hess. "Efficient Identity Based Signature Schemes Based on Pairings", http://www.springerlink.com/content/yg6xvk9lpcqe5784/, Springer Berlin , Heidelberg, 2003.

[7] Noel McCullagh and Paulo S.L.M. Barreto, "A New Two-Party Identity-Based Authenticated Key Agreement", http://eprint.iacr.org/2004/122, Feb 2005.

[8] C Gentry, A Silverberg. "Hierarchical ID-Based Cryptography", Advances in Cryptology — ASIACRYPT 2002.

[9] 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.

[10] X. Boyen, L. Martin. "Identity-Based Cryptography Standard (IBCS)", IETF RFC:5091, December 2007.

[11] T. Dierks, E. Rescorla. "The Transport Layer Security (TLS) Protocol Version 1.1", IETF RFC:4346, April 2006.

[12] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Ed.. "Extensible Authentication Protocol (EAP)", IETF RFC:3748, June 2004.

[13] B. Aboba, D. Simon. "PPP EAP TLS Authentication Protocol", IETF RFC:2716, October 1999.

[14] IEEE. "IEEE Std 802.1X-2004 : IEEE Standard for Local and metropolitan area networks Port-Based Network Access Control - Revision of 802.1X-2001", 2004.

[15] IEEE. "IEEE Std 802.11i-2004 : International Standard - Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) .specifications AMENDMENT 6: Medium Access Control (MAC) Security Enhancements", December 2006.

[16] Colin Boyd, Anish Mathuria. "Protocols for Authentication & Key Establishment", Springer, Berlin Heidelberg, 2003.