國立暨南國際大學資訊工程學系

碩士論文

使用非對稱式密碼增強會議初始協定認證機制

Enhancing the Security of SIP Credentials in a VoIP System with Asymmetric Cryptography

指導教授：吳坤熹博士

研究生：黃信富

中華民國九十九年六月

國立暨南國際大學資訊工程學系

碩士論文

使用非對稱式密碼增強會議初始協定認證機制

Enhancing the Security of SIP Credentials in a VoIP System with Asymmetric Cryptography

指導教授：吳坤熹博士

研究生：黃信富

中華民國九十九年六月

國立暨南國際大學碩士論文考試審定書

資訊工程 學系

研究生 黃信富 所提之論文

使用非對稱式密碼增強會議初始協定認證機制
Enhancing the Security of SIP Credentials in a VoIP System with Asymmetric Cryptography

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

許靜芳 委員兼召集人

吳坤熹 委員

李佩君 委員

中華民國 99 年 06 月 17 日

博碩士論文電子檔案上網授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之論文為授權人在 暨南國際大學 資訊工程學系

98 學年度第 2 學期取得 **碩士** 學位之論文。

論文題目：使用非對稱式密碼增強會議初始協定認證機制

指導教授：吳坤熹 博士

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館及本人畢業學校圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

- 讀者基非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人： 黃信富

簽　名： 黃信富 

中華民國 99 年 07 月 19 日

誌謝

隨著碩士論文的出版，我的研究所生活也即將畫上休止符。在此，最要感謝的人就是我的碩士論文指導教授 吳坤熹博士，老師總是很用心、也很有耐心的指導學生做研究，在平日的生活中也給予學生很多幫助與省思，做事嚴謹的態度更讓學生得以加速成長茁壯，學生以身為老師實驗室的一份子為傲。感謝國立成功大學資訊工程學系 許靜芳教授與本校電機工程學系 李佩君教授能夠撥冗前來擔任學生論文的口試委員，給予學生不同角度的思考方向，使得學生的論文得以更加完整。

感謝惟綸、鐘逸、文仁、嘉裕、政霖、汎嘉等學長在學弟剛進入實驗室對環境還很陌生的時候，帶領著我熟悉暨大的生活環境，也在適當的時候協助學弟解決生活與課業問題。謝謝韋立、韋勳、韋霖、瑋勵、霓雅、筱婷等同學，我永遠不會忘記大夥兒一起修課、做研究，邁向畢業大道，很榮幸能跟大家當同學。感謝創宏、冠達、俊克、豈嘉、麗雯等學弟妹，因為有你們協助處理實驗室事務，讓我們可以在這一年得以專心衝刺論文。

最後，謝謝我的父親與母親，您們一直支持著我，讓我可以無後顧之憂專心於研究中，如果沒有您們就不會有今天的我，您們是我人生最堅強的後盾，謹以此獻給我最敬愛父母。

論文名稱：使用非對稱式密碼增強會議初始協定認證機制

校院系：國立暨南國際大學科技學院資訊工程學系　　頁數：59

畢業時間：99 年 06 月　　學位別：碩士

研究生：黃信富　　指導教授：吳坤熹 博士

論文摘要

近年來，由於網路硬體建設的普及，網路上的各種多媒體應用服務如雨後春筍般的誕生，其中之一殺手級的應用即是網路電話(Voice over Internet Protocol，簡稱 VoIP)。在眾多的網路電話標準協定中，會議初始協定(Session Initiation Protocol，簡稱 SIP)因其功能擴充性較其它的標準協定佳、開發易，所以越來越多廠商用以發展產品。但是會議初始協定標準文件內所建議採行的使用者身分認證機制對於保護使用者身分的安全性尚有不足之處，如使用者並無法驗證註冊伺服器的合法性以及使用者的認證金鑰是以明文被存放在認證資料庫中，而現今使用者私密資料遭離職員工盜取後仿冒合法使用者身分的犯罪事件層出不窮，故如何增強會議初始協定認證機制的安全性更受到重視。

本篇論文將會敘述到目前為止各專家學者所提出的認證方法，然後說明各認證方法安全性弱點與可能遭受到的攻擊，隨後我們提出基於非對稱式密碼演算法的改善方法，主要目的是為了避免存放於資料庫內的使用者個人認證資訊受到竊取之後，使用者身分可能會被仿冒甚或被使用於不法用途。

***關鍵詞*：非對稱式密碼學、會議初始協定、網路電話、數位簽章、竊取驗證碼攻擊**

Title of Thesis：Enhancing the Security of SIP Credentials in a VoIP System with Asymmetric Cryptography

Name of Institute：Department of Computer Science and Information Engineering, Collage of Science and Technology, National Chi Nan University

Pages：59

Graduation Time：99/06 Degree Conferred：Master

Student Name：Shin-Fu Huang Advisor Name：Dr. Quincy Wu

Abstract

In recent years, due to the widely deployed network infrastructure, more and more services of the multimedia application are developed - one of the killer applications is Internet Telephony, which is also known as Voice over Internet Protocol (VoIP). Among many VoIP protocols, Session Initiation Protocol (SIP) is adopted by many companies to develop their products, because of its excellence in extension and ease of development. However the user authentication mechanism specified by SIP is still insecure to protect user identity. For example, users can not verify the legitimacy of the Registrar Server. Moreover, the user authentication key is stored in the database of the Registrar in plaintext form. Thus, the crime of identity theft occurs from time to time. Therefore, how to enhance the security of SIP authentication mechanism naturally gets more and more attention.

This thesis summarizes enhanced SIP authentication schemes proposed by many experts and scholars, and illustrates the security vulnerabilities and attacks to which those schemes may be subjected. Then, we propose an improved authentication scheme that is based on asymmetric key cryptography, which can prevent user identity from being impersonated even if the database of the Registrar is cracked and the user

authentication key is stolen. Finally, implementation results show that the proposed scheme is efficient to be adopted in daily VoIP applications.

Keywords **：Asymmetric Cryptography, Digital Signature, Internet Telephony, Session Initiation Protocol, Stolen-Verifier Attack**

目錄

圖目錄

表目錄

第1章 簡介

1.1 相關背景

近年來，由於網路硬體建設的普及，網路上的各種多媒體應用服務如雨後春筍般的誕生，其中之一殺手級的應用即是網路電話(Voice over Internet Protocol，簡稱 VoIP)[1]。在眾多的網路電話標準協定中，如 International Telecommunication Union - Telecommunication Standardization(ITU-T)所制定的 H.323 與 Internet Engineering Task Force(IETF)所提出的會議初始協定(Session Initiation Protocol，簡稱 SIP)[2][3]較為受到矚目。其中 SIP 因其功能擴充性較其它的網路電話標準協定佳、開發易，所以越來越多廠商用以發展產品。

SIP 是一種基於 TCP/IP 協定上層運作的信令控制協定(Signaling Protocol)，被用以建立、修改或終止一個或多個參與者之間的通話。在 SIP 協定系統架構中的基本元件可分為兩大類：SIP 使用者代理端(User Agent，簡稱 UA)和 SIP 伺服器端，其中伺服器端則又可以包括代理伺服器(Proxy Server)，重定向伺服器(Redirect Server)與註冊伺服器(Registrar)。表 1-1 是 SIP 標準文件所定義的系統基本元件表。

表 1-1 SIP 基本元件定義表

SIP 元件名稱	**行為定義**
使用者代理	SIP 協定中的用戶端設備，可以是個人電腦上的 SIP 客戶端軟體或者是獨立具有實體網路介面的 SIP 話機。
代理伺服器	SIP 協定的運作中心，負責代表 UA 或其他的代理伺服器產生的請求或將所收到的請求代為轉送到另一個目標 SIP 元件。
重定向伺服器	接受任何 SIP 元件的請求，回應被呼叫方的 SIP 位置所對應的一個或多個位址，並且將它傳給呼叫端。

註冊伺服器	接受 REGISTER 請求的伺服器，提供 UA 進行註冊的介面，用以管理以及特定的服務。通常 UA 一開機就必須對註冊伺服器進行使用者登錄，因此註冊伺服器還兼具有使用身分認證的功能。註冊完畢之後，該 UA 的資訊才得以被查詢及被呼叫建立通話。

SIP 定義了六種基本的方法，分別是 REGISTER、INVITE、ACK、BYE、CANCEL 與 OPTIONS；各方法的行為定義說明如表 1-2 所示。SIP 協定的建立主要是借用了兩個觀念：網頁瀏覽(Hypertext Transfer Protocol，簡稱 HTTP)[4]與電子郵件(Simple Mail Transfer Protocol，簡稱 SMTP)[5]。除了直接採用與 HTTP/1.1 相同的訊息編碼方式外，SIP 與 HTTP 及 SMTP 一樣都是使用明文方式來傳遞訊息。也就是說當使用者收到一個 SIP 訊息封包時，使用者不需要經由其他的解碼方式即可從封包中瞭解到 SIP 相關資訊。

表 1-2　SIP 基本方法

請求方法	**說明**
REGISTER	使用者代理端向伺服端註冊其所在網際網路協定地址資訊。
INVITE	啟動會話呼叫程序，或改變先前呼叫的參數(re-INVITE)。
ACK	被呼叫端確認 INVITE 的回覆。
BYE	結束此次會話。
CANCEL	呼叫端取消目前建立通話的程序。
OPTIONS	查詢被呼叫端的 SIP 協定相關支援能力。

1.2 研究目的

在以 SIP 協定為基礎的網路電話系統中，使用者的身分認證機制被應用在兩種情形下，第一種情形是當使用者向註冊伺服器(Registrar)進行註冊(REGISTER)請求時，註冊伺服器會發起挑戰給予使用者，然後使用者會針對該挑戰訊息給予回應，註冊伺服器收到此回應訊息之後就得以使用該回應訊息中的認證資訊來驗證使用者身分的合法性；第二種情形是當呼叫端(User Agent Client，簡稱 UAC)欲透過代理伺服器與被呼叫端(User Agent Server，簡稱 UAS)建立通話連線之前，代理伺服器端會先行請 UAC 進行身分認證的挑戰，待挑戰成功之後代理伺服器端才會代 UAC 向 UAS 發送建立通話請求。此兩種情形在實際的 SIP 系統運作中是採用相同的身分認證演算法。

目前在 SIP 標準文件[3]中是採用 HTTP Digest 帳號認證機制[6]來做為 SIP 身分認證機制。此方法運用單向雜湊函數等低計算複雜度的計算來保護使用者身分認證訊息的安全，雙方使用一把共享的認證金鑰（也就是使用者於事前所設定的密碼）來進行身分的認證。此認證演算法可以提供伺服器端驗證使用者身分的正確性，但是使用者並無法驗證伺服器的合法性，所以有可能會遭受到伺服器偽裝攻擊(Server Spoofing Attack)與離線式密碼猜測攻擊(Off-Line Password Guessing Attack)，我們在表 1-3 中定義各攻擊的行為。此外在 SIP 實際運作的系統中，使用者的認證帳號與密碼皆是以明文的方式被保存在伺服器端的資料庫內，這樣一來便有可能會遭受到竊取驗證碼攻擊(Stolen-Verifier Attack)。一旦使用者認證資料庫被非法人士入侵，或者是擁有權限的管理人員離職前監守自盜，都可以在取得使用者帳號密碼後，輕易地冒充其身分撥打長途電話，照成帳務的錯亂與使用者的困擾；若是用於恐嚇與詐欺電話，更可能造成無辜的使用者變成犯罪嫌疑人。由於管理人員監守自盜的事件時有所聞[7]，所以本篇論文的重點即是改善 SIP 的認

證方式，將原本使用對稱式認證金鑰的認證方式改變成使用非對稱式認證金鑰系統，以維護使用者帳號的安全性。

表 1-3　常見攻擊類型一覽表

代號.中文名稱 (英文名稱)	**行為模式**
A1.重播攻擊 (Replay Attack)	指攻擊者將從網路上截取的某些通訊內容(如認證資訊)重新發送，以欺騙伺服器的認證機制。
A2.離線式密碼猜測攻擊 I (Off-Line Password Guessing Attack I)	攻擊者利用持續觀察並記錄使用者與遠端伺服器認證過程的通訊內容，且 利用此通訊內容蒐集其有用的資訊。攻擊者 利用這些通訊內容來驗證其所猜測通行碼的正確性，以確認是否成功猜測到通行碼。
A3.伺服器偽裝攻擊 (Server Spoofing Attack)	指攻擊者假冒伺服器以騙取使用者資訊。
A4.使用者假冒攻擊 (Impersonation Attack)	指攻擊者假冒成其他的合法使用者入侵系統使用服務。
A5.竊取驗證碼攻擊 (Stolen-Verifier Attack)	指當攻擊者取得使用者存放在認證伺服器上的認證金鑰之後，可以直接使用該認證金鑰登入使用者帳號使用服務。
A6.離線式密碼猜測攻擊 II (Off-Line Password Guessing Attack II)	攻擊者竊取到儲存在遠端伺服器的認證表，即使認證表內的認證資訊為通行碼經過單向雜湊函數運算後的雜湊值，但其安全 度仍然不足。因為攻擊者依然可以反覆地去猜測通行碼，並經此單向雜湊函數運算後，再去判斷與認證表內的雜湊值是否相等， 若相等

就表示攻擊者所猜測的通行碼是正確的。

在表 1-3 中我們定義 A1~A6 六種攻擊類型，其中我們特別將離線式密碼猜測攻擊詳細定義出兩種類型，分別是 A2.離線式密碼猜測攻擊 I 與 A6.離線式密碼猜測攻擊 II。A2.離線式密碼猜測攻擊 I 是中間者竊聽一合法使用者與註冊伺服器進行認證時的訊息內容，並將訊息內容保存起來，然後中間者依照所得到的認證資訊求得該合法使用者的私密金鑰(一般而言，就是使用者所設定的密碼)，如圖 1-1 所示；A6.離線式密碼猜測攻擊 II 是不法人士透過入侵註冊伺服器或是管理人員監守自盜的方式以取得使用者的認證金鑰，然後利用該認證金鑰求得使用者的私密金鑰，如圖 1-2 所示。

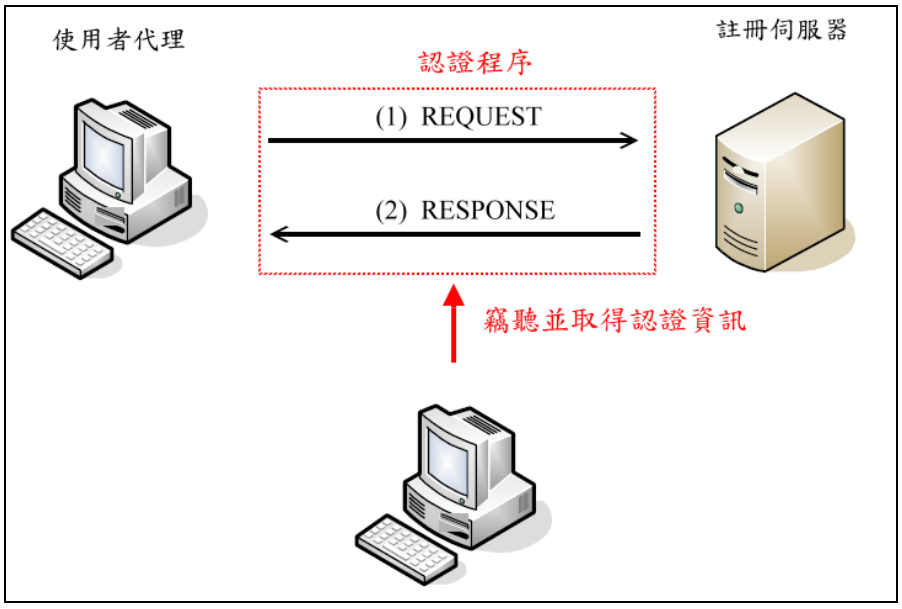

圖 1-1　離線式密碼猜測攻擊 I

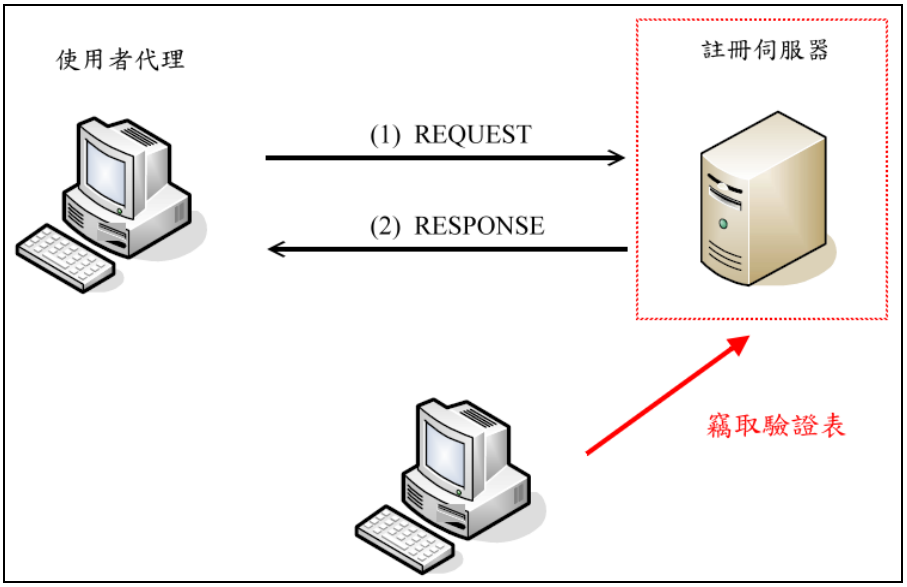

圖 1-2　離線式密碼猜測攻擊 II

1.3 論文架構

本篇論文總共分為六個章節：第 1 章先簡介會議初始協定所定義的系統基本元件與請求方法，然後說明此篇論文的研究目的與論文架構。第 2 章說明各專家學者對於此研究主題所提出的改善機制，並且逐一分析其認證方法安全性尚不足之處。第 3 章說明我們對於此研究主題依據不同的訴求，所提出兩種不同的改善方法。第 4 章說明如何實作系統以驗證可行性，並且對完成認證程序所需花費的時間進行量測。第 5 章將各專家學者所提出的方法與我們所提出的機制進行安全性的比較。最後於第 6 章提出我們的結論。

第2章 相關工作

在第 3 章提出我們的方法之前，我們先於本章說明各專家學者為了增強 SIP 協定認證機制安全性所提出的改善方法。本章 2.1 節描述現行 SIP 基於 HTTP 摘要認證演算法的認證機制[3][6]；2.2 節說明學者 Yang 等人[8]提出基於金鑰交換概念[9]的認證方法；隨後於 2.3 節說明學者 Huang 等人[10]於 2006 年提出僅使用位元邏輯運算等低計算複雜度的有效率認證方法；2.4 節是學者 Guillet 等人[11]於 2007 年提出在不需大幅修正現行以 HTTP 摘要認證為基礎之 SIP 認證機制的條件下，也可以達成雙向認證功能(Mutual Authentication)的認證方法；2.5 節說明學者 Tsai[12]於 2009 年所提出使用位元邏輯運算與單向雜湊函數等低計算複雜度的認證方法，同一年，學者 Lee[13]指出 Tsai 所提的認證方法於安全性上尚有疑慮之處。表 2-1 為本章節共同使用的符號定義。

表 2-1 本章節共同使用符號說明

符號	符號定義
username	代表欲被認證的帳號名稱。
realm	代表該認證伺服器所屬的領域。
PW	UA 與 Registrar 事先共享的通行密碼。
H(・)	單向雜湊函數。
⊕	Exclusive OR(XOR)，位元邏輯運算。

2.1 以HTTP摘要認證為基礎之SIP認證機制

在 SIP 的標準文件(RFC 3261 Section 22) [3]中說明其協定提供一個以 HTTP 認證演算法為基礎的 SIP 認證機制，此認證機制是採用 Challenge/Response 概念來進行使用者的身份驗證，此認證機制可達成提供訊息鑑定以及預防重播攻擊(Replay Attack)兩種安全性，並且其認證演算法在進行使用者身分認證的過程中可以無需記錄使用者的認證狀態(stateless)。而在 HTTP 認證機制的標準文件[14]內定義兩個認證方法，分別是基本認證(Basic Authentication Scheme)與摘要認證(Digest Access Authentication Scheme)。其中因為基本認證法會將所有認證的資訊(如使用者名稱、密碼)以 Base64[15]編碼，然後僅依此單薄的保護便於公開網路上傳輸；一旦此認證訊息被中間者所截取，中間者便可以輕而易舉地依 Base64 的規則[15]解出使用者的密碼，安全性堪憂。而摘要認證法則是運用單向雜湊函數(如 MD5[16])來打亂認證資訊字串，其具備無法復原的特性，故摘要認證法是被推薦使用的。我們將在此節介紹現行以 HTTP 摘要認證演算法為基礎的 SIP 認證機制。

以 HTTP 摘要認證為基礎的 SIP 認證機制可分為三個階段。1) UA 向 Registrar 發送 REGISTER 請求； 2) 當 Registrar 收到 UA 的請求訊息之後會隨機產生具備時效性的亂數字串為 nonce，然後將此字串回應給 UA，請 UA 進行身分認證的挑戰，此為認證挑戰訊息； 3) UA 收到 Registrar 的挑戰訊息之後，會根據訊息內提供的認證資訊進行認證公式的計算，再將計算後的結果回應給 Registrar，此為挑戰回覆訊息。表 2-2 為 SIP 認證演算法符號說明。圖 2-1 為 SIP 認證演算法流程圖。

表 2-2 SIP 認證演算法符號說明

符號	符號定義
nonce	由 Registrar 隨機所產生且具備時效性的亂數值。

method	SIP 請求訊息的行為。(例如 "REGISTER")
Request-URI	代表被呼叫端所在的網路位址。(例如 "sip:163.22.21.1")
:	進行字串連結(concatenate)之運算元。

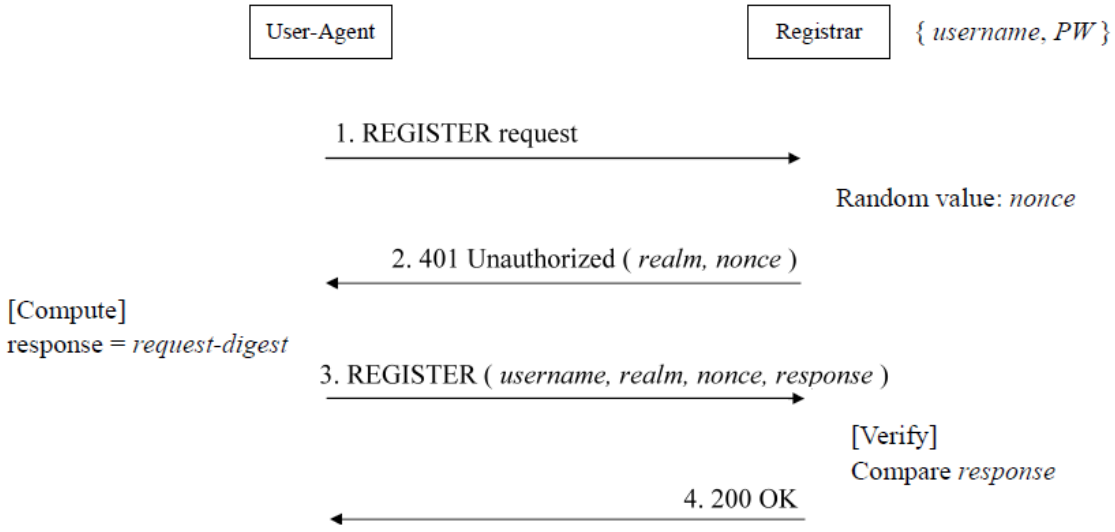

圖 2-1　SIP 認證演算法流程圖

我們假設 UA 和 Registrar 事先皆已共享同一把金鑰(也就是使用者密碼)，所以在 Registrar 端的認證資料庫裡面已具有使用者的名稱與通行密碼的資訊。在此假設條件之下，我們簡單說明基於 HTTP 摘要認證法的 SIP 認證機制是如何運作的。

Step 1. UA 向 Registrar 發送 SIP 方法為 REGISTER 的註冊請求訊息。

Step 2. 當 Registrar 收到註冊請求之後，隨機產生一具備時效性的亂數字串 nonce，接著 Registrar 會將具有{ realm, nonce }兩認證參數的挑戰訊息傳送給 UA，請 UA 依照挑戰訊息進行回應。

實例：

Registrar 挑戰訊息中所提供的認證資訊：

WWW-Authenticate: Digest

realm="163.22.21.1",

nonce="4bc5256f0000002ab12dcabf43548b47b1d8ef1316f7b1b7"

Step 3. UA 收到來自 Registrar 的挑戰訊息之後，會根據訊息所提供的資訊與使用者所輸入的密碼計算出 request-digest 值，也就是 response 值，計算公式如圖 2-2 所示。最後，再將{ username, realm, nonce, response }傳送給 Registrar 進行認證。

A1 = username ":" realm ":" PW	*(1)*
A2 = method ":" Request-URI	*(2)*
response = request-digest = H(H(A1) ":" nonce ":" H(A2))	*(3)*

圖 2-2　response 欄位計算公式

實例：

(1) 已知參數值如下：

username = 1005

realm = 163.22.21.1

PW = 1005

method = REGISTER

Request-URI = sip:xinfu.ncnu.edu.tw

nonce = 4bc5256f0000002ab12dcabf43548b47b1d8ef1316f7b1b7

H(・) = MD5

(2) 帶入認證公式中，計算出 response 欄位：

A1 = 1005:163.22.21.1:1005

H(A1) = 5a22ef09ca60ace23420e62666035910

A2 = REGISTER:sip:xinfu.ncnu.edu.tw

H(A2) = d2e4f80cce3b5236bbba5d6a7b12ecb0

response = H(H(A1): nonce: H(A2))

= 548d6f1277a915ccbb1b299c815404e9

(3) UA 將計算出的認證資訊，回應給 Registrar，進行認證挑戰：

Authorization: Digest

username="1005",

realm="163.22.21.1",

nonce="4bc5256f0000002ab12dcabf43548b47b1d8ef1316f7b1b7",

uri="sip:xinfu.ncnu.edu.tw",

response="548d6f1277a915ccbb1b299c815404e9",

algorithm=MD5

Step 4. 當 Registrar 收到 UA 的挑戰回覆訊息之後，會先檢查 nonce 值是否合法，如果該值是合法的，會再根據回覆訊息中所提供的 username 從使用者資料庫中取得雙方事先所共享的密碼，然後根據圖 2-2 公式所示，計算出自己的 response 值，再把此值與從 UA 回覆訊息中取得的 response 值做字串的比較。如果相同，則代表認證成功，會回傳 200 OK 的 SIP 訊息給 UA。只要上述任何一個驗證的環節不成立的話，那麼 Registrar 就會回到 Step 2，重新產生 nonce 值並請 UA 再進行挑戰。

經過上述認證流程的說明，我們可以發現目前 SIP 認證演算法是使用字串連結與單向雜湊函數等低計算複雜度的計算來達成身分認證的目的，這是一個很有效率並且容易實作出系統的演算法。一般而言，SIP 訊息是以明文模式傳輸，也就

是說在認證過程中，認證資訊皆可能被攻擊者所截取收集，此時攻擊者就可以根據取得的資訊進行離線式密碼猜測攻擊 I 猜出使用者的 PW，之後就可以仿冒使用者帳號登入；並且此認證方法只提供 Registrar 對 UA 做認證，但是 UA 無法認證 Registrar 的合法性，故 UA 容易遭受到伺服器冒充攻擊；最後，此認證方法也容易遭受到竊取驗證碼攻擊，因為雙方事先所共享的密碼在實際運作的環境中是以明文模式被保存在 Registrar 的認證資料庫中，所以一旦資料庫遭受到攻擊導致使用者密碼被竊取，將可能被不法人士直接使用該帳號密碼冒名登入。

2.2 Yang等人的認證方法

學者 Yang 等人因為 SIP 標準所定義的認證機制在安全性上有明顯的不足，所以在 2005 年時，共同提出一個以 Diffie-Hellman 金鑰交換概念為基礎的認證機制，此認證機制是利用解離散對數(Discrete Logarithm Problem，簡稱 DLP)的難題來保障認證資料在交換時的安全性。

學者 Yang 等人的認證機制可以分為三個部分。 1) UA 會傳送夾帶著欲認證的請求給 Registrar； 2) 在 Registrar 收到 UA 的訊息之後，會計算證明其合法性的認證資訊，並且將此認證資訊與給 UA 的認證挑戰一起傳送給 UA； 3) UA 收到 Registrar 的認證挑戰訊息之後，會先使用訊息所提供的資訊檢驗 Registrar 的合法性，待驗證成功之後，UA 才會接受 Registrar 的挑戰，提供挑戰回覆訊息給 Registrar，讓 Registrar 得以驗證 UA 的合法性。表 2-3 為 Yang 等人的認證方法符號說明。圖 2-3 為 Yang 等人的認證方法流程圖。

表 2-3　Yang 等人的認證方法符號說明

符號	符號定義
g	產生大數值的生成器。
p	一個夠大且安全的質數。
r1	由 UA 隨機產生的亂數值。
r2	由 Registrar 隨機產生的亂數值。

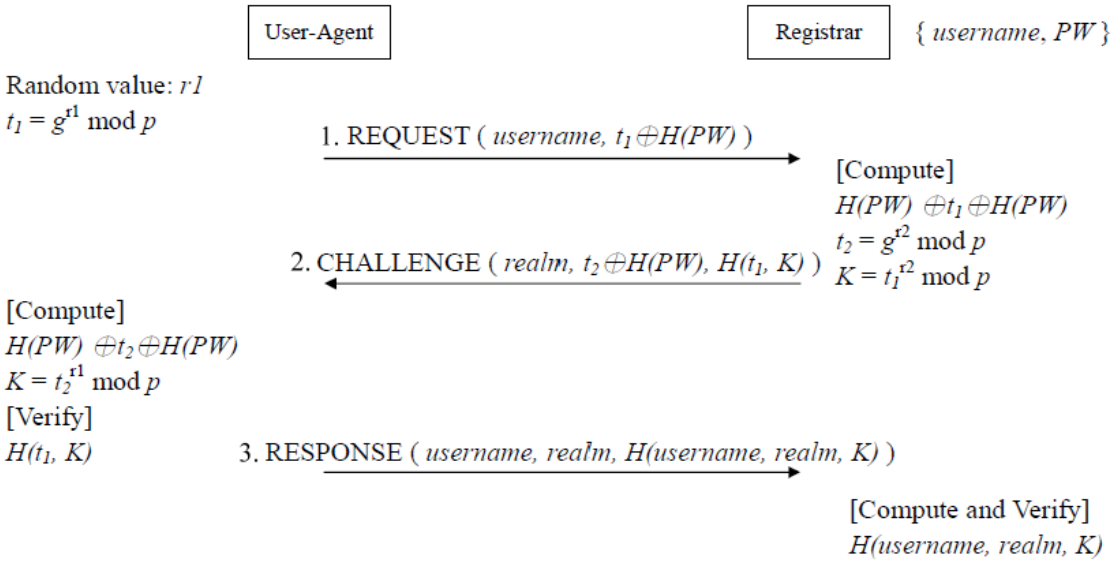

圖 2-3　Yang 等人的認證方法流程圖

我們假設 UA 和 Registrar 事先皆已共享同一把金鑰(也就是使用者密碼)，所以在 Registrar 端的認證資料庫裡面已具有使用者的名稱與通行密碼的資訊，並且 UA 與 Registrar 皆知道一個大的質數為 p 與大數產生器為 g。在此假設條件之下，我們簡單說明此認證方法的運作流程。

Step 1. UA 隨機取得一亂數值 r1，計算出 $t_1 = g^{r1} \bmod p$，並使用 H(PW)保護 t_1 為 $t_1 \oplus H(PW)$，然後送出夾帶{ username, $t_1 \oplus H(PW)$ }的註冊請求訊息給 Registrar。

Step 2. 當 Registrar 收到 UA 的註冊請求訊息之後，會先取得訊息中的 username 欄位，然後到使用者資料庫查詢該使用者的密碼，計算出 H(PW)，再將 H(PW)與 $t_1 \oplus H(PW)$做 XOR 的邏輯運算，即可觀察出 t_1 值。接著 Registrar 會隨機取得一亂數值 r2，分別計算 $t_2 = g^{r2} \bmod p$, $t_2 \oplus H(PW)$, $K = t_1^{r2} \bmod p$, $H(t_1, K)$。最後，Registrar 會送出夾帶{ realm, $t_2 \oplus H(PW)$, $H(t_1, K)$ }的認證挑戰訊息給 UA。

Step 3. UA 收到來自 Registrar 的認證挑戰訊息之後，使用已知的 H(PW)解出 t_2 為 $H(PW) \oplus t_2 \oplus H(PW)$，接著利用 t_2 算出自己的 K 值為 $K = t_2^{r1} \bmod p$，再將計算出來

的 K 值與 t_1 合併雜湊得到 H(t_1, K)，並且將此值與從 Registrar 端得到的 H(t_1, K)做比較，如果兩者是相同的，則代表 Registrar 是合法的。最後，UA 會合併 username, realm 與 K 值雜湊計算得到 H(username, realm, K)，再將夾帶著{ username, realm, H(username, realm, K) }的挑戰回覆訊息給 Registrar。

Step 4. 當 Registrar 收到 UA 的挑戰回覆訊息之後，會利用在 Step 2 時所計算出來的 K 值合併 username 與 realm，計算出自己的 H(username, realm, K)，再將計算結果與 UA 回覆訊息內的 H(username, realm, K)做比較，如果兩個值相同，則代表 UA 的身分被認證成功。

學者 Yang 等人的認證方法確實可以解決 HTTP 摘要認證機制所容易遭遇的安全性攻擊，如離線式密碼猜測攻擊 I 與伺服器冒充攻擊，但是它所需要的計算量相當龐大，也需要大幅度的更動現有的認證機制，並且也沒有考慮到使用者資料庫有可能遭受到的竊取驗證碼攻擊。在越來越重視資訊安全的現在，如何預防竊取驗證碼攻擊也是設計一個身分認證演算法不可不考慮的重點。

2.3 Huang等人的認證方法

學者 Huang 等人認為 Yang 等人所提出基於 Diffie-Hellman 金鑰交換概念的認證機制計算成本太高，對於運算能力較低的設備如智慧卡(Smart Cards)或是行動裝置(Mobile Devices)而言，不是一個很適當的解決方案。於是學者 Huang 等人在 2006 年提出一個僅需要單向雜湊函數與邏輯位元運算。就可以滿足基本安全性需求且又能同時具備高效率的認證方法。

學者 Huang 等人所提出的有效率認證機制分為三個部分。 1) UA 傳送註冊請求訊息給 Registrar； 2) Registrar 收到 UA 的請求訊息之後，會先計算證明自己身分合法性的認證訊息，同時將認證訊息與檢驗 UA 身分合法性的挑戰訊息一起傳送給 UA； 3) UA 接收 Registrar 的挑戰訊息後，會先檢驗 Registrar 的合法性，然後才計算確認自己合法性的認證資訊回覆給 Registrar，提供 Registrar 對 UA 端進行身分確認。表 2-4 為 Huang 等人的認證方法符號說明。圖 2-4 為 Huang 等人的認證方法流程圖。

表 2-4　Huang 等人的認證方法符號說明

符號	符號定義
a_1	由 UA 隨機所產生的亂數值。
a_2	由 Registrar 隨機所產生的亂數值。

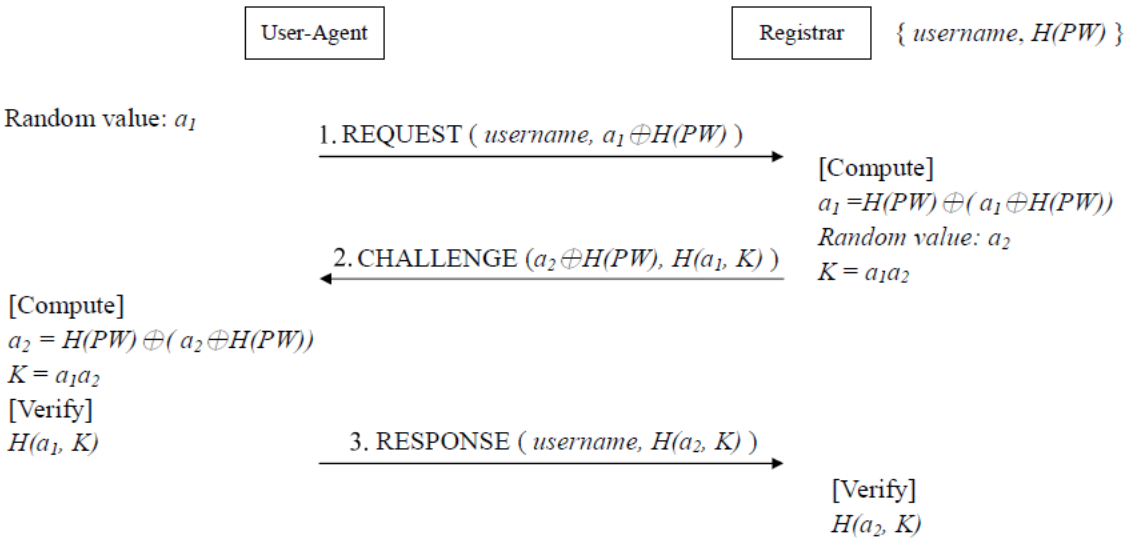

圖 2-4　Huang 等人的認證方法流程圖

我們假設 UA 和 Registrar 事先皆已共享同一把金鑰(也就是使用者密碼)，但是在 Registrar 端的認證資料庫裡面所儲存的是使用者的帳號名稱與被雜湊過後的使用者密碼為 H(PW)。在此假設條件之下，我們將簡單說明 Huang 等人認證方法是如何運作的。

Step 1. UA 隨機取得一亂數值為 a_1，並且與 H(PW)計算得到 $a_1 \oplus H(PW)$，再將具有{ username, $a_1 \oplus H(PW)$ }兩認證參數的註冊請求訊息給 Registrar。

Step 2. Registrar 收到來自 UA 的請求訊息之後，會利用訊息所提供的 username 查詢使用者資料庫以取得該認證帳號的 H(PW)值，並利用該值觀察得到 a_1 為 $H(PW) \oplus (a_1 \oplus H(PW))$。接著 Registrar 會隨機產生一亂數為 a_2，並利用 H(PW)保護 a_2 為 $a_2 \oplus H(PW)$，再計算 K 值為 $K = a_1 a_2$，將 K 值與 a_1 合併雜湊為 $H(a_1, K)$，傳送具{ $a_2 \oplus H(PW)$, $H(a_1, K)$ }兩認證參數的認證挑戰訊息給 UA。

Step 3. UA 收到 Registrar 的認證挑戰訊息之後，使用 H(PW)去觀察得到 a_2 值為 $H(PW) \oplus (a_2 \oplus H(PW))$，接著利用 a_2 值計算得到自己 K 值為 $K = a_1 a_2$，利用 K 值與 a_1 雜湊為 $H(a_1, K)$，將 UA 自己計算得到的 $H(a_1, K)$與來自 Registrar 認證挑戰訊息

所提供的 H(a_1, K)做字串的比較。如果相同，則代表 Registrar 是合法的。最後，UA 會將 K 值與 a_2 合併雜湊為 H(a_2, K)，傳送具有{ username, H(a_2, K) }認證資訊的挑戰回覆訊息給 Registrar。

Step 4. Registrar 收到 UA 的挑戰回覆訊息之後，會利用自己於 Step 2 所算出的 K 值與 a_2 雜湊為 H(a_2, K)，將自己所計算出的 H(a_2, K)與 UA 挑戰回覆訊息中所提供的 H(a_2, K)做字串的比較。如果相同，則代表 UA 是合法的，開始提供 UA 服務；如果不同，便會拒絕 UA 任何的請求。

由學者 Huang 等人所提出的認證機制，因為在認證的計算公式中僅使用位元運算與單向雜湊函數，所以在計算上可以很有效率，並且也改善部分現行 SIP 認證機制安全性的問題。但是在此認證方法中 Registrar 需要於認證過程內暫時性的保存認證資訊，如 Step 2 中 K 值的暫時保留，所以具有阻斷式服務攻擊(Denial of Service)的隱憂。對於離線式猜測攻擊 I 也未能有效預防，做法如下：假設攻擊者於訊息傳送中間分別取得 Step 1 與 Step 2 的認證資訊{ username, a_1⊕H(PW) }、{ a_2⊕H(PW), H(a_1, K) }，再自行假設使用者密碼為 PW'，計算出 H(PW')，再利用 H(PW')分別計算出 a_1'為 H(PW')⊕(a_1⊕H(PW))、a_2'為 H(PW')⊕(a_2⊕H(PW))，那麼我們就可以得到 K'= a_1'a_2'， 有了 K'我們就可以計算出 H(a1', K')，再將 H(a_1', K')與 Step 2 取得的 H(a_1, K)做比較。如果 H(a_1', K')與 H(a_1, K)相同，就可以得知 PW'=PW。

2.4 Guillet等人的認證方法

學者 Guillet 等人認為在 SIP 標準中所定義基於 HTTP Digest 的 SIP 認證機制僅允許 Registrar 驗證 UA，而 UA 卻無法驗證 Registrar，這樣一來容易遭受到伺服器冒充攻擊；為了避免遭受此攻擊，認證機制就需要提供雙向認證的功能。在目前以 HTTP Digest 為基礎的 SIP 認證機制中若要具備有雙向認證的功能，則需要仰賴傳輸層安全協定(Transport Layer Security Protocol，以下簡稱 TLS) [17]，並且 SIP 協定所使用的傳輸層協定必需支援傳輸控制協定(Transmission Control Protocol，以下簡稱 TCP)。然而 SIP over TLS 需要付出很高的計算成本，主要的原因是 SIP over TLS 會將整個 SIP 訊息做加解密的保護，所以學者 Guillet 等人在 2008 年提出一個不必大幅度更動現行的 SIP 認證機制，並且一樣僅需使用低計算複雜度的運算就可支援雙向認證功能，提高使用者身分認證的安全性。

Guillet 等人所提出的認證方法一樣可分為三個階段。 1) UA 會發送請求訊息給 Registrar； 2) 當 Registrar 收到 UA 的請求訊息之後，會根據請求訊息內所提供的資訊產生新的字串，然後將此字串回應給 UA，請 UA 進行挑戰，此為認證挑戰訊息； 3) UA 收到 Registrar 的認證挑戰訊息之後，會根據此訊息中的認證資訊先驗證 Registrar 的合法性之後，才進行自己身分認證的計算。接著再將計算後的結果回覆給 Registrar，此為認證回覆訊息。表 2-5 為 Guillet 等人的認證方法符號說明。圖 2-5 為 Guillet 等人的認證方法流程圖。

表 2-5　Guillet 等人的認證方法符號說明

符號	符號定義
callid-value	每個 SIP 請求訊息皆會隨機亂數產生的參數值。
nonce	根據 callid-value, realm, username, PW 合併所計算出來的數值。

method	SIP 請求訊息的行為。(例如 "REGISTER")
Request-URI	代表伺服器所在的 IP address。(例如 "sip:163.22.4.78")

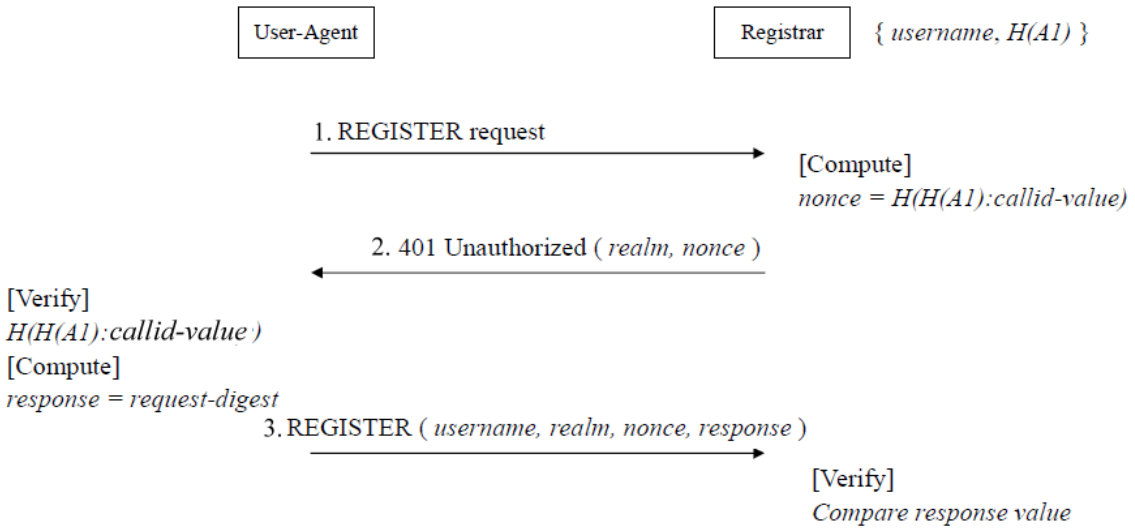

圖 2-5　Guillet 等人的認證方法流程圖

我們假設 UA 和 Registrar 事先皆已共享同一把金鑰(也就是使用者密碼)，但是在 Registrar 端的認證資料庫裡所存放的不是使用者原本所設定的密碼，而是存放使用者密碼與其他參數值合併雜湊之後的結果為 H(A1)，如圖 2-2 公式(1)所示。在此假設條件之下，我們簡單說明 Guillet 等人認證方法是如何運作的。

Step 1. UA 發送註冊請求訊息給 Registrar。

Step 2. 當 Registrar 收到註冊請求訊息之後，先利用 username 查詢該認證帳號存放於認證資料庫中的 H(A1)值，再從註冊請求訊息中取出具有亂數性質的參數如 call-id 值，合併計算 H(H(A1) : callid-value)為 nonce 值。再將具有{ realm, nonce } 兩認證參數的認證挑戰訊息傳送給 UA，請 UA 進行挑戰。

Step 3. UA 收到 Registrar 的認證挑戰訊息之後，會將使用者所輸入的密碼、realm 與 username 合併做雜湊為 H(A1)，如圖 2-2 公式(1)所示。再將於 Step 1 時所發出

註冊請求訊息中夾帶的 callid-value 參數值與 H(A1)合併雜湊 H(H(A1) : callid-value) 為 nonce，最後將自行計算出來的 nonce 與來自 Registrar 訊息內的 nonce 做字串的比較。如果相同，代表 Registrar 驗證成功。接著，UA 會利用 nonce 與其他的認證參數計算出 request-digest 值，也就是 response 值，計算公式如圖 2-2 所示。最後，再將具有{ username, realm, nonce, response }認證資訊的挑戰回覆訊息傳送給 Registrar 進行認證。

Step 4. 當 Registrar 收到 UA 的挑戰回覆訊息之後，會根據圖 2-2 所示，計算出自己的 response 值，再與從 UA 訊息中取得的 response 值做字串的比較。如果相同，則代表認證成功，Registrar 會回傳 200 OK 的 SIP 訊息通知 UA 認證成功。只要上述任何一個驗證的環節是不成立的話，那麼 Registrar 就會拒絕任何來自 UA 的資源請求。

此機制利用 SIP 某些欄位參數具備亂數產生的特性，使得原本基於 HTTP Digest 的 SIP 認證機制可以很容易就支援雙向認證的功能。同時為了預防竊取驗證碼攻擊，在 Registrar 端的認證資料庫內不直接存放使用者的密碼，而是改存放 H(A1) 值。雖然如此，此方法仍會受到離線式密碼猜測攻擊 I 與重播攻擊。

離線式密碼猜測攻擊，攻擊者僅需竊聽 Step 2 的認證訊息{ realm, nonce }，nonce 的產生是 H(H(A1) ":" callid-value)，也就是 H(H(username ":" realm ":" PW) ":" callid-value)。由於 username、relam 與 callid-value 皆為已知，我們僅需要假設使用者密碼為 PW'，便可求得 nonce' = H(H(username:realm:PW') : callid-value)，當 nonce' = nonce 的話，我們就可以證明 PW'就是使用者的密碼。

至於重播攻擊部分，攻擊者僅需要收集 Step 1 與 Step 3 的 SIP 註冊請求訊息，就可以直接使用這兩個 SIP 註冊請求訊息向 Registrar 進行註冊，而攻擊者不需要真正知道使用者的密碼就可以成功冒充使用者的身分撥打網路電話。

2.5 學者Tsai的認證方法

學者 Tsai 認為 2005 年 Yang 等人所提出基於 Diffie-Hellman 金鑰交換概念的認證機制所花費的計算成本太高，所以在 2008 年提出一個以隨機亂數為基礎的有效率認證方法。此方法僅使用單向雜湊函數與位元運算等低計算複雜度的運算，又可以保有 Yang 等人所宣稱可提供的安全性。

Tsai 的認證方法可分為三個階段。 1) UA 發送註冊請求訊息給 Registrar； 2) 當 Registrar 收到 UA 的請求訊息之後，會接受 UA 的認證挑戰並且回覆，同時也請 UA 進行身分認證挑戰，此為認證挑戰訊息； 3) UA 收到 Registrar 的認證挑戰訊息之後，需要先驗證 Registrar 的合法性，然後才進行自己身分認證的計算，再將計算後的結果回覆給 Registrar，此為挑戰回覆訊息。表 2-6 為 Tsai 的認證方法符號說明。圖 2-6 為 Tsai 的認證方法流程圖。

表 2-6　Tsai 的認證方法符號說明

符號	符號定義
N_C	由 UA 隨機所產生的亂數值。
N_S	由 Registrar 隨機所產生的亂數值。

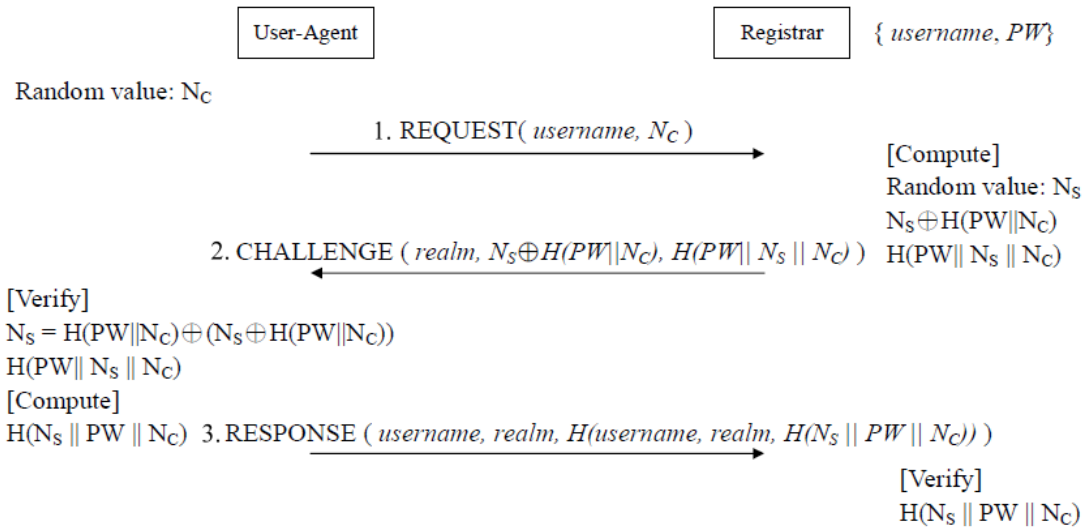

圖 2-6　Tsai 的認證方法流程圖

我們假設 UA 和 Registrar 事先皆已共享同一把金鑰(也就是使用者密碼)，所以在 Registrar 端的認證資料庫裡面已具有使用者的名稱與通行密碼的資訊。在此假設條件之下，我們簡單說明 Tsai 的認證方法是如何運作的。

Step 1. UA 先隨機產生一亂數值為 N_C，然後發送夾帶著{ username, N_C }兩認證參數的註冊請求訊息給 Registrar。

Step 2. Registrar 收到 UA 的註冊請求訊息之後，使用訊息中所提供的 username 查詢該認證帳號之密碼，並隨機產生一亂數值為 N_S，分別計算出 $N_S \oplus H(PW||N_C)$與 $H(PW||N_S||N_C)$，其中 $H(PW||N_S||N_C)$是 Registrar 給 UA 的挑戰回覆，提供 UA 得以此參數驗證 Registrar 的合法性，再將{ realm, $N_S \oplus H(PW||N_C)$, $H(PW||N_S||N_C)$ }認證挑戰訊息傳送給 UA，請 UA 進行挑戰。

Step 3. UA 收到 Registrar 的認證挑戰訊息之後，會使用已知的 PW 與 N_C 計算得到 $H(PW||N_C)$，使用 $H(PW||N_C) \oplus (N_S \oplus H(PW||N_C))$可以觀察得到 N_S，再使用 N_S 計算出 $H(PW||N_S||N_C)$，最後，將自行計算出的 $H(PW||N_S||N_C)$與從 Registrar 訊息中所取得的 $H(PW||N_S||N_C)$做字串比較。如果兩者相同，則代表 Registrar 是合法的。接著，UA 會計算 $H(N_S||PW||N_C)$，再將計算出的結果與其他的認證參數合併雜湊後為

H(username, realm, H(N_S||PW||N_C))，最後 UA 將具有{ username, realm, H(username, realm, H(N_S||PW||N_C)) }認證資訊的挑戰回覆訊息傳送給 Registrar。

Step 4. 當 Registrar 收到 UA 的挑戰回覆訊息之後，使用自己已知的 N_S、PW 與 Step 1 取得的 N_C 計算出自己的 H(N_S||PW||N_C)，然後將自行計算出的 H(N_S|| PW|| N_C)與從 UA 訊息中所取得的 H(N_S||PW||N_C)做字串比較。如果兩者相同，則代表 UA 認證成功。

學者 Tsai 所提出的認證方法主要是藉由雙方各產生不同的亂數來驗證對方，其認證公式僅使用到單向雜湊函數與 XOR 運算，故其計算複雜度較低，也因此在認證運作的效率上可以有較好的表現。不過，因其認證方法是將認證金鑰(也就是使用者的密碼)以明文模式存放在 Registrar 的認證資料庫中，所以有機會遭受到竊取驗證碼攻擊，並且學者 Lee 也指出其認證演算法無法抵擋離線式密碼猜測攻擊。

離線式密碼猜測攻擊 I：攻擊者在截取 Step 1 與 Step 2 的訊息後，可以輕易的得到{ N_C, N_S⊕H(PW||N_C), H(PW||N_S ||N_C) }三個認證資訊；攻擊者假設使用者的密碼為 PW'，利用 PW'與 N_C 計算得到 H(PW'||N_C)，再經由 H(PW'||N_C)與 N_S⊕H(PW||N_C)合併計算可取得 N_S'值。因為 PW'是被假設出來的，所以求得的 N_S'值未必是正確的；但是我們可以利用 N_C, PW'與 N_S'計算得到 H(PW'||N_S'||N_C)，如果 H(PW'|| N_S' ||N_C)值跟竊聽 Step 2 訊息時所取得的 H(PW|| N_S||N_C)值是相等的話，那麼攻擊者就可以得知其所假設的使用者密碼 PW'是正確的。

第3章 基於非對稱式金鑰認證演算法

我們在前一章說明各專家學者對於增強 SIP 認證機制的安全性所提出的各種認證演算法，並且分析其認證方法安全性不足之處，所以我們在本章節提出我們對於增強 SIP 認證機制的認證演算法。首先，我們於 3.1 節提出以使用者密碼為基礎並且具備公開金鑰安全性的認證方法；我們先說明我們提出這個認證方法的主要訴求為何，再描述我們的認證方法是如何運作的，最後則是分析我們所提認證方法的安全性。而在 3.2 節提出另一個以非對稱式密碼來增強使用者身分認證的方法；我們先說明我們提出這個方法最主要改善的目的為何，接著說明我們的方法是如何運作的，最後則是分析我們所提認證方法的安全性。

3.1 以SRP為基礎之SIP認證機制

離線式密碼猜測攻擊(Off-Line Password Guessing Attacks)是目前每個被提出增強 SIP 認證機制都較無法抵擋的安全性攻擊，在這裡我們對於離線式密碼猜測攻擊成功與否的定義是指該認證演算法能否讓該認證金鑰於有限的時間內不被攻擊者所猜出。現行 SIP 認證機制如果要能夠抵擋離線式密碼猜測攻擊，並且也達到相互認證的功能，可以採用的方式就是以認證方法搭配傳輸層安全協定(Transport Layer Security Protocol，簡稱 TLS)運作，即是 SIP over TLS。但是 TLS 機制所需要的加解密計算對於 SIP 協定的終端設備來說不夠輕量，並且 SIP 訊息中並不是每個標頭欄位的資訊皆需要被加密保護。於是我們思考如何能夠僅對認證資訊做保護，並且也不需要像 TLS 需要公開金鑰基礎設施(Public Key Infrastructure，簡稱 PKI)的設置與大量訊息內容加解密的計算，更重要的是可以不需更改使用者的習慣，只要使用自己所設定的密碼即可。

基於上述的需求，我們在此介紹現今已具備實作成果的認證系統－秘密遠端

密碼協定(Secure Remote Password Protocol，簡稱 SRP)[18][19]至 SIP 認證機制中，為 SRP on SIP Authentication[20]，簡稱為 SRPSA。SRP 是一種安全的新型密碼認證和金鑰交換式的通訊協定，由於在認證過程中所使用到的認證金鑰不是原本使用者所設定的密碼明文，所以即使攻擊者竊取存放於 Registrar 認證資料庫中的認證金鑰也無法使用其認證金鑰直接登入冒用使用者的帳號；同時，SRP 協定應用 Diffie-Hellman 金鑰交換的概念，能抵抗離線式密碼猜測攻擊。SRP 協定已經逐漸的被應用在 teletype network (簡稱 Telnet)[21][22]與檔案傳輸協定(File Transfer Protocol，簡稱 FTP)使用者身分認證，並且在每一次認證成功之後都會產生出一把會議鑰匙(Session Key)，用以對後續傳遞訊息做加密保護。以下我們將介紹我們如何將 SRP 協定導入至 SIP 認證機制中。表 3-1 為 SRPSA 的符號說明。圖 3-1 為 SRPSA 認證流程圖。

表 3-1　SRPSA 符號說明

符號	符號定義
username	欲進行認證帳號名稱。
password	使用者所設定的密碼。
realm	註冊伺服器所在領域位置。
g	大數值的生成器。
N	大數值且安全的質數。
H(・)	雜湊函數。
a	由 UA 隨機所產生的亂數值。
b	由 Registrar 隨機所產生的亂數值。
||	字串連結。

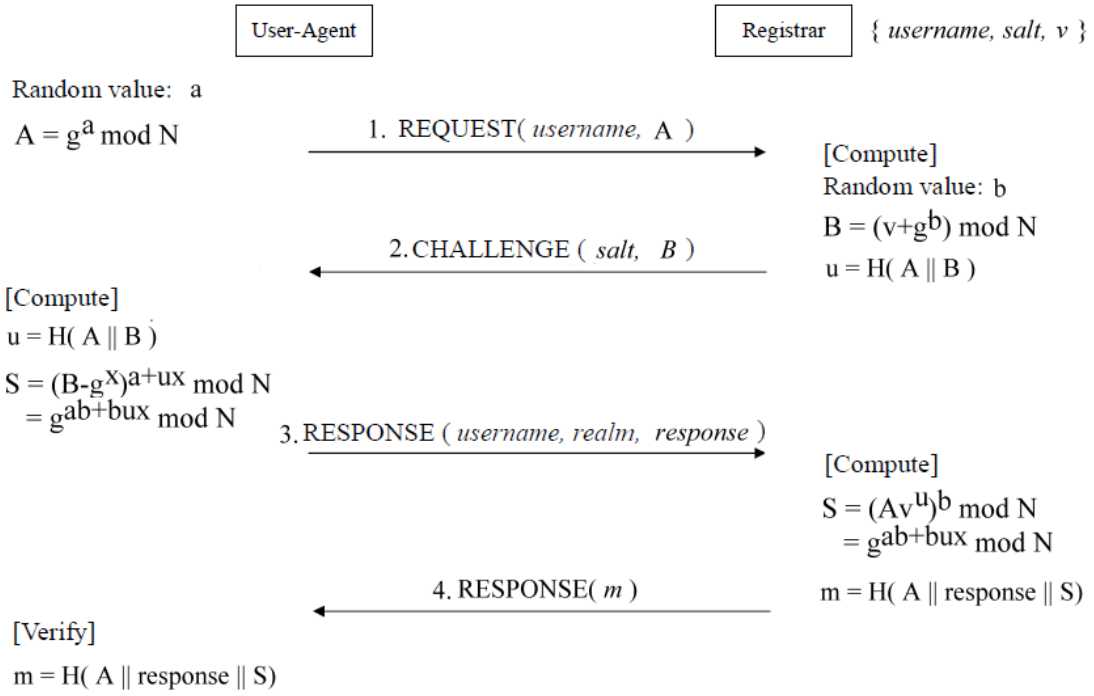

圖 3-1　SRPSA 認證流程圖

(1) 使用者新註冊帳號階段

$$x = H(salt \,||\, H(username \,||\, ":" \,||\, password)) \quad (4)$$

$$v = g^x \bmod N \quad (5)$$

圖 3-2　Password Verifier 計算公式

在我們所提出的 SRPSA 中，Registrar 與 UA 事先共享使用者密碼，但是 Registrar 並不直接採用使用者密碼為認證金鑰，而是將使用者密碼進行轉換得到 Password Verifier，轉換步驟如下：

Step 1. Registrar 會先將使用者帳號(username)與使用者密碼(password)合併雜湊，再將雜湊後的結果與隨機亂數產生的 salt 值進行一次合併雜湊為 x 值，如圖 3-2 公

式(4)。

Step 2. 將 x 值代入 g^x mod N 求得 v 值，如圖 3-2 公式(5)，v 值即為認證金鑰。

Step 3. Registrar 會將{ username, salt, v }三個進行認證程序時所需要用到的資訊存入認證資料庫內。

(2) 使用者帳號登入認證階段

本小節將說明 SRPSA 是如何進行使用者身分認證程序的，步驟如下：

Step 1. UA 會先產生一亂數為 a 值，再利用 a 值求得 $A = g^a$ mod N 後，傳送夾帶{ username, A }認證資訊的請求訊息給 Registrar。

Step 2. Registrar 收到註冊請求訊息之後，會先從訊息內取得要求進行認證的帳號名稱，查詢該帳號在認證資料庫內的相關訊息，如 salt 值與 v 值。如果該認證帳號存在，會隨機產生一亂數 b 值，再利用 b 值與從認證資料庫內取得的 v 值代入公式計算出 $B = (v + g^b)$ mod N，接著 Registrar 會傳送夾帶著{ salt, B }認證資訊的認證挑戰訊息給 UA。

A1 = username ":" realm ":" S	(6)
A2 = method ":" Request-URI	(7)
response = request-digest = H(H(A1) ":" u ":" H(A2))	*(8)*
m = H(A || response || S)	*(9)*

圖 3-3　SRPSA：response 計算公式

Step 3. 當 UA 收到從 Registrar 來的認證挑戰訊息之後，UA 會經由圖 3-2 公式(4)求得 x 值，並且將 A 值與 B 值合併雜湊求得 u 值，再利用已知的 A、B、a、x、u 值代入認證公式導出 $S = (B\text{-}g^x)^{a+ux}$ mod N。我們將求出的 S 值與原本已知的

username-value 和 realm-value 代入圖 3-3 公式(6)為 A1 字串；SIP Method 與 Request-URI 代入圖 3-3 公式(7)為 A2 字串：分別將 A1 字串與 A2 字串做雜湊之後與 u 值代入圖 3-3 公式(8)即可求得 response 值，再將夾帶著{username, realm, response}等認證資訊的挑戰回覆訊息傳送給 Registrar。

Step 4.

(1) 驗證 UA 的合法性：

Registrar 收到由 UA 傳來的挑戰回覆訊息之後，會先從訊息內取出由 UA 端計算出來的挑戰回覆 response 值。將自己亂數得到的 b 值、從 UA 那所得到的 A 值、資料庫內所保存的 v 值與 A, B 合併雜湊之後得的 u 值，代入公式求得 $S = (Av^u)^b \bmod N$。將求出的 S 值、username-value 和 realm-value 代入圖 3-3 公式(6)為 A1 字串；SIP Method 與 Request-URI 代入圖 3-3 公式(7)為 A2 字串：分別將 A1 字串與 A2 字串做雜湊之後與 u 值一起代入圖 3-3 公式(8)可求得 Registrar 端的 response 值。再將從 UA 挑戰回覆訊息中所取得的 response 值與 Registrar 端自行計算出的 response 值做字串比對。

(2) 若字串相同比對相同，則進行挑戰回覆：

Registrar 會將已知的 A 值、response 值與 S 值代入圖 3-3 公式(9)求得 m 值，而 m 值會被夾帶在 Registrar 的回覆訊息裡，傳送給 UA，提供 UA 驗證 Registrar 是否為合法。

Step 5. UA 端收到來自 Registrar 端的回覆訊息之後，會取出從 Registrar 端計算出來的 m 值，然後 UA 會依照圖 3-3 公式(9)求出 UA 端的 m 值，將從 Registrar 端傳送來的 m 值與自己計算出來的 m 值做比對。如果字串相同，則代表 Registrar 為合法的伺服器。

我們所提出的 SRPSA 主要是藉由驗證兩方 S 值是否相同來達成相互認證的安全性。為了避免受到偽裝伺服器攻擊，所以我們在 Step 1 與 Step 2 中是先讓 UA

和 Registrar 交換個別亂數所產生的認證金鑰；在 Step 3 與 Step 4 則是 UA 與 Registrar 再藉由從對方那所得到的認證金鑰，兩端各分別透過$(B\text{-}g^{x})^{a+ux} \bmod N$與$(Av^{u})^{b} \bmod N$兩公式推導出有相同結果的 S 值($S = g^{ab+bux} \bmod N$)。然後再將推導出來的 S 值與其他需要被驗證有無被竄改過的標頭欄位合併雜湊，這樣一來可以保證 UA 與 Registrar 各別計算出來的 S 值對方都不會知道。由於 S 值產生是依靠兩端亂數產生的認證金鑰經由認證公式推導得到的，所以 S 值在認證前是不可能被預測得到的，因此 S 值在進行該次註冊認證成功之後，可以被當作該次連線的會議金鑰。

3.2 以數位簽章為基礎之SIP認證機制

我們在 3.1 節中提出 SRPSA，該認證演算法可以達到 UA 與 Registrar 互相認證的功能，並且也能防範目前所提出的大多數 SIP 認證機制的安全性缺失，如離線式密碼猜測攻擊與竊取驗證碼攻擊。但是其認證方法需要大幅的修正現有基於 HTTP Digest 的 SIP 認證系統程式，並且該演算法雖然可以防止竊取驗證碼攻擊，卻可以藉由使用者身分的驗證碼與窮舉法得到的結果兩者比對驗證後，即可順利猜得使用者真正的密碼，安全性依然堪慮。於是，我們思考著另一種可能的方向，有沒有辦法讓使用者並不需要把自己私密的密碼資訊存放在 Registrar 認證資料庫，然而卻一樣可以對使用者身分進行驗證；又或者是，使用者真正的密碼與認證伺服器端所保存的身分驗證碼是各別不同的鑰匙，並且存放在認證伺服器中的使用者身分驗證碼也無法反推求得使用者的真正密碼。

基於以上考量，我們評估現行的公開金鑰密碼學(Public-Key Cryptography)是一個不錯的解決方案。在 2009 年我們提出以非對稱式認證金鑰為基礎的認證方法(Asymmetric-key-based SIP Authenticaiton Scheme，簡稱 AAS)[23]，但是其認證方法被人所詬病的是在於非對稱式密碼在加、解密時所需要花費的計算成本遠高於基於對稱式密碼學(Symmetric Cryptography)的認證方法。在大多數被提出的 SIP 認證機制中，欲提供挑戰的資訊或者是針對挑戰回應的認證資訊皆會被做加密保護。若是直接以公開金鑰密碼演算法來做加、解密運算，效能較不佳。因此在幾經改良後，我們在本論中提出的方法，將採用同樣具有非對稱特性，但不需要付出高計算代價的數位簽章系統(Digital Signature)，來改善先前效能較為不彰的 AAS。

在介紹我們所提出的改善方法之前，我們先說明數位簽章概念。數位簽章系

統主要是由訊息摘要演算法(Message Digest Algorithm)搭配數位簽章演算法(Digital Signature Algorithm)合併運作。目前常見的訊息摘要演算法有Message-Digest Algorithm 5(簡稱 MD5)與 Secure Hash Algorithm 1(簡稱 SHA-1)[24]。另外，為了提高安全性還有 SHA-256 與 SHA-512…等；而常見的數位簽章演算法有 Rivest Shamir Adleman[25](簡稱 RSA)、Digital Signature Algorithm(簡稱 DSA)、Elliptic Curve Digital Signature Algorithm[26](簡稱 ECDSA)。接下來，我們簡單說明數位簽章系統的運作機制。圖 3-4 為數位簽章運作流程圖。

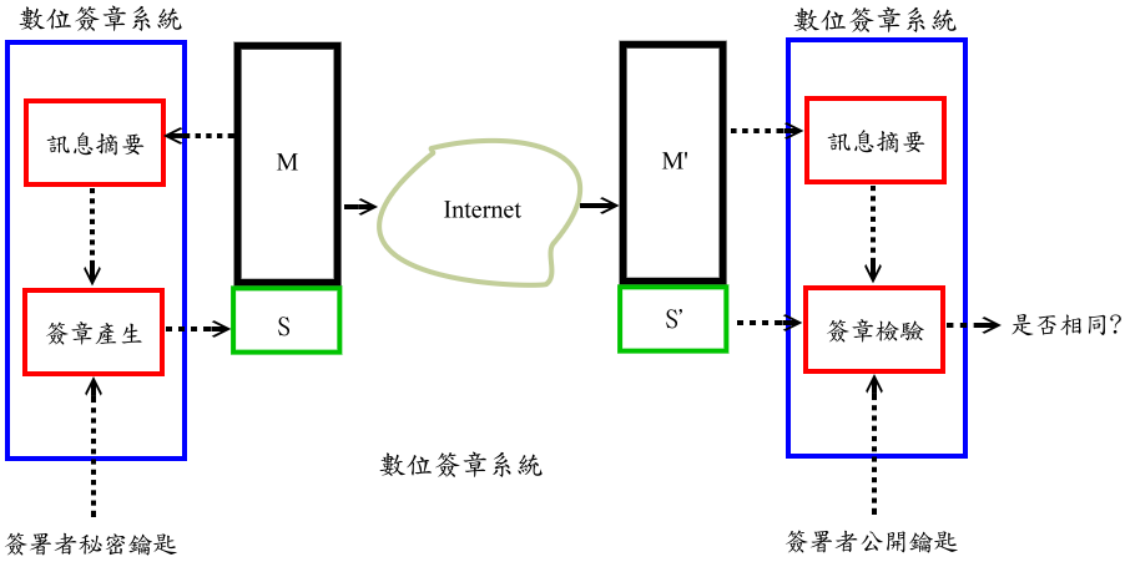

圖 3-4　數位簽章運作流程圖

數位簽章系統的運作機制分為兩個部分，第一部分為簽署者使用自己的秘密鑰匙對訊息做簽章；第二部分則為訊息接受者使用簽署者的公開鑰匙驗證該簽章值，檢查訊息是否合法，運作步驟如下所述：

(1) 簽署者進行明文簽署：

Step 1. 將明文 M 經過雜湊函數的運算取得訊息摘要。

Step 2. 使用簽署者的秘密鑰匙對訊息摘要做簽署，即可得到 S 值；S 值就是此明文 M 的簽章值。

Step 3. 接著，我們將明文 M 與簽章 S 一起傳送給目標接收者。

(2) 訊息接收者驗證簽名：

Step 1. 將收到的明文 M'經過單向雜湊函數的運算取得該訊息摘要值。

Step 2. 利用簽署者的公開鑰匙來驗證簽章值 S'是否與在 Step 1 中所取得的訊息摘要相同。

在介紹數位簽章概念之後，我們於本節提出基於數位簽章演算法的 SIP 認證機制(Digital Signature on SIP Authenticaiton，簡稱 DSSA)。表 3-2 為 DSSA 符號說明。圖 3-5 為 DSSA 認證流程圖。

表 3-2　DSSA 符號說明

符號	符號定義
username	欲進行認證帳號名稱。
realm	註冊伺服器所在領域位置。
Call-ID	SIP 定義的標頭欄位之一，具備亂數不可事先預知的特性。
nonce	由註冊伺服器隨機產生且具備時效性的亂數值。
R_X	註冊伺服器的公開鑰匙。
R_Y	註冊伺服器的私密鑰匙。
C_X	使用者的公開鑰匙。
C_Y	使用者的私密鑰匙。
||	字串連結。

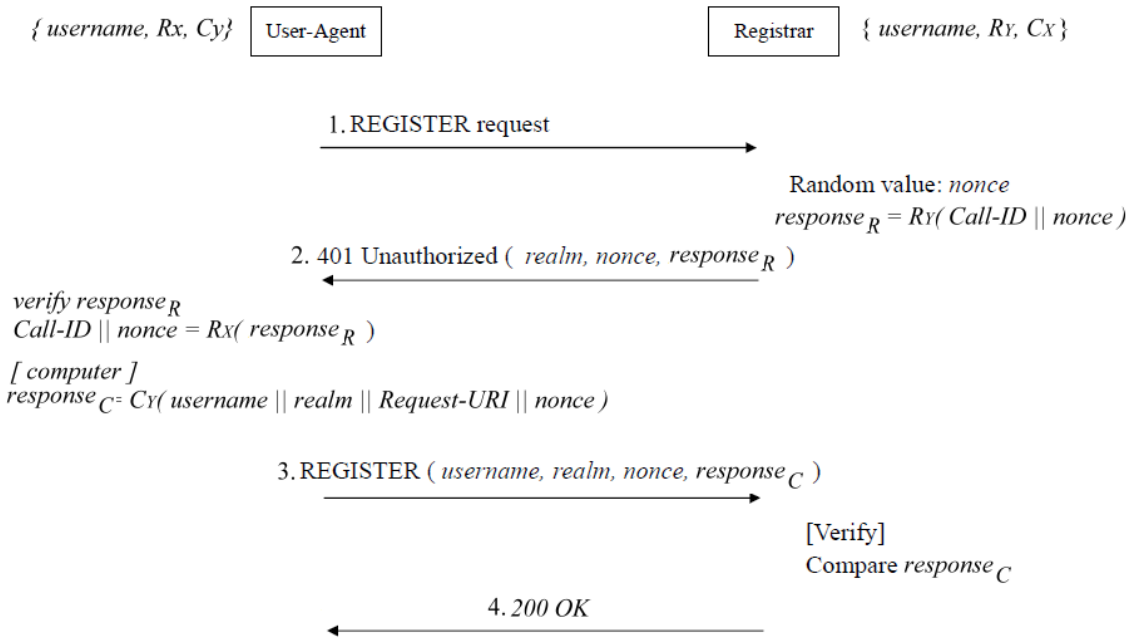

圖 3-5　DSSA 認證流程圖

我們在此先假設下列條件：使用者與 Registrar 各都有一對金鑰對分別稱為公開鑰匙與秘密鑰匙，其中使用者會與 Registrar 互相擁有對方的公開鑰匙。這樣一來，Registrar 的認證資料庫中會保有使用者帳號名稱與使用者的公開鑰匙，當然還有 Registrar 自己擁有的金鑰對；而使用者則有自己的金鑰對與 Registrar 的公開鑰匙。認證步驟如下：

Step 1. UA 發送註冊請求訊息給 Registrar。

Step 2. Registrar 收到 UA 的請求訊息之後，先取出訊息中標頭欄位為 Call-ID 的參數值，然後隨機產生一亂數值為 nonce，再將 Call-ID 與 nonce 合併後使用自己的秘密鑰匙產生簽章值為 response_R，將夾帶{ realm, nonce, response_R }三個認證挑戰的認證挑戰訊息傳送給 UA。

Step 3. UA 收到 Registrar 的認證挑戰訊息之後：

(1) 驗證 Registrar 的合法性：

UA 將在 Step 1 中自己亂數產生的 Call-ID 欄位參數與 Registrar 挑戰訊息內的 nonce 值合併，使用 Registrar 的公開鑰匙驗證訊息內的 $response_R$ 值是否與字串合併後的結果相同。

(2) 若驗證成功，則進行挑戰回覆：

UA 將要被認證帳號名稱, Request-URI 與認證挑戰訊息內所給予的{ realm, nonce }兩個參數進行字串合併為 *username* || *realm* || *Request-URI* || *nonce*，再使用自己的秘密鑰匙對合併後的字串產生簽章為 $response_C$ 值，最後將夾帶{ username, realm, nonce, $response_C$ }四個認證資訊的挑戰回覆訊息傳送給 Registrar。

Step 4. 當 Registrar 收到 UA 的挑戰回覆訊息之後，先檢查訊息中的 nonce 是否合法，如果合法，Registrar 就會利用 username 查詢認證資料庫，以取得該認證帳號存放於認證資料庫中相對應的公開鑰匙，再用此公開鑰匙驗證訊息內所提供的 $response_C$ 值是否與{ username, nonce, realm, Request-URI }合併後的字串相同。如果相同，則表示使用者身分驗證成功，註冊伺服器會回傳 200 OK 的認證回覆訊息給 UA，UA 便可以使用 SIP 代理伺服器的各種服務；如果不同，則表示使用者的身分驗證失敗，SIP 代理伺服器便拒絕接受 UA 端發送的請求訊息。

在我們所提出的 DSSA 中，我們讓 UA 與 Registrar 都各自產生亂數值，以預防重播攻擊(Replay Attack)，並且也不需要對兩個被提供用以認證挑戰的亂數值做加密保護，即使它們被公開於網路上傳送也不會對整個認證程序的安全性造成影響。我們利用公開金鑰密碼學中的數位簽章的概念，故簽署者的秘密金鑰與驗證者所保有的簽署者公開金鑰並不是相同的一把認證金鑰，即使存放在驗證端的公開金鑰被竊取，也不會影響到簽署者的權益，這樣一來便可防止竊取驗證碼攻擊。在我們實作數位簽章系統中，所使用的認證金鑰長度預設值是 1024 bits。對於普羅大眾所能取得電腦設備的計算能力而言，這並不是短時間之內的計算就可以破

解取得簽署者的私密金鑰，所以對於阻擋離線式密碼猜測攻擊也能保有一定的安全性。最後，因為 UA 與 Registrar 各有一對金鑰對，並且也各自使用自己的秘密鑰匙對認證資訊產生簽署，可以保障該認證資訊是由本人所發出，所以我們所提出的認證機制也可達成互相認證的功能。

第4章 系統實作與時間量測

因為 SRP 認證機制已被實作在多種協定(如 Telnet、FTP)的認證機制上，所以我們在這就不探討 SRPSA 的實作機制。我們於本章節實作於 3.2 節所提出的 DSSA 認證演算法，並且進行認證程序所需花費時間的量測。4.1 節先行說明我們所建立的 SIP 認證機制的系統架構、實驗環境與認證機制運作；4.2 節則是展示與說明實作的成果；4.3 節則是針對使用者代理端不同認證金鑰長度的需求進行時間的量測，並且將量測後的結果與現行 SIP 認證機制做比較。

4.1 系統架構

在我們實作 SIP 認證機制系統中，建立了兩個端點分別為使用者代理端(網際網路協定位址為 10.10.21.174)與註冊伺服器(網際網路協定位址為 163.22.21.1)。圖 4-1 為系統架構圖。

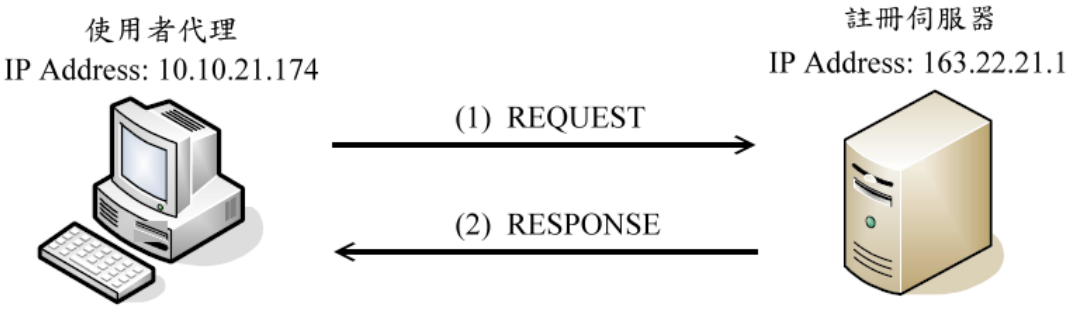

圖 4-1 系統架構圖

兩端點分別建立的實驗環境如下：在作業系統方面，使用者代理端是採用 Linux 眾多發行版之一的 CentOS[27]，註冊伺服器則是安裝 FreeBSD[28]；在中央處理器部分，使用者代理端是使用 Intel Pentium Dual E2140 1.6G Hz(雙計算核心，64 位元)，註冊伺服器則是 Intel Pentium 4 3.4G Hz(單計算核心，32 位元)；在系統

記憶體部分，使用者代理端是 2560 MB，註冊伺服器是 1024 MB。在支援 SIP 協定的軟體部分，使用者代理端是由我們自己開發符合 SIP 標準文件中所定義的使用者身分認證程式[29]，而註冊伺服器則是架設現成的 SIP 代理伺服器(Open SIP Server，簡稱 OpenSIPS)[30]。OpenSIPS 是一套成熟並且開放原始碼可提供我們自行依照需求進行修改的 SIP 協定系統。此外為了讓 SIP 註冊伺服器可以分辨多個認證機制的運作，我們使用 *Supported* 欄位於使用者代理端的註冊請求訊息中。*Supported* 欄位是 SIP 協定標準中一項非常有彈性的設計，方便使用者代理端向註冊伺服器表明它所支援的衍生功能清單，讓註冊伺服器可以選擇使用者代理端所支援的認證方法進行驗證。在建立數位簽章系統方面，我們選擇 RSA 簽章演算法與 MD5 訊息摘要演算法來實做認證機制，並且使用 OpenSSL 函式庫[31]撰寫程式以實現簽章與驗證的功能。我們預設使用者代理端與註冊伺服器使用的認證金鑰長度都是 1024 個位元。表 4-1 為使用者代理端與註冊伺服器系統實驗環境表。

表 4-1　系統實驗環境

系統角色	**使用者代理**	**註冊伺服器**
購置日期	2008 年	2005 年
作業系統	CentOS 5.3	FreeBSD 7.0
中央處理器	Intel Pentium Dual E2140 1.6G Hz	Intel Pentium 4 3.4G Hz
系統記憶體	2560 MB	1024 MB
網際網路協定位址	10.10.21.174	163.22.21.1
SIP 軟體	自行開發	opensips-1.4.4-tls
OpenSSL	0.9.81	
訊息摘要演算法	MD5	

數位簽章演算法	RSA	
預設金鑰長度	1024 bits	1024 bits

4.2 實驗結果展示

我們於本小節利用封包擷取軟體 Wireshark[32]於使用者代理端觀察並且展示使用者代理端於認證過程中所發出與接收到的 SIP 訊息，如圖 4-2 所示。在圖 4-2 中可以看到 Wireshark 能提供給我們的資訊有得到封包的時間、封包的來源與目的地位址、封包各層的網路協定以及封包訊息...等。在使用者代理端與註冊伺服器皆使用認證金鑰長度為 1024 位元的情況之下，此次認證過程所花費的時間是 0.01609 秒(8.888877-8.872680)，也就是 16.09 毫秒(ms)。

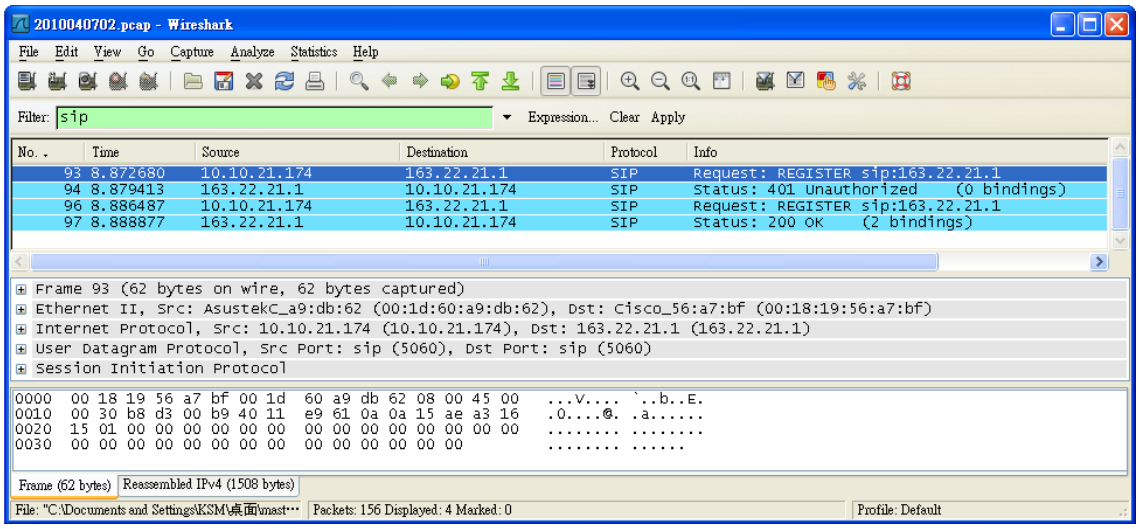

圖 4-2　利用 Wireshark 觀察得到的 SIP 封包訊息

圖 4-3 則是將抓取到的封包傳遞圖形化表示，此圖也符合我們於 3.2 節中所提出以非對稱式密碼為基礎的認證演算法之訊息傳遞方向與認證流程。

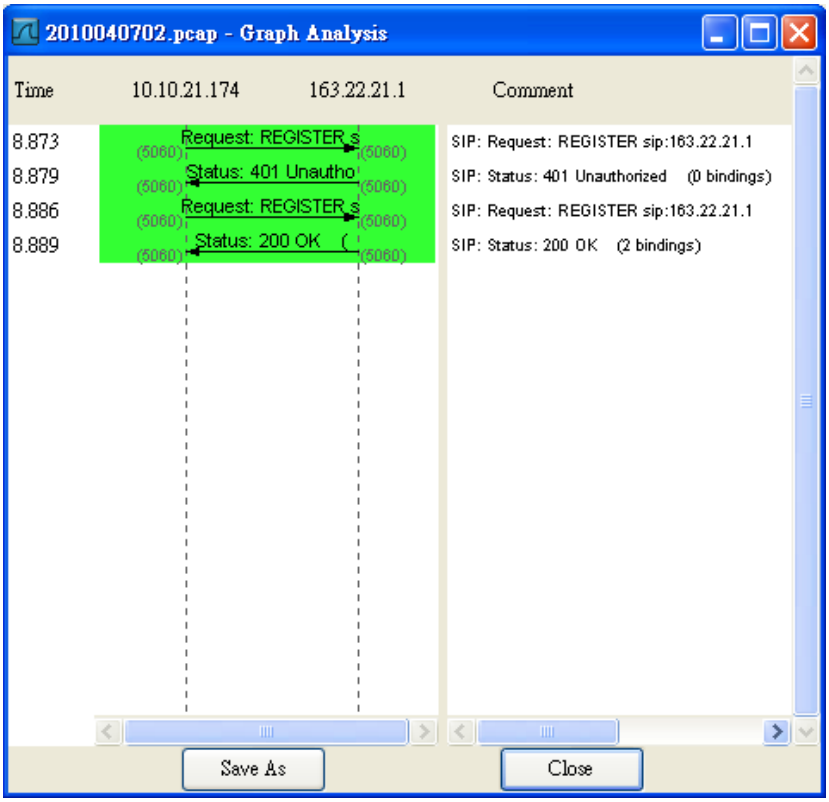

圖 4-3　SIP 訊息傳遞圖

圖 4-4 是使用者代理端向註冊伺服器請求註冊的 REGISTER 訊息。此訊息跟現行 SIP 認證訊息大致相同，唯一不同的是我們依據 SIP 標準文件中的定義增加新的標頭欄位為 *Supported*。此標頭欄位方便使用者代理端向註冊伺服器表明它所支援的衍生功能清單，讓註冊伺服器可以選用使用者代理端所支援的認證方法進行驗證。

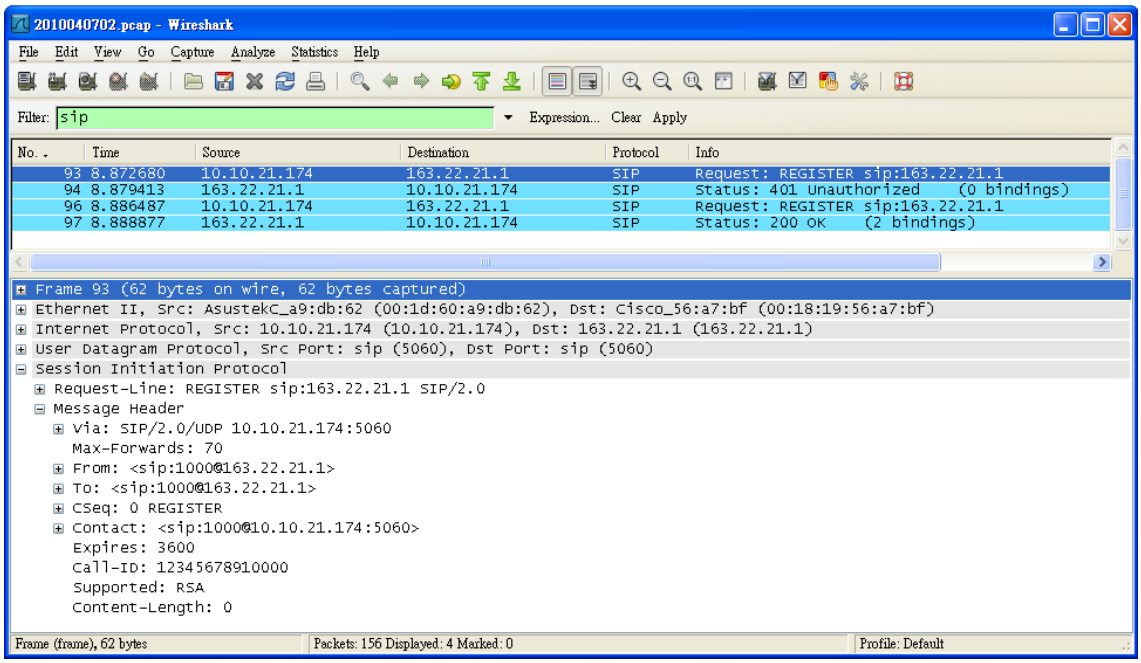

圖 4-4　使用者代理端向註冊伺服器請求註冊的 REGISTER 訊息

圖 4-5 是註冊伺服器發送給使用者代理端的認證挑戰訊息。此挑戰訊息跟現行認證機制的挑戰訊息內文大致相同，唯一不同的是我們於 *WWW-Authenticate* 欄位內新增加一個參數為 *Response*。此參數值是為了讓註冊伺服器用以存放自己計算出來的身分認證資訊，提供使用者代理端得檢查該參數值以驗證註冊伺服器的合法性。

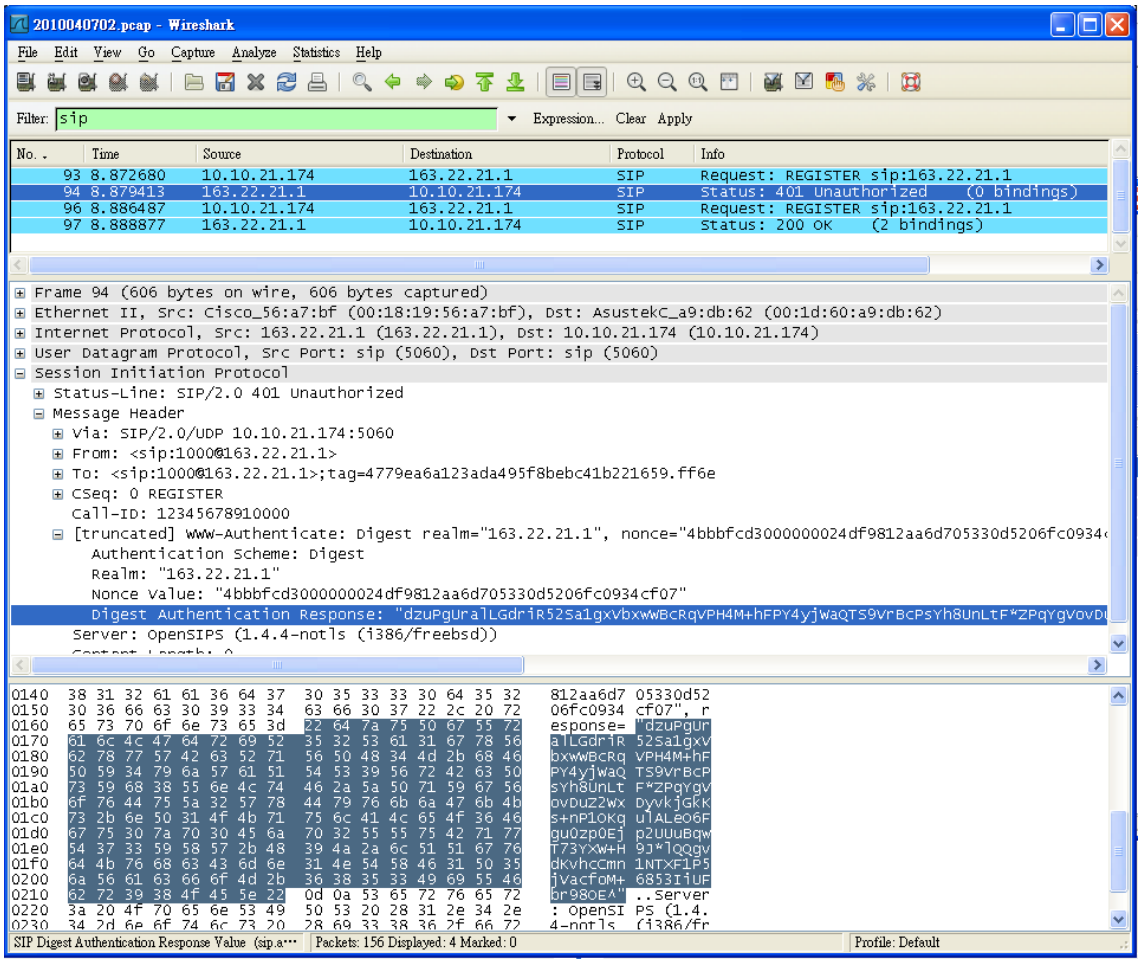

圖 4-5　註冊伺服器發送給使用者代理端的認證挑戰訊息

圖 4-6 是使用者代理端傳送給註冊伺服器挑戰回覆訊息。如同現行 SIP 認證機制的註冊訊息內容，使用者代理端利用標頭欄位 *Authorization* 來存放所要給予註冊伺服器的認證挑戰訊息，提供註冊伺服器利用此標頭欄位來驗證使用者身分的合法性。

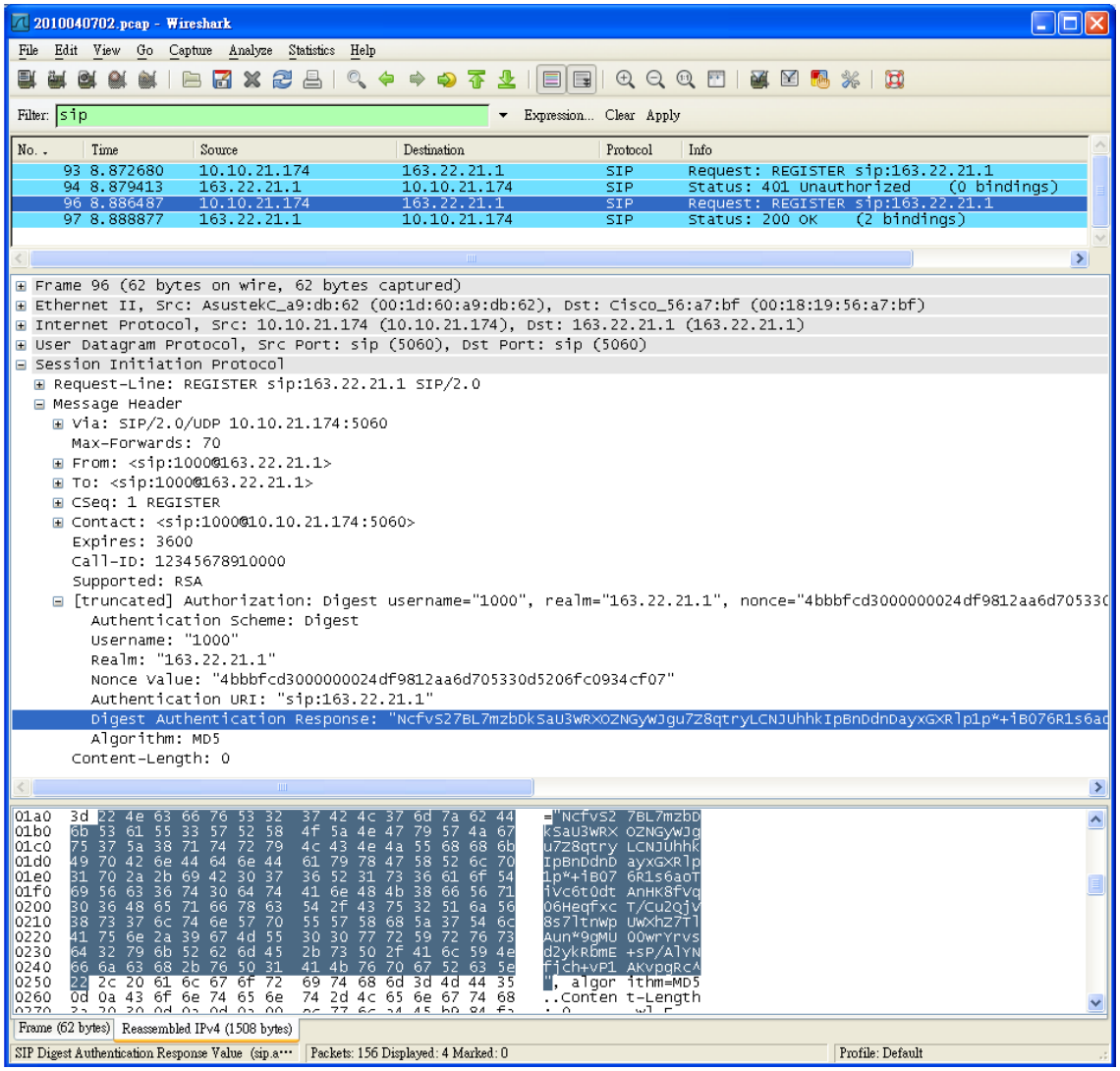

圖 4-6　使用者代理端傳送給註冊伺服器挑戰回覆訊息

圖 4-7 是註冊伺服器回應給使用者代理端認證成功訊息。此回應訊息表示使用者身分認證成功，其中標頭欄位 *Contact* 會表示目前此認證帳號已經註冊的網際網路位置、連接埠以及 *expires* 帳號註冊到期時間(以秒為單位)，如圖中的 <sip:1000@10.10.59.70:5060>;expires=3480,<sip:1000@10.10.21.174:5060>;expires=3600。

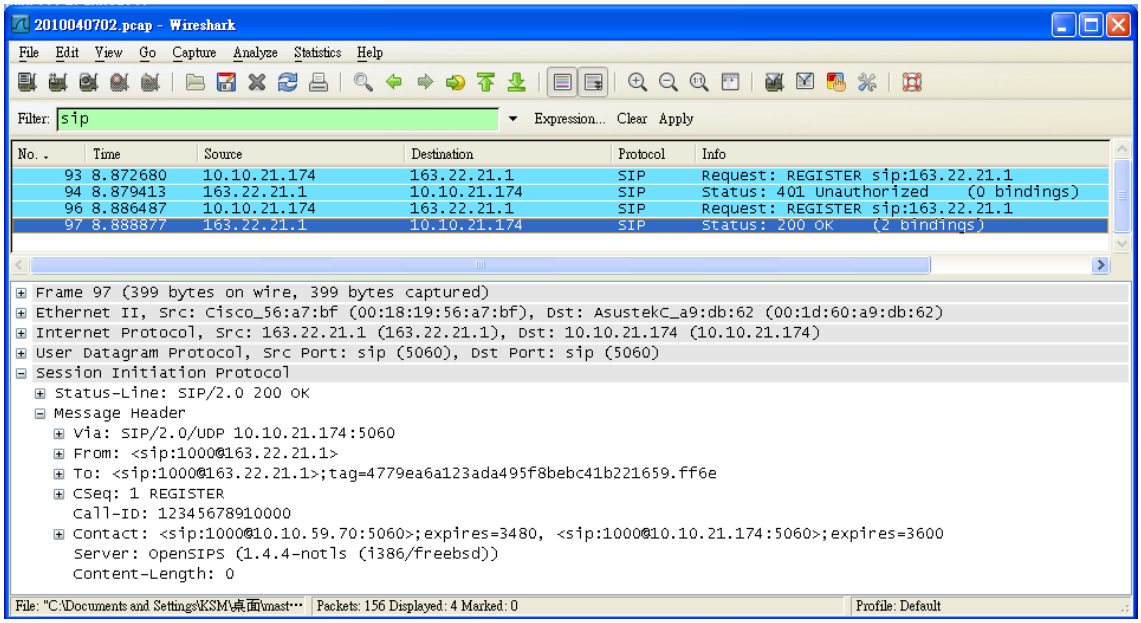

圖 4-7　註冊伺服器回應給使用者代理端認證成功訊息

4.3 時間量測

我們於本節進行認證程序時所需花費時間的量測工作。在 4.3.1 節中，我們先量測認證機制中會使用到的 OpenSSL 函式庫，各別觀察 OpenSSL 函式庫分別在使用者代理端與註冊伺服器上執行明文簽章與簽章驗證所需要花費的時間。之後於 4.3.2 節中，我們對 3.2 節中實作出來的認證系統分別針對使用者代理端不同的認證金鑰長度進行認證程序所需花費時間量測。表 4-2 為本節中將共同使用的符號說明。

表 4-2　本節共同使用符號說明

符號	符號定義
T_{sign}	進行簽名所需要花費的時間。
T_{verify}	進行簽名驗證所需要花費的時間。
RTT	網路封包往返的時間(Round-Trip Time)，表示從發送端發送訊息開始，到發送端收到來自接收端的確認訊息，總花費的時間。
T_{sum}	實際量測認證程序所得到的時間值。

4.3.1 OpenSSL函式庫時間效能量測

本小節首先針對作業系統已預設安裝的 OpenSSL 函式庫進行時間量測，OpenSSL 已內建演算法時間測量的選項，如表 4-3 所示。表 4-4 為註冊伺服器針對不同的認證金鑰長度執行簽章與驗證簽署所需花費時間；表 4-5 為使用者代理端(單核心處理) 針對不同的認證金鑰長度執行簽章與驗證簽署所需花費時間。因為我們使用者代理端的計算設備是雙核心處理器，並且 OpenSSL 本身也有支援平行處理的功能，所以我們也對 OpenSSL 函式庫執行平行處理這部分進行時間量測以供參考。表 4-6 為使用者代理端(雙核心平行處理) 針對不同的認證金鑰長度執行簽章

與驗證簽署所需花費時間。

表 4-3　OpenSSL 內建公開金鑰演算法時間量測選項

指令	**指令說明**
speed	對 OpenSSL 所提供的所有加、解密演算法做時間量測。
speed rsa	僅針對 RSA 演算法進行時間量測。
speed rsa512	僅針對 RSA 演算法其金鑰長度為 512 個位元進行時間量測。
speed rsa -multi 2	僅對 RSA 演算法，以雙核心處理器進行平行處理時的時間量測。

表 4-4　註冊伺服器簽章/驗證所需花費時間

金鑰長度(bits)	T_{sign} (ms)	T_{verify} (ms)
512	0.768	0.063
1024	4.013	0.204
2048	26.195	0.715
4096	175.137	2.555

表 4-5　使用者代理端(單核心處理)簽章/驗證所需花費時間

金鑰長度(bits)	T_{sign} (ms)	T_{verify} (ms)
512	0.760	0.060
1024	3.395	0.150
2048	17.846	0.453
4096	107.957	1.488

表 4-6　使用者代理端(雙核心處理)簽章/驗證所需花費時間

金鑰長度(bits)	T_{sign} (ms)	T_{verify} (ms)
512	0.375	0.029
1024	1.665	0.074
2048	8.749	0.224
4096	53.060	0.734

由表 4-5、表 4-6 可以清楚看到 OpenSSL 函式庫中的 RSA 演算法在同一顆 CPU 中使用單處理器與使用雙處理器情況下，所執行的加解密時間大為不同。若金鑰長度為 1024 個位元，雙處理器執行簽章與驗證所需要花費的時間大約是單處理器的一半，可以有效的減低非對稱加、解密所需要的時間。因為現今大多數的電腦最低配備都已具備雙核心處理能力，所以此份數據值得參考。

4.3.2　DSSA認證機制時間量測

在進行時間量測之前，我們先行分析一個註冊認證的程序其所應該會花費的各時間區段，如圖 4-8 所示。以及各時間區段的定義說明，請參閱表 4-7。

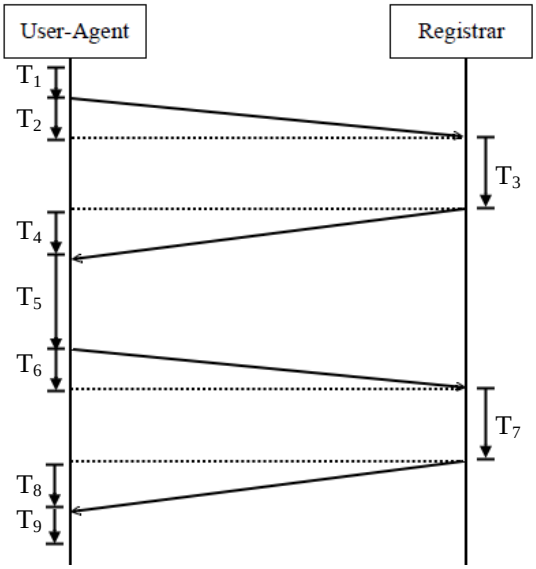

圖 4-8　認證程序時間分析圖

表 4-7　時間分析符號定義

符號	符號定義
T_1	使用者代理端建立 SIP 註冊訊息所需花費的時間。
T_2, T_4, T_6, T_8	SIP 訊息於網路上傳遞所需花費的時間。
T_3	註冊伺服器解析訊息、產生認證資訊與建立挑戰訊息所需花費的時間。
T_5	使用者代理端解析訊息、驗證與產生認證資訊及建立認證回應訊息所需花費的時間。
T_7	註冊伺服器解析訊息、驗證認證資訊與建立回應訊息所需花費的時間。
T_9	使用者代理端解析訊息所需花費的時間。

我們所量測得到時間值是各時間區段相加過後的值，是使用者代理端從發起 SIP 註冊請求訊息到最後收到認證成功訊息所花費的總時間(Wall Time)。表 4-8 是

我們固定註冊伺服器的金鑰長度，然後對使用者的金鑰長度做改變所量測到的時間值。在表中所呈現的值為表 4-7 內定義各時間分隔的總和，如公式(10)。

$$T_{sum} = T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_7 + T_8 + T_9 \qquad (10)$$

表 4-8　使用者代理端不同金鑰長度進行認證時所花費的時間

使用者的金鑰長度(bits)	T_{sum} (ms)
512	11.194
1024	17.787
2048	32.290
4096	135.550

現行以 HTTP Digest 為基礎的 SIP 認證機制完成認證程序所需要花費的平均時間為 3.253 ms，而我們提出的 DSSA 在以 1024 bits[33]為預設非對稱金鑰長度的情況下，完成認證程序所需要花費的時間為 17.787 ms；與現行的 SIP 認證機制相比，我們的方法需要比它多花費不少的時間，主要的原因是在於現行 SIP 認證機制所用到的計算公式是以字串連結、XOR 邏輯運算與單向雜湊函數等低計算複雜度的運算所構成，而我們的方法則是由字串連結與非對稱密碼學中數位簽章系統所構成。然而我們的方法雖然需要多花費一些時間，但是其所需要花費的時間仍然是在我們沒有等待的感覺內完成認證。我們方法可以提供比現行 SIP 認證機制更多的安全性，像是可以抵抗離線式密碼猜測攻擊與竊取驗證碼攻擊，並且也可達到相互認證的功能，大大的增加了使用者身分認證的安全性。

第5章 安全性與計算複雜度分析

我們在前一章說明系統實作與時間量測結果之後，在本章我們將我們所提出的方法與各專家學者針對 SIP 認證機制所提出的演算法做一個安全性的比較與計算複雜度的統計。我們根據 1.2 節表 1-3 所列的攻擊類型來比較各專家學者與我們所提出針對 SIP 認證演算法的安全性分析，表 5-1 為各種認證演算法安全性綜和比較表。表 5-2 是我們在計算複雜度統計時會使用到的符號說明。表 5-3 則是我們統計各 SIP 認證演算法在 UA 與 Registrar 各自需要的計算複雜度，並且最後合併取得該認證方法所需的總計算複雜度。

我們所提出 SRPSA 認證演算法主要比較對象是現行 SIP over TLS 的認證機制。SRPSA 不存放明文形式的密碼於註冊伺服器端，這樣一來可確保足以抵擋 A5. 竊取驗證碼攻擊。在認證程序中採用隨機產生亂數生成認證金鑰並且進行彼此認證金鑰交換，來用以保護使用者的真實密碼，這樣的機制可以抵擋 A2.離線式密碼猜測攻擊 I。並且我們也不需透過第三者公正方來達成互相認證的功能，重要的是使用者不需改變使用習慣，僅需要記住自己所設定的登入密碼即可。在計算複雜度方面，雖然以 HTTP Digest 為基礎的 SIP 認證機制遠比我們所提出的 SRPSA 認證機制低，如表 5-3 所示；但若是加入 TLS 機制協同運作，TLS 需要對整個 SIP 訊息做非對稱式加、解密運算，認證程序的四個流程皆會使用到指數運算，而指數運算的計算複雜度是 T_{EXP}，所以 SIP over TLS 總計算複雜度會是 $6T_H+8T_{EXP}$。

表 5-1　各認證演算法安全性綜和比較

	A1	A2	A3	A4	A5	A6
HTTP Digest[6]	Secure	Insecure	Insecure	Secure	Insecure	Insecure
Yang 等人[8]	Secure	Secure	Secure	Secure	Insecure	Insecure

Huang 等人[10]	Secure	Insecure	Secure	Secure	Secure	Insecure
Guillet 等人[11]	Insecure	Insecure	Secure	Secure	Secure	Insecure
Tsai[12]	Secure	Insecure	Secure	Secure	Insecure	Insecure
SRPSA[20]	Secure	Secure	Secure	Secure	Secure	Insecure
DSSA	Secure	Secure	Secure	Secure	Secure	Secure
A1.重播攻擊 (Replay Attack) A2.離線式密碼猜測攻擊 I (Off-Line Password Guessing Attack I) A3.伺服器偽裝攻擊 (Server Spoofing Attack) A4.使用者假冒攻擊 (Impersonation Attack) A5.竊取驗證碼攻擊 (Stolen-Verifier Attack) A6.離線式密碼猜測攻擊 II (Off-Line Password Guessing Attack II)						

表 5-2　計算複雜度符號說明

符號	行為定義
T_{XOR}	執行一次位元運算所需要花費的時間。
T_H	執行一次單向雜湊函數所需要花費的時間。
T_{EXP}	執行一次指數運算所需要花費的時間。
T_{SIGN}	執行一次簽名所需要花費的時間，以 RSA 數位簽章演算法為例是一次指數運算加一次單向雜湊運算。 ($1\ T_{SIGN} = 1\ T_H + 1\ T_{EXP}$)
T_{VERIFY}	執行一次驗證簽名所需要花費的時間，以 RSA 數位簽章演算法為例是一次指數運算加一次單向雜湊運算。 ($1\ T_{VERIFY} = 1\ T_H + 1\ T_{EXP}$)

表 5-3　各認證演算法計算複雜度比較

	UA 端 計算複雜度	Registrar 端 計算複雜度	總計算複雜度
HTTP Digest[6]	$3T_H$	$3T_H$	$6T_H$
SIP over TLS	$3T_H + 4T_{EXP}$	$3T_H + 4T_{EXP}$	$6T_H + 8T_{EXP}$
Yang 等人[8]	$2T_{XOR}+3T_H+2T_{EXP}$	$2T_{XOR}+3T_H+2T_{EXP}$	$4T_{XOR}+6T_H+4T_{EXP}$
Huang 等人[10]	$2T_{XOR}+2T_H$	$2T_{XOR}+2T_H$	$4T_{XOR}+4T_H$
Guillet 等人[11]	$3T_H$	$3T_H$	$6T_H$
Tsai[12]	$1T_{XOR}+4T_H$	$1T_{XOR}+4T_H$	$2T_{XOR}+8T_H$
*SRPSA[20]	$5T_H+2T_{EXP}$	$5T_H+2T_{EXP}$	$10T_H+4T_{EXP}$
*DSSA	$1T_{SIGN}+1T_{VERIFY}$ $= 2T_H+2T_{EXP}$	$1T_{SIGN}+1T_{VERIFY}$ $= 2T_H+2T_{EXP}$	$2T_{SIGN}+2T_{VERIFY}$ $= 4T_H+4T_{EXP}$
備註：星號(*)表示為本論文所提出的方法。			

我們所提出 DSSA 認證方法中，利用了公開金鑰密碼學中金鑰對的特性，將使用者的公開金鑰存放於註冊伺服器的認證資料庫中，因為公開金鑰要反推得到私密金鑰是非常困難的，所以保證可抵擋 A5.竊取驗證碼攻擊，並且也可以抵擋 A6.離線式密碼猜測攻擊 II，對於使用者於認證程序上的安全性可以得到很好的保障。另外，因為是採用公開金鑰密碼學演算法，故其對於抵擋 A2.離線式密碼猜測攻擊 I 上的表現可以較其他的計算複雜度低的身分認證演算法更佳。

在計算複雜度方面，我們所提出的 DSSA 與學者 Yang 等人所提出，同是基於非對稱式密碼學的認證方法作比較，且針對 A5.竊取驗證碼攻擊與 A6.離線式密碼猜測攻擊 II 進行安全性分析。首先，我們先假設系統認證資料庫中所保存的使用者帳號與認證所需金鑰已經被不法人士入侵竊取或是管理人員監守自盜，所以第

三者已知使用者帳號及其相對應之認證金鑰的情況下，兩認證演算法攻擊成功的時間複雜度為何。表 5-4 為 A5、A6 攻擊時間複雜度分析。

(1) A5.竊取驗證碼攻擊時間複雜度分析：

Yang 等人所提出的認證方法是以使用者所設定的密碼為認證金鑰，並且將其以明文模式存放在註冊伺服器的認證資料庫中，所以當不法人士將使用者的密碼竊取之後，便可以直接以該認證金鑰冒名成功登入使用者帳號，所以其時間複雜度為常數，O(1)；而我們所提出的 DSSA 因為保存在認證資料庫中的是該使用者的公開鑰匙，所以即使不法人士竊取得到該使用者的公開鑰匙，也無法直接以公開鑰匙成功進行登入認證。因為不可能攻擊成功，所以此項時間複雜度是不存在的。

表 5-4　A5、A6 攻擊成功時間複雜度分析

	A5	A6
Yang 等人	f(n) = 1 = O(1)	f(n) = 1 = O(1)
*Our	N/A	$f(n) = 2^n = O(2^n)$
註: N/A =不存在。		

(2) A6.離線式密碼猜測攻擊 II：

Yang 等人的認證方法於驗證表中所存放的認證金鑰是使用者所設定的密碼，所以不法人士無需再使用窮舉法反推，即可求得使用者的私密鑰匙，所以在這個項目中，Yang 等人所提的方法攻擊成功的時間複雜度為 O (1)；而我們所提出的 DSSA 於驗證表中所存放的認證金鑰為使用者的公開鑰匙，若不法人士想要知道使用者的私密鑰匙進行註冊登入，則必須使用窮舉法先假設一使用者私密鑰匙為 C_x’，然後再使用假設的私密鑰匙求得公開鑰匙為 C_y’，再比較兩公開鑰匙是否相

同。如果不同，就必須重新假設一私密鑰匙，一直重覆比對，直到字串比對相同為止。若該鑰匙對的長度為 n bits，$f(n)=2^n$，所以要攻擊成功的時間複雜度為 $O(2^n)$。

實例說明：

假設我們以雲端計算(Cloud computing)採用窮舉法的方式以破解取得使用者的私密鑰匙，此雲端系統可同時提供 10 億台電腦進行運算，每台電腦的 CPU 有 4G Hz，每 1Hz 可比對出一次結果，Yang 等人與我們所提出的 DSSA 認證演算法使用者的私密鑰匙長度皆為 1024 bits，以窮舉法攻擊成功所需花費的時間如表 5-5 所示。

表 5-5　以窮舉法攻擊成功所需花費的時間

認證演算法	時間
Yang 等人	$1 /((10^9) * (4*10^9)) = 2.5*10^{-19}$ (秒)
*Our	$2^{1024}/((10^9) * (4*10^9)) / (365*24*60*60)$ $= 1.42511188392814236723847760558 49*10^{282}$ (年)

Yang 等人所提出的方法：由於其驗證表中的認證金鑰就是使用者的私密金鑰，執行一次即可比對出使用者的私密鑰匙，所以僅需要 $2.5*10^{-19}$ 秒的時間使用者的私密金鑰即可被破解成功。

我們所提出的 DSSA：在私密金鑰為 1024 bits 的情況下，要由驗證表中的認證金鑰求得使用者的私密金鑰最多需要執行 2^{1024} 次的比對，我們同時使用 10^9 台電腦並且每台電腦的計算能力為 $4*10^9$Hz，合併計算為 $2^{1024} / ((10^9) * (4*10^9))$即可求得破解成功所需要花費的秒數，但因為秒數太大了不是這麼直觀，所以我們再換算成以年為單位，一年有 365 天、一天有 24 小時、一個小時有 60 分鐘、一分鐘有 60 秒，合併運算為 $2^{1024}/((10^9) * (4*10^9)) / (365*24*60*60)$，最後求出需要花

費 $1.42511188392814236723847760558 49*10^{282}$ 年的時間，才得以攻擊成功，取得與公開鑰匙相對應的私密金鑰。

第6章 結論

在本論文中，我們提出以非對稱式金鑰的特性來改善現行 SIP 認證方法中使用對稱式金鑰的弱點，目的是為了要加強使用者帳號管理的安全性，防止管理人員離職前的監守自盜或是認證資料庫遭到不明人士入侵竊取使用者密碼後冒名頂替的狀況，藉以保障帳號使用者個人的身分隱私與財務系統的安全。我們所提出的非對稱式金鑰系統與 SIP 原本使用對稱式金鑰的認證方式相比，SIP 信令個數是相同的；我們增加了對認證字串的簽章，經由對簽章的驗證，可以確保使用者身份與伺服器的合法性；經由實作認證系統並且進行認證程序時間的量測，驗證了我們演算法運用於實際通訊的可行性。

由於網路上每個人皆可以自行產生金鑰對(公開金鑰與私密金鑰)，那對方要怎麼確認你私密金鑰所對應的公開金鑰是合法的呢？又如何確認該公開金鑰是由本人所發佈的，所以一般而言，非對稱式密碼學需要第三方公正機制，也就是所謂 Public Key Infrastructure (PKI)的設置。在我們於 3.2 節所提出的以數位簽章為基礎的 SIP 認證機制可以不需要 PKI，因為使用者第一次註冊帳號需要臨櫃辦理，由網路電話服務業者隨機產生與提供使用者金鑰對，然後將使用者的金鑰對與註冊伺服器的公開金鑰一起存放至智慧卡中，同時也將使用者的公開金鑰存放於註冊伺服器中。使用者欲進行登入時，僅需讀取智慧卡取得合法帳號的金鑰對即可，所以我們的系統可以在不需要 PKI 的情況下順利運作。

將 SIP 改用非對稱式金鑰的驗證方式，一來可以有效的防止遭受到離線式密碼猜測攻擊與竊取驗證碼攻擊時，就不怕使用者的秘密金鑰外洩造成合法使用者權益受到影響；二來可以提供使用者代理端對註冊伺服器進行身分合法的驗證，是值得考慮的改進方向。

參考文獻

[1] Daniel Collins, *"Carrier Grade Voice over IP"*, 2nd Edition, McGraw-Hill, Sept. 2002.

[2] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, *"SIP: Session Initiation Protocol"*, IETF, RFC 2543, Mar. 1999.

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, *"SIP: Session Initiation Protocol"*, IETF, RFC 3261, Jun. 2002.

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *"Hypertext Transfer Protocol -- HTTP/1.1"*, IETF, RFC 2616, Jun. 1999.

[5] J. Klensin, *"Simple Mail Transfer Protocol"*, IETF, RFC 5321, Oct. 2008.

[6] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, *"HTTP Authentication: Basic and Digest Access Authentication"*, IETF, RFC 2617, Jun. 1999.

[7] YAHOO 新聞網，*"金泰希、具惠善等35名韓星個資外洩 身分證號碼、手機、住址全都露"*， http://tw.news.yahoo.com/article/url/d/a/090923/35/1rna0.html，2009 年 9 月。

[8] C. C. Yang, R. C. Wang, and W. T. Liu, *"Secure authentication scheme for session initiation protocol"*, Computers and Security, Vol. 24, pp. 381-386, 2005.

[9] W. Diffie and M. Hellman, *"New directions in cryptology"*, IEEE Transaction on Information Theory, Vol. 22, no. 6, 1976.

[10] H. F. Huang and W. C. Wei, *"A new efficient authentication scheme for Session Initiation Protocol"*, Proceedings of the 2006 Joint Conference on Information Sciences, Oct. 2006.

[11] T. Guillet, A. Serhrouchni, and M. Badra, *"Mutual Authentication for SIP: A Semantic Meaning for the SIP Opaque Values"*, New Technologies, Mobility and Security, pp. 1-6, Nov. 2008.

[12] J. L. Tsai, *"Efficient Nonce-based Authentication Scheme for Session Initiation Protocol"*, International Journal of Network Security, Vol. 9, No. 1, pp. 12-16, 2009.

[13] C. C. Lee, *"On Security of An Efficient Nonce-based Authentication Scheme for SIP"*, International Journal of Network Security, Vol. 9, No. 3, pp. 201-203, 2009.

[14] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "*HTTP Authentication: Basic and Digest Access Authentication*", IETF, RFC 2617, Jun. 1999.

[15] S. Josefsson, *"The Base16, Base32, and Base64 Data Encodings"*, IETF, RFC 3548, Jul. 2003.

[16] R. Rivest, *"The MD5 Message-Digest Algorithm"*, IETF, RFC 1321, Jun. 1992.

[17] T. Dierks and E. Rescorla, *"The Transport Layer Security (TLS) Protocol Version 1.2"*, IETF, RFC5246, Aug. 2008.

[18] T. Wu, *"The Secure Remote Password Protocol"*, Proceedings of the Internet Society Symposium on Network and Distributed System Security, pp. 97-111, 1998.

[19] T. Wu, *"The SRP Authentication and Key Exchange System"*, IETF, RFC 2945, Sept. 2000.

[20] Shin-Fu Huang and Quincy Wu, *"SIP Authentication Mechanism Based on the Secure Remote Password Protocol"*, Proceedings of Taiwan Academic Network Conference (TANet2009), Changhua, Taiwan, Oct. 2009.

[21] T. Wu, *"Telnet Authentication Option"*, IETF, RFC 2941, Sept. 2000.

[22] T. Wu, *"Telnet Authentication: SRP"*, IETF, RFC 2944, Sept. 2000.

[23] Shin-Fu Huang and Quincy Wu, *"An Asymmetric-key-based Authentication Scheme for Session Initiation Protocol"*, Proceedings of National Computer Symposium (NCS2009), Taipei, Taiwan, Nov. 2009.

[24] D. Eastlake and P. Jones , *"US Secure Hash Algorithm 1"*, IETF, RFC 3174, Sept. 2001.

[25] R. Rivest, A. Shamir, and L. Adleman, *"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems"*, Communications of the ACM 21 (2): 120–126, 1978.

[26] Accredited Standards Committee X9, *"American National Standard X9.62-2005, Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)"*, Nov. 2005.

[27] [CentOS, http://www.centos.org/]

[28] [FreeBSD, http://www.freebsd.org/]

[29] 李國川、陳建豪、陳宜暐、林琮哲、廖祥安、黃信富，*"嵌入式 VoIP 異質簡訊整合通訊平台設計"*，國立聯合大學資訊工程學系九十六學年度專題報告，2008 年 1 月。

[30] [OpenSIPS, http://www.opensips.org/]

[31] [OpenSSL, http://www.openssl.org/]

[32] [Wireshark, http://www.wireshark.org/]

[33] Elaine Barker and Allen Roginsky, *"Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes"*, DRAFT NIST Special Publication 800-131, NIST, Jan. 2010.