國立交通大學

資訊管理研究所

碩士論文

一個適用於 Ad-hoc 網路環境下的分散式提名代理簽章機制

A Distributed Nominative Proxy Signature Mechanism for Ad Hoc Networks

研究生：羅元琮

指導教授：羅濟群 教授

中華民國 九十五 年 六 月

一個適用於 Ad-hoc 網路環境下的分散式提名代理簽章機制

A Distributed Nominative Proxy Signature Mechanism for Ad Hoc Networks

研 究 生：羅 元 琮　　Student: Yuan-Tsung Lo

指導教授：羅 濟 群　　Advisor: Chi-Chun Lo

國立交通大學

資訊管理研究所

碩士論文

A Thesis

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science in Information Management

June 2006

Hsinchu, Taiwan, the Republic of China

中華民國　九十五 年 六 月

一個適用於 Ad-hoc 網路環境下的分散式提名代理簽章機制

研究生：羅元琮　　指導教授：羅濟群 老師

國立交通大學資訊管理研究所

摘要

在現代數位化的時代中，數位簽章常被利用於各種電子文件的簽署。而在過去，數位簽章通常都是要由簽章人自行來產生，如此一來也造成了許多的不便。因此，為了破除這種當簽章者不在便不能簽署文件的窘境，有學者提出了代理簽章的概念。如此一來，就算簽署者不在，亦可以指定一位代理人來幫他執行簽章的動作以產行有效的數位簽章。然而在相較於結構較為鬆散及節點流動性極大的 Ad-hoc 網路環境下，目前並沒有一個較為適合的代理簽章方式可以套用。因此，本篇論文便試著將各種現存的代理簽章方式截長補短，以 Threshold Proxy Signature 中的 secret sharing 方法來分散運算，此外，亦搭配 El-Gamal 的觀念來提高新方法的 security，提出一個更符合 Ad-hoc 網路環境的分散式提名代理簽章方法，而在論文的最後面，我們會利用 Manik Lal Das 提出的幾點數位代理簽章安全性需求及 Kaliski 提出的方法來對新舊方法做分析評估，以證明新方法較具安全性及彈性

關鍵字：Ad-hoc network，代理簽章，提名式代理簽章，門檻式代理簽章

A Distributed Nominative Proxy Signature Mechanism for Ad Hoc Networks

Student：Yuan-Tsung Lo Advisor：DR. Chi-Chun Lo

Institute of Information Management

Nation Chiao Tung University

Abstract

Digital signature which is usually used in many kinds of digital documents and only generated by the original signer in this digital age may result in several inconveniences. So, in order to solve the dilemma that digital signatures can't be generated without the original signer, some scholars proposed a new scheme called Proxy Signature. By the showing up of Proxy signature scheme, the original signer could designate a proxy to perform the signing by generating valid digital signatures on behalf of the original signer when it is not there. However, there is no suitable scheme which could be applied to the Ad-hoc network environment due to its loose structure and unstable topology of mobile nodes. In order to find out a new scheme that could be used in Ad-hoc network environment, I tried to collect and analysis advantages and disadvantages of every related scheme, for example, I try to incorporate the secret sharing mechanism of Threshold Proxy Signature into my new mechanism, besides, I also benefit from using the concept of El-Gamal to improve the security to make a more suited distributed nominative proxy signature mechanism for Ad-hoc network. At last, we will show better security and flexibility the new mechanism has by Manik Lal Das's basic proxy signature scheme security requirements and Kaliskis' methodology to perform computing time analysis.

Keywords: Ad-hoc network, proxy signature, nominative proxy signature, threshold proxy signature

誌謝

謝謝羅濟群教授的指導，以及俊傑學長在論文上的諸多幫忙與指點，要謝的實在太多了，不如就謝天吧。

目錄

圖目錄

表目錄

第一章 緒論

在此章中，我們會說明本論文的研究背景、動機、研究目的，以及後續各章節的大略介紹。

1.1 研究背景與動機

隨著科技不斷的發展與進步，無線網路變得越來越普及，頻寬與連線速度也越來越快，而以往受傳統網路架構的限制，使用者只能在某一特定範圍內連上網路，這對資訊的取得是一個十分嚴重的限制。因此，建立一個可以隨時隨地上網的無線網路環境，的確是十分迫切的需要。

在無線網路中，有兩種主要的運作方式，一種是稱為有基礎架構模式（Infrastructure Based Operation Mode），也就是目前最廣為公司行號或家庭私人使用的方式。透過無線基地台（Access Point），各種 Mobile Device 可以在它的涵蓋範圍內互相通訊或連接上 Internet。另一種方式則稱之為無基礎架構操作模式（Infrastructureless Based Operation Mode），又稱為 Ad-hoc network。此架構下的節點並不利用 Access Point 來做溝通，它們彼此以某種路由方式來形成拓樸，相較於有基礎架構模式，Ad-hoc network 其可動性（Mobility）更佳，更具有彈性。但由於缺少了 Access Point 的集中管制，路由的形成與節點之間關係的建構十分複雜。

另一方面，自從網路開始發展以來，資訊安全便一直是網路使用者最頭疼的問題。無線網路基於其本身無線通訊的特質，安全性更是較有線網路顯得脆弱許多；此外，許多在有線網路使用無礙的資訊安全方法或技巧，也可能因為無線網路本身的一些特性或限制而顯得滯礙難行。舉例來說，目前許多廣被政府機關或在電子交易上使用的電子簽章或自然人憑證，在 Ad-hoc network 這種

拓樸多變的網路環境下便顯得不太適用。但許多時候因為時間或空間上的限制，我們或許只能採用 Ad-hoc network 模式的網路架構來連繫（例如南亞大海嘯）。因此，如何找出一個合理且具備彈性之電子簽章機制則成為本論文之主要動機。

1.2 研究目的

目前世面上已存有許多的電子簽章技術，但由於 Ad-hoc network 本身成員變動十分頻繁，因此一般的電子簽章技術在 Ad-hoc network 的環境下便顯得不那麼適用。舉例來說，如 Ad-hoc network 中某一 group 中的某一 node 因為通訊鏈路的問題而暫時無法與其它 node 連繫上，或是因為某些原因必須先離開，但該 node 本身又負有重大責任或扮演著重要的角色，則它勢必要找人來代理它目前的職務。因此，為配合 Ad-hoc network 多變的網路狀態，代理簽章（Proxy signature）制度似乎是解決方案的不二人選。

在代理簽章其實已經有許多的方法和技術提出，但沒有一個是基於 Ad-hoc network 本身的特質來量身訂做。本研究的目的，正是要研究目前已存在的代理簽章方法，並從中去蕪存菁，或加入許的概念，來為 Ad-hoc network 打造出一個更合用的簽章方式。除此之外，對於訊息的接收者，如何分辨簽章的真偽以及來源，或是由誰來簽署等等電子簽章中所會面臨的問題，也一併在本研究中來做探討。

1.3 章節介紹

在第二章我們將藉由文獻探討，來熟悉無線網路（Infrastructure based 及 Ad-hoc network）架構上的差別，並了解並回顧截至目前為止發展的各類型代理簽章方式；第三章則先說明本論文所引用做為根基的舊方法以及加入新概念的新方法各別的演算步驟；第四章則對新舊方法的差異及耗用的計算複雜度做比較圖表分析；第五章則是針對本系統討論未來可研究的方向。

第二章　文獻探討

在本章中，主要是介紹與說明與本論文相關的一些研究，包含無線網路和目前的一些數位簽章技術。

2.1　無線網路

2.1.1　何謂無線網路[1]

無線區域網路的技術有兩大分類

表 2- 1 無線網路技術的主要分類

利用無線電	光傳導
IEEE802.11 、HomeRF 以及藍芽技術	紅外線(Infrared)與雷射光(Laser)作為資料傳輸的載波(Carrier)

1985 年，美國聯邦通訊委員會（FCC；Federal Communications Commission）決定開放三個 ISM 頻帶（Industrial Scientific Medical bands），即 902~928MHz，2.4~2.483GHz、5.725~5.875GHz 等三個頻帶。此一動作不僅滿足了當時對通訊頻帶日益增加的需求，對於無線網路發展更有著重要的影響。到了 90 年代初，使用 ISM 頻帶的通訊產品紛紛出現在市場上，為了使各種競爭的產品之間能夠互通，標準的制訂就成了重要的工作，而後便有 IEEE 802.11 無線區域網路（wireless LAN）的標準產生。

IEEE 802.11 主要的目的是要制訂一套適合在無線網路環境下作業的通訊協定，最重要的工作，就是要制訂出 MAC 層（Media access control sublayer）和實體層。因此 IEEE 802.11 的參考模式主要分成三個部份；第一部份定義適

[1]何謂無線網路：http://www.lib.ncu.edu.tw/wireintro.htm

用於所有無線網路系統的 MAC 規格；第二部份制訂和傳輸媒介相關的 PHY 規格；第三部份則是說明 power saving functionality 的部份。

為了要達到無線網路的透明化，無線區域網路希望做到在邏輯鍵結層（LLC）就能和別的網路相通，這使得無線區域網路必須將處理移動性收發站及保持資料傳輸可靠性的能力全部做在 MAC 層中，這和傳統有線網路在 MAC 所需具有的功能是不同的。此外，針對三種不同的 ISM 頻帶，也都有不同的 PHY 規格

IEEE 802.11 無線網路的主要特性如下：

- 傳輸速率最低為 1Mbps。
- 傳輸媒介為無線電波。
- 通訊協定為 CSMA/CA，提供優先權服務。
- 訊框為 IEEE 802.11 CSMA/CA 訊框。
- 提供保證傳送延遲服務。如果同時有兩個或兩個以上的工作站同時傳送訊框，則會發生衝撞並將訊框視為無效且丟棄。而使用 CSMA/CA 可避免大部份不必要的衝撞，因此可提供保證傳送延遲的服務。
- 頻寬使用不保證公平。每個工作站實際使用的頻寬量可能不同。
- 較不適合多媒體資訊傳輸。雖然提供保證傳送延遲服務，但 1~2Mbps 尚不足以應付具有及時要求的多媒體資訊。

2.1.2 Ad-hoc Networking

IEEE 802.11 制訂了兩種不同類型的無線區域網路架構：

有基礎架構之無線區域網路（Infrastructure Wireless LAN），如圖 2-1 所示

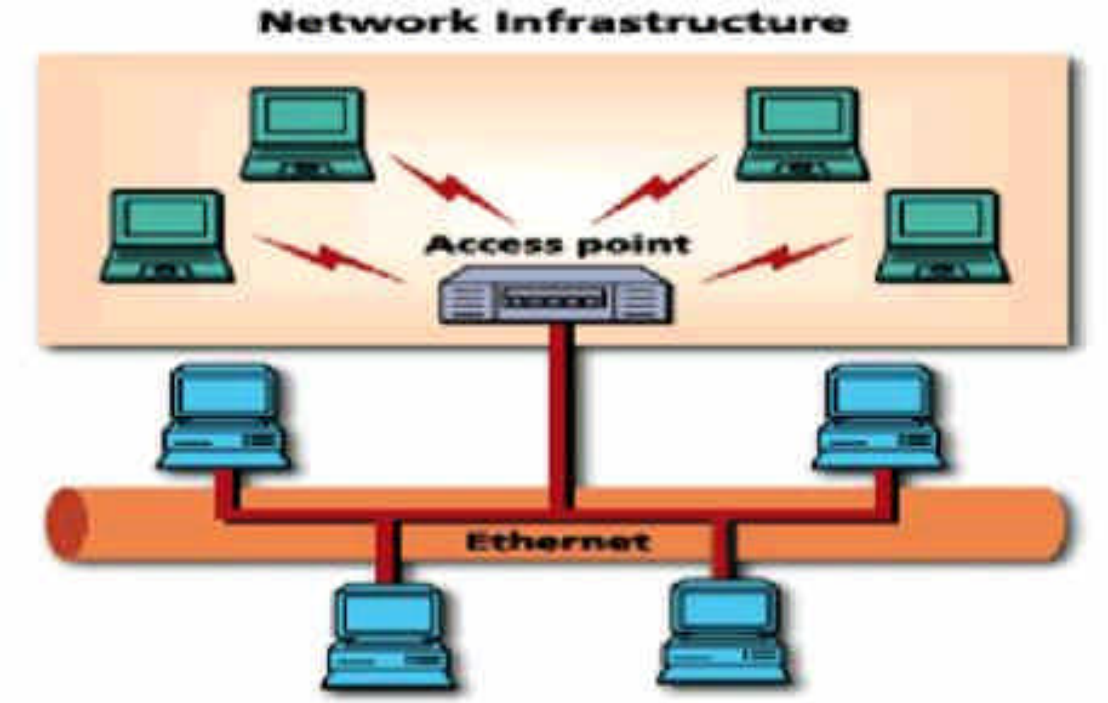


圖 2- 1 有基礎架構之無線區域網路[2]

由上圖可清楚看出，在這種架構下的所有 node，均需透過中央控管的 Access Point 來做網路的存取，而 Access Point 最主要的功能也就是將各個 mobile node 所傳需給它的 data frame 做 relay 的動作，讓一個或多個有線及無線的網路區段能互通有無。此外，由於有 Access Point 做中央的控管，mobile node 便不需要自己做通路鏈結及網路拓樸的管理，省事許多。

無基礎架構之無線區域網路（Ad Hoc Wireless LAN），如圖 2-2 所示

[2]何謂無線區域網路：http://www.npic.edu.tw/CC/wlan/1.htm

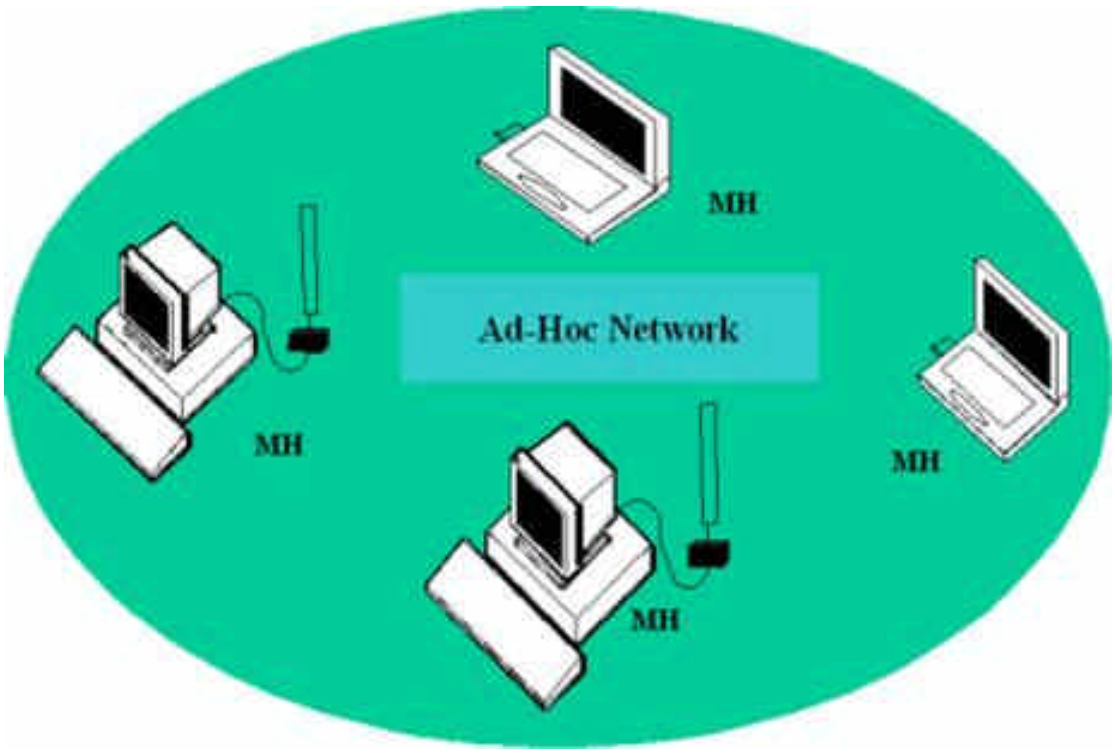


圖 2- 2 無基礎架構之無線區域網路

相較於 Infrastructure based WLAN，Ad-hoc network 少了 Access point 的集中管理，沒有什麼限制，隨時隨地都能架起無線通信的網路，且網路中任兩個 node 亦可以直接溝通而不需假手他人（這是在若對方是在一個 hop 的範圍之內可以連繫上，若不行則還是得經過路徑或拓樸的計算，藉由其他 node 來幫忙傳遞 data frame），這一類的無線網路多用在即時的應用，例如會議室開會；或是先天地形上有限制的區域，沒辦法期待有任何無線網路基礎建設的地方，如戰場，沙漠或是山區等等。

由於本論文是專注探討 Ad-hoc network 下的代理簽章，以下僅針對 Ad-hoc network 做更深一步的解釋

2.2 代理簽章機制

數位簽章就像是在電子形式文件上的親筆簽名一樣，是一種無法被偽造的資料，它代表了發訊者製作或至少同意了附有其數位簽章的電子文件。數位簽章比起手寫的簽名字跡提供更高的安全。收到具有數位簽章訊息的人也可以用簽章來得知發訊者的身分，進而確認訊息的內容在經過簽章後是否遭受故意或是意外的竄改。另外，由於數位簽章具有不得否認的特性，所以若使用者在訊息上使用了數位簽章，則他將無法狡辯其簽章遭到偽造而逃避曾經撰寫過此訊息的事實。簡單的說，數位簽章在數位訊息中所扮演的角色為 "鑑別" 。它可以讓收訊者對於訊息的來源及其完整性更有信心。

簽章機制一般常見的有數位簽章、盲目簽章、群體簽章及代理簽章。不管是哪一種簽章機制都有相同之處，即是必須達到簽章者的不可否認性，以及至少需要有兩個個體會參與，一個是文件簽署者(Signer)而另一個是驗證者(Verifier)。其中驗證者必須藉由簽署者的公開訊息經過一系列運算後比對是否相等，若相等，則相信簽署文件的合法性，並確定由誰簽署此份文件。一般而言，數位簽章可區分為確定式數位簽章機制，例如 RSA 及 Rabin，和機率式的數位簽章機制，例如 ElGamal 數位簽章。

所謂的盲目簽章指的是簽署人對簽署文件的內容是盲目的，且即使日後公怖此文件及文件簽名之資訊，簽署人也無從追蹤文件與當時盲簽名的相互關係，亦即是說，盲目簽章必須滿足不可追蹤的性質。一般常將此特性應用於電子投票及電子現金的系統上。而所謂的群體簽章，指的是一群人同時對一份文件作簽署，最早提出此方式的由 Boyd 在 1986 年提出(n, n)團體式簽名法，但由於這個方式不切實際，所以後人加以修正。

而所謂代理簽章機制，它是一種特殊的數位簽章機制，亦即是說，簽署者可委託某一個人，代理它完成文件的簽署，且驗證者能驗證此簽章之有效性與

來源性，其架構如圖 2-3。此機制非常適用於無基礎行動網路環境，以下即就代理簽章做詳細說明。

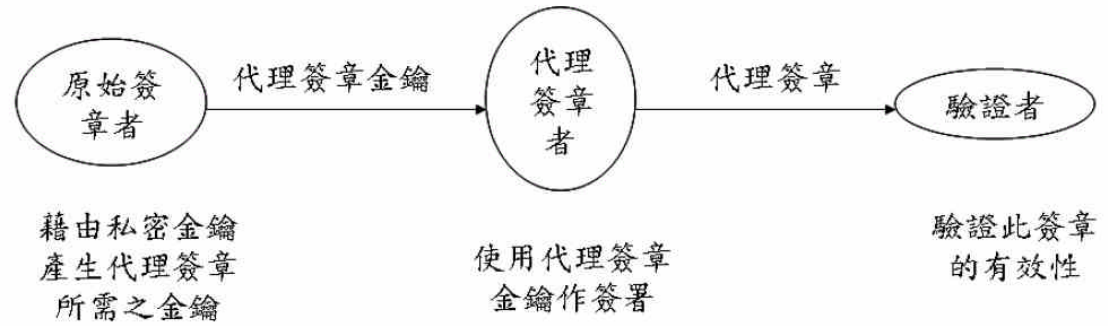


圖 2- 3 代理簽章(完全授權)架構

最早討論代理簽章是由 Mambo 等學者於 1996 年所提出，他們所提的概念是原始簽章者可以指派一個代理人來代為簽章，而此代理人所簽署文件的效力就等同原始簽章者一樣，但由於他們所提的機制無法解決不可否認性，亦即是說，代理者可以否認曾經簽署過此份文件，故不適用。所以，後續有很多學者就此問題提出解決方案。下面將介紹代理簽章的特性及其種類。一般而言，簽章所代表的含意是它應具有不可偽造性及可驗證性，也因為有此特性方能達到不可否認性。而代理簽章機制亦是如此。以下即就代理簽章的特性做介紹：

- 可區別性：即是驗證者它可以驗證被簽署之文件，到底是由原始簽章者所簽署或是由代理簽章者所簽署；若為多重代理簽章，除了需符合上述之要求外，還需要分辨是由那位代理簽章者所簽署。
- 不可偽造性：只有原始簽章者所指派代理簽章者，才可產生有效的代理簽章文件，沒有任何人可偽造此代理簽章。
- 可驗證性：驗證者可以從代理簽章者所送過來的公開訊息中驗證此簽章之合法性，且需能驗證此簽署文件是由原始簽章者所授權。
- 識別性：從代理簽章資訊，原始簽章者可識別代理簽章者的身份。
- 不可否認性：其效果就和一般數位簽章一樣，即代理簽章者不可否認曾經簽署過此份文件。

- 公平性：代理簽章者不能將代理簽章的權力轉移給它人使用，否則對原始簽章者是不公平。

而原始簽章者授予代理簽章者之授權方式大致而言，可區分下述幾種：

完全授權：所謂完全授權可以想成代理簽章者就是原始簽章者，亦即是說原始簽章者將其簽章所需的私密金鑰交給代理簽章者，此時代理簽章者就如同它的分身。此種授權方式非常危險，因為代理簽章者擁有它的私密金鑰，所以它可以隨意簽署任何文件，而原始簽章者無法約束，且驗證者無法從所簽署之文件辨別是由原始簽章者所簽署或是由代理簽章者所簽署，故無法滿足上述之特性。

部份授權：部份授權就是要解決上述完全授權的缺點，換句話說，原始簽章者授予代理簽章者權力時，並非將個人的私密金鑰直接送給代理簽章者，而是利用個人的私密金鑰經過一系列的計算而產生代理金鑰，然後再將此代理金鑰送給代理簽章者。因此，代理簽章者可利用此金鑰產生代理簽章。由於原始簽章者所持的金鑰與代理簽章者所持代理金鑰不一樣，因此，驗證者可以區別此簽署文件是由誰所簽署。而在此授權模式下，又可區分成兩種狀況：第一種，代理簽章者直接使用原始簽章者所產生代理金鑰，然後對文件作簽署，此種方式雖可區別原始簽章與代理簽章之不同，但對代理簽章者而言並不公平，因為假若原始簽章者利用此代理金鑰對文件簽署，對驗證者而言，它會認定此文件是由代理簽章者所簽署，所以代理簽章者必須概括承受，故有第二種方式產生。第二種方式，即代理簽章者收到原始簽章者送過來的代理金鑰，並非直接使用它，而是將它與自己的私密金鑰再經過一系列的計算，以產生新的代理金鑰，當有文件需要簽署時，則使用新的代理金鑰加以簽署，當然此架構仍需滿足代理簽章機制之要求，但它也解決了公平性問題，亦即是說，原始簽章者由於沒

有代理簽章者之代理金鑰，故無法假冒代理簽章者對文件作簽署。

授權憑證：此種授權方式即由原始簽章者產生一個授權憑證給代理簽章者，而此授權憑證乃利用原始簽章者之私密金鑰簽署後產生，此憑證內容除了包括代理簽章者有權力代為執行簽章外，還需包括代理權限與期限及可簽署文件的類型等等。代理簽章者拿到此憑證之後，即可利用自己的私密金鑰簽署所代簽之文件，並將憑證一起送給接收者。而驗證者它除了需驗證此簽署文件的正確性外，還需檢查此憑證之合法性，以確保是否由原始簽章者授權給此代理簽章者簽署此份文件。

結合授權憑證之部份授權：此授權機制是將第二種與第三種的授權機制結合起來。原始簽章者先設定代理簽章者所應具有的權力，例如上述的代理權限與期限、可簽署文件的類型等等，然後再利用自己的私密金鑰連同所規範的簽署權力加以計算，再將結果送至代理簽章者。代理簽章者收到此訊息後，即利用此訊息與自己的私密金鑰加以計算，以產生代理金鑰，而代理簽章者所簽署的文件中亦須包含原始簽署者所規定的簽署權力，因此，只有符合此簽署權力的簽署文件才具有合法性。

2.3 提名式代理簽章

提名式代理簽章(Nominative proxy signature)與一般的代理簽章最不同的地方，在於一般的簽章在訊息接收者在收到被簽署的文件後，可以透過 CA 或是依靠自身的計算來驗證該簽名是否有效，是否來自原簽署者(Original signer)等等；而提名式代理簽章則必須透過第三者，亦即被提名者(Nominated verifier or Nominee)的幫忙，才可以驗證簽章是否有效，訊息接收者是不能自己驗算的。

而依據驗證者被提名的方式，提名式代理簽章可分為－原始簽章者提名式(Original-nominative proxy signature)，其驗證者是由原始簽章者所提名。另一種則是由代理簽章者來提名驗證者(Proxy-nominative proxy signature)，由代理簽章者來負責提名驗證者。

事實上提名式代理簽章十分適合用在行動通訊(Mobile communication)的環境。在行動通訊中，使用者(user)就像是原始簽章者，代理者(agent entity)則像是代理簽章者，由於提名式代理簽章只能由被提名者來驗證，因此使用者及代理者的身份都可以被保密(anonymity guaranteed)。除了保密性的優點，一般的行動裝置本身的計算能力及電力都較為缺乏，透過代理簽章的方式，原始簽章者就可以把簽章原本要執行的大量計算，例如大量的模數與指數計算，交付給代理簽章者來做，以節省電力。一般來說，original-nominative proxy signature 較適合訊息接收者是由使用者決定的無線通訊環境，而另一方面，proxy-nominative proxy signature 則較適合用於行動電子商務(Mobile electronic commerce)，在 e-commerce 中，制造商就像是原始簽章者的角色，它負責提供有品質保障的服務台或產品，但它沒必要參與所有的販賣行為，可以交給批發商或零售商。

提名式代理簽章必須滿足下列四點

1. 只有原始簽章者(或代理簽章者)，可以提名驗證者
2. 原始簽章者與代理簽章者否認他們所產生的簽章
3. 只有被提名者可以直接驗證代理簽章是否有效
4. 如果有必要，只有被提名者可以向第三方證明某個簽章是否有效

2.4 Threshold Proxy Signature

Threshold proxy signature 是 proxy signature 一個變種。舉例來說，一個(t,n)其 proxy signature key 是由 n 個 proxy signer 所組成的 proxy group 所擁有，只要這個 proxy group 中有 t 個以上的人願意合作，它們就可以一起產生一個 threshold proxy signature；反過來說，如果沒辦法找到至少 t 個 proxy signer，就無法產生有效的簽章，就算只少一個也不行

對 Threshold proxy signature 來說，其不可否認性(nonrepudiation)最主要是指一但一個有效的簽章產生了，則負責產生這個簽章的 proxy signer group 就不可否認這個簽章是由它們合作產出的。

Threshold proxy signature 的概念在本論文中被大量採用，它最主要的技巧是利用了 Lagrange interpolation polynominal 來做 secret sharing

2.4.1 Verifiable Secret Sharing Scheme

(t,n) secret sharing scheme 是一種將 secret 拆成 n 份將由 n 個不同的人，並由至少 t 個以上的人通力合作才能重建該 secret 的技術。Verifiable secret sharing scheme 則是其中的一種，其重建回來的 secret 可以被驗證是否為當初有效的 secret：

p: a large prime

q: a prime factor of $p-1$

g: an element of order q in Z_p^*

1. 假設有 $p_1, p_2, p_3 ... p_n$ 個 signer 形成一個 proxy signer group，其中每個 proxy p_i 都有自己的多項式 $f_i = s_i + a_{i,1}x + a_{i,2}x^2 + ... + a_{i,t-1}x^{t-1} \bmod q$ of

degree $t-1$ in Z_q

2. p_i 會以安全保密的方式送出 $f_i(j) \bmod q$ 給 $p_j(\forall 1 \le j \le n, j \ne i)$，並且廣播各個參數如 $g^{s_i}, g^{a_{i,1}}, g^{a_{i,2}}, ..., g^{a_{i,t-1}}$

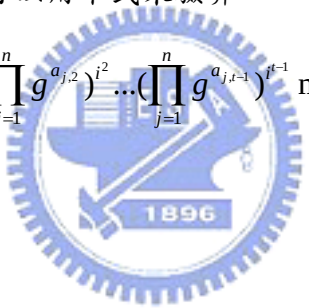
3. p_i 在收到其它 p_j 送給它的 $f_j(i)$ 時，會做下式驗算，看是否相符：

$g^{f_j(i)} = g^{s_j}(g^{a_{j,1}})^i(g^{a_{j,2}})^{i^2}...(g^{a_{j,t-1}})^{i^{t-1}} \bmod p$，如果所收到的 $f_j(i)$ 都符合，則 p_i 便會計算 $s_i^{'} = \sum_{j=1}^{n} f_j(i)$，當做它自己的 share

4. 假設

$f(x) = \sum_{j=1}^{n} f_j(x) = \sum_{j=1}^{n} s_j + (\sum_{j=1}^{n} a_{j,1})x + (\sum_{j=1}^{n} a_{j,2})x^2 + ... + (\sum_{j=1}^{n} a_{j,t-1})x^{t-1} \bmod q$，則 $s_i^{'} = f(i)$，且每個 $s_i^{'}$ 都可以用下式來驗算：

$g^{s_i^{'}} = (\prod_{j=1}^{n} g^{s_j})(\prod_{j=1}^{n} g^{a_{j,1}})^i(\prod_{j=1}^{n} g^{a_{j,2}})^{i^2}...(\prod_{j=1}^{n} g^{a_{j,t-1}})^{i^{t-1}} \bmod p$



只要有任何 t 以上的 share，$s_{i_1}^{'}, s_{i_2}^{'}, s_{i_3}^{'}, ..., s_{i_t}^{'}$，$\{i_1, ...i_t\} \subset \{1, ..., n\}$，則 secret $\sum_{j=1}^{n} s_j$ 可以很容易以 Lagrange interpolating polynomial 計算出：

$\sum_{j=1}^{n} s_j = f(0) = \sum_{k=1}^{t} s_{i_k}^{'} \prod_{j=1, j\ne k}^{t} \frac{(0-i_j)}{(i_k - i_j)} \bmod q$，其有效性也可以由 $g^{\sum s_j} = \prod_{j=1}^{n} g^{s_j}$ 此式是否成立來判斷

2.4.2 Conventional threshold signature

事實上許多現在的 threshold proxy signature 都是利用上述的 verifiable

secret sharing scheme 來達到 share 的產生，簡單介紹過程如下：

1. Share generation： p 、q 、g 等參數都和 2.4.1 一樣，且每個 p_i 都有自己的 private key $x_i \in Z_q$ 及 public key $y_i = g^{x_i} \bmod p$，則該 group 的 secret key $x = \sum_{i=1}^{n} x_i \bmod p$ ，public key $y = \prod_{i=1}^{n} y_i = g^{\sum_{i=1}^{n} x_i \bmod p}$

2. 一但該 proxy signer group 中有 t 個以上的人願意合作來為一篇文章簽名，則每個 p_i 會用自己的 share 來產生 partial signature，而只要將 t 份 partial signature 結合在一起，就可以做出該篇文章真正的簽章。

2.5 Zuo-Wen Tan 與 Zhuo-Jun Liu 提名式代理簽章

本論文在演算法的設計上是架構在這兩人所提出的方法，並在參數的組合上融合了門檻式代理簽章的概念。在此先介紹兩位學者的方式。

總共分為三個階段：

[Preliminary]

p : a large prime

q : a prime factor of $p-1$

$H()$: a public one-way function

T : a time stamp

M : the message

A: Original signer

B: Proxy signer

C: Receiver

A、B、C have their own private key x and public key $y = g^x \bmod p$

[Delegation Phase]

[Proxy Generation]

A 首先產生一個 warrant m_w，其中記錄了代理的範圍和限制，代理期限，以及 A 和 B 的識別，之後 A 選取一個數 $k \in_R Z_q^*$，然後計算：

$$r = g^k \bmod p, s_A = x_A \times H(m_w \| T \| r \| y_C) + k \bmod q$$

計算完後，A 會將 (m_w, T, r, y_C, s_A) 送給 B

[Delegation Verification]

B 在收到上面那串東西後，會計算下列式子是否成立：

$g^{s_A} = r y_A^{H(m_w \| T \| r \| y_C)} \bmod p$，如果成立，則繼續往下一步，否則 B 會拒絕該要求

[Proxy Signature Key Generation]

B 計算下式做為 proxy signature key：

$$s_p = s_A + x_B \cdot H(m_w \| T \| r \| y_C) \bmod p$$

[Proxy Signature Generation Phase]

B 先隨機選出兩個數 $k_1, k_2 \in_R Z_q^*$，並且計算下式：

$$R = g^{k_1 - k_2} \bmod p, Z = y_C^{k_1} \bmod p$$
$$e = H(M \| m_w \| y_C \| R \| Z), s = k_2 - e \cdot s_p \bmod q$$

然後 B 會計算出 nominative proxy signature $(M, m_w, T, y_C, r, R, Z, s)$，並送給 C

[Verification Phase]

C 首先會確認收到的訊息 M 與 m_w 的資訊是否吻合，若沒有問題則計算出 proxy signature public key $y_p = g^{s_p} = r(y_A y_B)^{H(m_w \| T \| r \| y_C)} \bmod p$ ，然後計算下式來確認 nominative proxy signature 的合法性： $(g^s \cdot y_p^e \cdot R)^{x_C} = Z \bmod p$, where $e = H(M \| m_w \| T \| r \| y_C \| R)$ ；若上式等號成立，則代表簽章有效。

則下來將介紹本篇論文用來評估與比較新舊方法的準則

2.6 Kaliski 評估法

Kaliski 學者針對數位簽章計算過程所需花費的計算量提出了一個方法：

$WM(b)$ ：執行 b-bit 模數乘法所需耗費的計算量

$WS(b)$ ：執行 b-bit 平方所需耗費的計算量

而基本的計算法則如下：

1. $WS(b) = 0.75WM(b)$
2. $\frac{WM(b_1)}{b_1^2} = \frac{WM(b_2)}{b_2^2}$

依據上述法則，可以試著找出一個執行指數次方與模數計算所需的計算工作量。例如： $g^s \bmod p$ ，其所需的計算量，W ，為 $|s|WS(|p|) + 0.5|s|WM(|p|)$ ，再搭配上述基本法則後，可得到 $W = 640WM(512)$ (其中 s, p 都是 512-bit)

以下提供一些經由 Kaliski 方法所計算出來的比較結果

$WI(b)$: 執行 b-bit modular inversion 所需要的計算量

$WH(b)$: 執行一個以 b-bit 做為輸入的 hash function 所需的工作量

m_w: Original signer 所做出的 warrant

m_p: 經由 proxy signer 所簽署的文件

(縮寫)

Com: Computation

cre.: creation

P.C.: Proxy Creation

Sig. cre., S.C.: Signature creation

Ver.: Verfication

		ElGamal		
		Original (or Proxy, Full)	Proxy	
			Partial	Warrant
Total length		$1024+\|m_p\|$	$1536+\|m_p\|$	$2048+\|m_p\|+\|m_w\|$
Com.	Proxy cre.	–	641	$642+WI(512)$
	Sig. cre.	$642+WI(512)$	$642+WI(512)$	$642+WI(512)$
	Ver.	1921	2562	3842
	Total	$2563+WI(512)$	$3845+WI(512)$	$5126+2WI(512)$

圖 2- 4 El-Gamal 的執行效率

		Schnorr		
		Original (or Proxy, Full)	Proxy	
			Partial	Warrant
Total length		$288+\|m_p\|$	$736+\|m_p\|$	$736+\|m_p\|+\|m_w\|$
Com.	P.C.	–	175	$175+WH(\|m_w\|)$
	S.C.	$175+WH(\|m_p\|)$	$175+WH(\|m_p\|)$	$175+WH(\|m_p\|)$
	Ver.	$336+WH(\|m_p\|)$	$512+WH(\|m_p\|)$	$672+WH(\|m_p\|)+WH(\|m_w\|)$
	Total	$511+2WH(\|m_p\|)$	$862+2WH(\|m_p\|)$	$1022+2WH(\|m_p\|)+2WH(\|m_w\|)$

圖 2- 5 Schnorr scheme 的執行效率

		Okamoto		
		Original (or Proxy, Full)	Proxy	
			Partial	Warrant
Total length		$408+\|m_p\|$	$920+\|m_p\|$	$816+\|m_p\|+\|m_w\|$
Com.	P.C.	–	351	$351+WH(\|m_w\|)$
	S.C.	$351+WH(\|m_p\|)$	$351+WH(\|m_p\|)$	$351+WH(\|m_p\|)$
	Ver.	$512+WH(\|m_p\|)$	$688+WH(\|m_p\|)$	$1024+WH(\|m_p\|)+WH(\|m_w\|)$
	Total	$863+2WH(\|m_p\|)$	$1390+2WH(\|m_p\|)$	$1726+2WH(\|m_p\|)+2WH(\|m_w\|)$

圖 2- 6 Okamoto scheme 的執行效率

第三章 分散式提名代理簽章

在第一章我們已經充份了解本研究的動機以及進行的方向。而在第二章，我們針對了無線網路的特性和目前現有的簽章技術和將用來做為衡量基準的工具都做了一個初步的介紹。接下來在此章中，首先會先就問題就更進一步的討探，為何現有的方法沒辦法用在無線網路上，而接著會進一步來詳細介紹本論文所提出的新演算法。

3.1 問題分析

目前現有的簽章技術五花八門，每一種方法都有其獨特的優點與缺點。但尤於這些方法在被提出來的時候，無線網路並不那麼熱門，也因此這些方法並沒有考量到無線網路比起有線網路還複雜多的拓樸和移動性等等狀況。除了無線網路的移動性外，無線裝置的電力和網路本身的寬頻，比起有線網路來說都顯得缺乏許多。

因此，本論文打算藉由整理融合了各家簽章方法的優點，並針對無線網路的特性來提出一個較為合用的演算法。

3.2 分散式提名代理簽章機制

在 2.3，我們提到了提名式代理簽章，也在裡頭說明了為何它較適合用於無線網路的環境。

而在 2.4，我們提到了 Threshold proxy signature，而它分散式的處理方式，

正巧符合了無線裝置本身運算和電力都較為薄弱的窘境。因此，本篇論文所提出的方法，將會大量採用 Threshold proxy signature 的概念，來將計算量分擔出去。

我們也在 2.5 時對 Zuo-Wen Tan 與 Zhuo-Jun Liu 兩人所提出的演算法做了描述。而本篇論文的演算方法就是以此兩人的方法為基礎，並結合了分散運算的觀念。

最後在 2.6 述說了 Kaliski 的計算需求評估法，將用於新舊演算法的計量。

3.2.1 演算法

演算法可分為五個階段：

1 Proxy group share generation: 此一階段，proxy group 成員將會透過 secret sharing 的技巧來合力創造出這個 proxy group 的 group public key 與 group private key

2 Verifier group share generation: 同上，verifier group 也將用同樣的技巧來做出該 verifier group 的 group public key 與 group private key

3 Proxy key generation: 在此一步驟中，original signer 首先做出一支 proxy key，交給該 proxy signer group，而 proxy signer group 在接收 key 後，再經過變換改成自己的 proxy signer key，以防止 original 的偽造

4 Proxy signature generation and issuing: proxy signer group 透過合作的方式為 original signer 代簽，其中至少要有 t 個以上的 proxy signer 願意合作才有可能成功簽署文章

5 Verification:同上，verifier group 透過合作的方式來為第三方驗證簽名是否有效，其中至少要有 l 個以上的 verifier 願意合作才能驗證簽名真偽

表 3- 1 符號對照表

p	a large prime
q	a large prime factor of $p-1$
g	generator, its order is q
O	Original signer
PG	Proxy signers; where $P=\{p_1, p_2, ..., p_n\}$
VG	Verifier group; where $V=\{v_1, v_2, ..., v_m\}$
$h()$	A one-way hash function

以下是關於本論文演算法的詳細介紹

[Initial]

Each of them has public and private key pairs, for

$O:(x_o, y_o)$

$p_i : (x_{p_i}, y_{p_i}); \forall i = 1, 2, ..., n$

$v_j : (x_{v_j}, y_{v_j}); \forall j = 1, ..., m$

[Phase 1 - Proxy group share generation]

1. Each p_i chooses the random numbers $a_{p_i}, b_{p_i} \in_R Z_q^*$; and computes

$$r_{p_i} = g^{a_{p_i}} \bmod p$$
$$b_{p_i} \equiv x_{p_i} \cdot r_{p_i} + a_{p_i} \cdot c_{p_i} \bmod q$$
$$\Rightarrow y'_{p_i} = g^{b_{p_i}} = y_{p_i}^{r_{p_i}} \cdot r_{p_i}^{c_{p_i}} \bmod p$$

2. p_i publishes $Sign(h(b_{p_i}), r_{p_i}, c_{p_i})$ as its signature and computes the following polynomial: $f_i(z) = b_{p_i} + e_{i,1}z + e_{i,2}z^2 + \cdots + e_{i,t-1}z^{t-1} \bmod q$, then computes $f_i(j) = b_{p_i} + e_{i,1}j + e_{i,2}j^2 + \cdots + e_{i,t-1}j^{t-1} \bmod q; \forall j = 1, ..., n; j \neq i$; sends $f_i(j)$ to the proxy signer p_j by the secure channel, and

broadcasts $g^{b_{p_i}}, g^{e_{i,1}}, ..., g^{e_{i,t-1}} (\forall j = 1,2,...,n; and\ j \neq i)$. On the same way, all the other p_j do the same thing. After that, p_i can verify the validity of $f_j(i)$ by checking the following equation: $g^{f_j(i)} = g^{b_{p_j}} \cdot (g^{e_{j,1}})^i \cdot (g^{e_{j,2}})^{i^2} \cdots (g^{e_{j,t-1}})^{i^{t-1}} \bmod p = g^{b_{p_j}} \cdot A_{j,1}^i \cdot A_{j,2}^{i^2} \cdots A_{j,t-1}^{i^{t-1}} \bmod p$; where $A_{j,k} = g^{e_{j,k}}; \forall k = 1,...,t-1$

3. If all $f_j(i)$ s are hold, then p_i computes: $s_{p_i} = \sum_{j=1}^{n} f_j(i)$ as its share.

4. Without lost of generality, let the following polynomial:

$f(z) = b_0 + e_1 z + e_2 z^2 + \cdots + e_{t-1} z^{t-1} \bmod q = \sum_{j=1}^{n} f_j(z) \bmod q; where b_0 = \sum_{j=1}^{n} b_{p_j}$

$\Rightarrow f(z) = \sum_{j=1}^{n} b_{p_j} + (\sum_{j=1}^{n} e_{j,1}) z + (\sum_{j=1}^{n} e_{j,2}) z^2 + \cdots + (\sum_{j=1}^{n} e_{j,t-1}) z^{t-1} \bmod q$

$\Rightarrow s_{p_i} = f(i) = \sum_{j=1}^{n} f_j(i) \bmod q$

$y_{PG} = g^{b_0} \bmod p = g^{\sum_{j=1}^{n} b_{p_j}} \bmod p = \prod_{j=1}^{n} g^{b_{p_j}} \bmod p = \prod_{j=1}^{n} y_{p_j}' \bmod p$

we define y_{PG} as the public key of the proxy group, and b_0 is the private key of the proxy group.

[Phase 2 - Verifier group share generation]

1. Each v_j chooses the random numbers $a_{v_j}', b_{v_j}' \in_R Z_q^*$; and it computes

$r_{v_j}' = g^{a_{v_j}'} \bmod p$

$b_{v_j}' \equiv x_{v_j} \cdot r_{v_j}' + a_{v_j}' \cdot c_{v_j}' \bmod q$

$\Rightarrow y_{v_j}'' = g^{b_{v_j}'} = y_{v_j}^{r_{v_j}'} \cdot (r_{v_j}')^{c_{v_j}'} \bmod p$

2. v_j publishes $Sign(h(b_{v_j}'), r_{v_j}', c_{v_j}')$ as its signature and does the following polynomial: $\psi_j(z) = b_{v_j}' + e_{j,1} z + e_{j,2} z^2 + \cdots + e_{j,l-1} z^{l-1} \bmod q$, and computes $\psi_j(i) = b_{v_j}' + e_{j,1} i + e_{j,2} i^2 + \cdots + e_{j,l-1} i^{l-1} \bmod q; \forall i = 1,...,m; i \neq j$; sends $\psi_j(i)$ to v_i by

the secure channel, and broadcasts $g^{b'_{v_j}}, g^{e_{j,1}}, ..., g^{e_{j,l-1}} (\forall i = 1,2,...,m; and\, i \neq j)$. On the same way, all the other v_i do the same thing. After that, v_j can verify the validity of $\psi_i(j)$ s by checking the following equation:

$$g^{\psi_i(j)} = g^{b'_{v_i}} \cdot (g^{e_{i,1}})^j \cdot (g^{e_{i,2}})^{j^2} \cdots (g^{e_{i,l-1}})^{j^{l-1}} \bmod p$$

3. If all $\psi_i(j)$ s are hold, then v_j computes: $s_{v_j} = \sum_{i=1}^{m} \psi_i(j)$ as its share.

4. Without lost of generality, we define the following polynomial

$$\psi(z) = b'_0 + e_1 z + e_2 z^2 + \cdots + e_{l-1} z^{l-1} \bmod q = \sum_{i=1}^{m} \psi_i(z) \bmod q; where b'_0 = \sum_{i=1}^{m} b'_{v_i}$$
$$\Rightarrow \psi(z) = \sum_{i=1}^{m} b'_{v_i} + (\sum_{i=1}^{m} e_{i,1}) z + (\sum_{i=1}^{m} e_{i,2}) z^2 + \cdots + (\sum_{i=1}^{m} e_{i,l-1}) z^{l-1} \bmod q$$
$$\Rightarrow s_{v_j} = \psi(j) = \sum_{i=1}^{m} \psi_i(j) \bmod q$$

Let $Y_{VG} = g^{b'_0} \bmod p = g^{\sum_{i=1}^{m} b_{v_i}} \bmod p = \prod_{i=1}^{m} g^{b_{v_i}} \bmod p = \prod_{i=1}^{m} y''_{v_i} \bmod p$; we define y_{VG} as the public key of the verifier group, and b'_0 is the private key of the proxy group.

[Phase 3 - Proxy key generation]

1. Original signer O randomly chooses $k \in_R Z_q^*$, and computes $K = g^k \bmod p$, then concatenates a warrant M_w, which describes the information about the proxy group and verifier group, with T, K, y_{VG}, where T is the time stamp; Y_{VG} is the public key of the verifier group, to construct the hash value $e_h = h(M_w \| T \| K \| y_{VG})$. The original signer generates the proxy key σ by the following equation:

$$\sigma = x_o \cdot e_h + k \bmod q$$

2. O splits the proxy key into t parts by applying the threshold scheme (t, n) to generate shares to the proxy group. It generates the following polynomial:

$f'(z) = \sigma + d_1 z + d_2 z^2 + \cdots + d_{t-1} z^{t-1} \bmod q$; And it publishes the values

$D_j = g^{d_j} \bmod p; \forall j = 1,...,t-1$. And it computes the proxy key shares to the proxy

member p_i by the above polynomial: $\sigma_{p_i} = f'(i); \forall i = 1,...,n$

3. After that it sends σ_{p_i} to each proxy signer in a secure manner, and broadcasts $h(M_w \| T \| K \| y_{VG})$.

4. Each of the proxy signers p_i verifies the validity of those messages by checking $\begin{cases} e_h = h(M_w \| T \| K \| y_{VG}) \\ g^{\sigma_{p_i}} \stackrel{?}{=} y_o^{e_h} \cdot K \cdot \prod_{j=1}^{t-1} D_j^{i^j} \bmod p \end{cases}$. If it holds, then the proxy group does the proxy share generation to satisfy partial delegation with protected scheme.

5. The proxy signer p_i computes $\sigma'_{p_i} = \sigma_{p_i} + s_{p_i} \cdot e_h \bmod q$; then it uses σ'_{p_i} as its proxy share for signing message.

[Phase 4 - Proxy signature generation and issuing]

1. Without lost of generality, we assume that $p_1, p_2, ..., p_t$ are proxy signers who want to cooperate sign the message M on behalf of an original signer. These t proxy signers randomly choose k_{1,p_i} and k_{2,p_i}, then they execute the same operations as described above to share the two random numbers k_1 and k_2.

2. Without lost of generality, we define the following polynomial

$f''(z) = k_2 + e_1 z + e_2 z^2 + \cdots + e_{l-1} z^{l-1} \bmod q = \sum_{j=1}^{t} f_k''(z) \bmod q; where k_2 = \sum_{j=1}^{t} k_{2,p_j}$

$\Rightarrow f''(z) = \sum_{j=1}^{t} k_{2,p_j} + (\sum_{j=1}^{t} e_{j,1}) z + (\sum_{j=1}^{t} e_{j,2}) z^2 + \cdots + (\sum_{j=1}^{t} e_{j,t-1}) z^{t-1} \bmod q$. Then,

$\Rightarrow k_2 = f''(z) = \sum_{j=1}^{t} f''(z) \bmod q, Q_j = g^{e_j} \bmod p; \forall j = 1,2,3,...,t-1$

the proxy signer p_i computes:

$\xi_{p_i} = k_{2,p_i} - \sigma'_{p_i} \cdot e'_h \bmod q; where\ e'_h = h(M \| M_w \| T \| K \| y_{VG})$; And it sends ξ_{p_i} to the other proxy signers in a secure manner. All the other proxy signers do the same process as p_i.

3. After all, the proxy signer p_i can confirm the validity of each ξ_{p_j} by checking the following equation:

$$g^{\xi_{p_j}} \bmod p = g^{k_{2,p_j} - \sigma'_{p_j} \cdot e'_h} \bmod p$$

$$\Rightarrow g^{\xi_{p_j}} \bmod p = (y_{k_{2,p_j}} \cdot \prod_{i=1}^{t-1} Q_i^{j^i}) \cdot [y_o^{e_h} \cdot K \cdot \prod_{j=1}^{t-1} D_j^{i^j} \cdot (y_{PG} \cdot \prod_{i=1}^{t-1} A_i^{j^i})^{e_h}]^{-e'_h} \bmod p$$

4. If all of them are hold, then the proxy signer can compute k_1 and k_2, $R = g^{k_1 - k_2} \bmod p$ and $Z = y_{VG}^{k_1} \bmod p$; finally, it can generate the signature, T, on message M:

$$S = k_2 - \sigma' \cdot e''_h \bmod q$$
$$= f''(0) - (f'(0) + f(0) \cdot e_h) \cdot e''_h \bmod q$$
$$where\ e''_h = h(M \| M_w \| K \| T \| y_{VG} \| R \| Z)$$

. Then it sends $S, M, M_w, K, T, y_{VG}, R, Z$ to the verifier group

[Phase 5 - Verification]

1. The verifier group can cooperate to generate the share group's private key b'_o; and then check the following equation:

$$(g^S \cdot y_{ps}^{e''_h} \cdot R)^{b'_0} \bmod p \stackrel{?}{=} Z \bmod p$$
$$where\ y_{ps} = g^{f'(0) + f(0) \cdot e_h} \bmod p$$

第四章 分析與比較

此章主要是在對本論文所提的分散式提名簽章演算法進行分析，並且會與 Zuo-Wen Tan 與 Zhuo-Jun Liu 的演算法做比較。此章節的內容，包含安全性分析，Kaliski 方法評估，以及小結。

4.1 安全性分析

任何的數位簽章，不論其方法為何，都要符合幾項安全性的分析。根據(#8)，任一 proxy signature scheme 基本的安全性需求應包括：unforgeability、nonrepudiaty、identifiability、verifiability、prevention of misuse。因此，我們提供了對於上述幾項安全性的評估分析：

- Proxy signers' deviation 攻擊：一個成功且安全的簽章技術，其簽名是可以被唯一確認亦不能否認的。因此我們提出的方法必需要符合這個要求，亦既，它必需能抵抗這種攻擊(proxy signer 可以制造出一個有效，又不會被認出是它所做的簽章)，否則它就不是一個成功的簽章法。

  假設現在 proxy signer 手上握有 σ 、 K 以及 original signer 的 public key y_o ，而只要 proxy signer 能夠：

  1. 算出 x_o 或是 k
  2. 產生出一個新的且有效的 $(\tilde{\sigma},\tilde{K})$ 並滿足 $\tilde{\sigma} \equiv x_o \cdot \tilde{e}_h + \tilde{k} \bmod q$ 及 $\tilde{K} = g^{\tilde{k}} \bmod p$

  上述任一項只要 proxy signer 能夠達成，就代表這個方法是不安全的。而要算出上面兩項式子的解，事實上也就代表你必需找出 original

signer 的 private key x_o。因此，只要 original signer 的 private key 能保護好，對 proxy signer 來說要做這種攻擊幾乎不可能。

- 不可偽造性(unforgeability)：安全的簽章法必需保證除了 original signer，或是被授權的 proxy signer 能替它產生有同等效力的簽章外，其他人無法偽造出和 original signer 有同等效力的簽章。事實上這種攻擊要比上一種還更為艱難，因為它不屬於 proxy signer group，它必需先試著解出 $\tilde{\sigma} \equiv x_o \cdot \tilde{e}_h + \tilde{k} \bmod q$ 這個多項式，而它所擁有的只有一些公開的資訊：original signer 的公開金鑰 y_o 以及 K
- 秘密金鑰的依賴性(secret key's dependence)：proxy group 的 secret key σ 是由 original signer 的 private key x_o 計算出來，因此，若沒有 original signer 的 private key，攻擊者是無法產生出一個 σ 的。
- 可驗證性(verifiability)：驗證者必需有能力確認 original signer 確實授權給了 proxy signer。這個可以藉由 K 來達成，因為當 original signer 在授權給不同的 proxy signer group 時，都會產生不同的 K，而在驗證時，original signer 的公開金鑰 y_o 就會是它授權給 proxy signer 的證據
- 可區別性(distingushiability)：驗證者必需有能力能區別出該簽章是由 original signer 或是 proxy signer 所簽署；因為我們採用的是 partial delegation 的簽章技術，兩者可以很容易區別出來。
- 可識別性(identifiability)：original signer 必需要有能力藉由 proxy signer 所簽署的簽章，辨認出這個簽章是哪個 proxy signer group 所簽。這點可以很容易就達成，因為除了 original signer 之外，沒有人可以產生有效的 σ、K 和 e_h 組合，也只有被給與這些參數的 proxy signer 才能替 original signer 產生有效的代理簽章；此外，因為當 original signer 在授權給不同的 proxy signer group 時，都會產生不同的 K，所以不同 proxy signer group 所產生的簽章是可以被辨認出來的

- 不可否認性(nonrepudiaty)：一但一個有效的代理簽章被產生出來後，做出該簽名的 proxy signer 就無法否認是由它所產生。Proxy signer 在簽章時，使用了自己產生的ξ_{p_i}來簽署文件，因此它無法否認所產生的簽章

上述的幾項分析證明了本論文中的新方法依然符合 proxy signature scheme 對於安全的基本需求。除此之外，由於新方法將原本 Zuo-Wen Tan & Zhuo-Jun Liu 方法中幾個重要參數用一些運算給取代掉，讓攻擊者得花更多的精力在破解這些參數上，所以相較之下比起舊方法更安全；而由於新的方法一開始就是針對 ad-hoc network 來設計，因此融入了 threshold proxy signature 裡的技巧，以達到分散計算的效果，也讓風險降低，使得整個簽章機制不會因為 ad-hoc network mobile node 的隨意移動而失去效力，比原方法更具彈性。

在此針對各個方法的安全性及彈性做個小小的總結：

表 4-1 安全及擴充性比較

	Zuo-Wen Tan & Zhuo-Jun Liu	Distributed－1 個 proxy signer 與 verifier	Distributed－t 個 proxy singers 與 l 個 verifiers
所需 proxy signer 數目	1	1	最少 t 個
所需 verifier 數目	1	1	最少 l 個
安全性	差 ⟷ 佳		
彈性	差 ⟷ 佳		

4.2 Computation Time 評估

在 2.6 時，我們曾經介紹 kaliski 學者所提出用來評估數位簽章效率及演算法所需耗費計算量的方法。在此，我們就要用這個方法來對 Zuo-Wen Tan 與 Zhuo-Jun Liu 的方法及本論文的方法來做計算與比較。

以下比較單位為 WM(512)，意即執行 512 bits 模數乘法所需耗費的計算。其餘相關的符號定義可參考 2.6 的說明及表 4-2

表 4- 2 符號表

縮寫符號	意義
H_1	$H(m_w \| T \| r \| y_C)$
H_2	$H(M \| m_w \| y_C \| R \| Z)$
H_3	$H(M \| m_w \| T \| r \| y_C \| R)$
e_h	$H(M_w \| T \| K \| y_{VG})$
$e_h^{'}$	$H(M \| M_w \| T \| K \| y_{VG})$
$e_h^{''}$	$H(M \| M_w \| K \| T \| y_{VG} \| R \| Z)$

1. Zuo-Wen Tan 與 Zhuo-Jun Liu 的方法

[Preliminary Phase]

As mentioned in 2.5 and table 4-2

[Delegation Phase]

A computes:	$k \in_R Z_q^*, r = g^k \bmod p$	640
	$s_A = x_A \times H(m_w \| T \| r \| y_C) + k \bmod q$	$1+WH(H_1)$
A→B:	(m_w, T, r, y_C, s_A)	
B checks:	$g^{s_A} = r \cdot y_A^{H(m_w \| T \| r \| y_C)} \bmod p$	$1+640+WH(H_1)$
B computes:	$s_p = s_A + x_B \cdot H(m_w \| T \| r \| y_C) \bmod p$	1

Delegation Phase Summary: $3+2\cdot WH(H_1)+2\cdot 640$

[Proxy Signature Generation Phase]

B computes:	$R = g^{k_1-k_2} \bmod p$	640
	$Z = y_C^{k_1} \bmod p$	640
	$e = H(M \| m_w \| y_C \| R \| Z)$	$WH(H_2)$
	$s = k_2 - e \cdot s_p \bmod q$	1
B→C:	$(M, m_w, T, y_C, r, R, Z, s)$	

Proxy Signature Generation Phase Summary: $1+WH(H_2)+2\cdot 640$

[Nominative Proxy Signature Verification Phase]

C computes:	$y_p = g^{s_p} = r(y_A y_B)^{H(m_w \| T \| r \| y_C)} \bmod p$	$2+WH(H_1)+640$
C verifies:	$e = H(M \| m_w \| T \| r \| y_C \| R)$	$WH(H_3)$
	$(g^s \cdot y_p^e \cdot R)^{x_C} = Z \bmod p$	2+640

Nominative Proxy Signature Verification Phase: $4+WH(H_1)+WH(H_3)+2\cdot 640$

[Total]

$3848+3\cdot WH(H_1)+WH(H_2)+WH(H_3)$

2. Distributed nominative proxy signature with 1 proxy signer & 1 verifier

[Initial Phase]

B computes:	$r_{p_i} = g^{a_{p_i}} \bmod p$	640
	$b_{p_i} \equiv x_{p_i} \cdot r_{p_i} + a_{p_i} \cdot c_{p_i} \bmod q$	2
	$\Rightarrow y'_{p_i} = g^{b_{p_i}} = y_{p_i}^{r_{p_i}} \cdot r_{p_i}^{c_{p_i}} \bmod p$	640
C computes:	$r'_{v_j} = g^{a'_{v_j}} \bmod p$	640
	$b'_{v_j} \equiv x_{v_j} \cdot r'_{v_j} + a'_{v_j} \cdot c'_{v_j} \bmod q$	2
	$\Rightarrow y''_{v_j} = g^{b'_{v_j}} = y_{v_j}^{r'_{v_j}} \cdot (r'_{v_j})^{c'_{v_j}} \bmod p$	640

Initial Phase Summary: $4 + 4 \cdot 640$

[Delegation Phase]

A computes:	$K = g^k \bmod p$	640
	$e_h = h(M_w \| T \| K \| y_{VG})$	$WH(e_h)$
	$\sigma = x_o \cdot e_h + k \bmod q$	1
A→B:	$(M_w \| T \| K \| y_{VG})$ and σ	
B verifies:	$e_h = h(M_w \| T \| K \| y_{VG})$	$WH(e_h)$
	$g^{\sigma} \stackrel{?}{=} y_o^{e_h} \cdot K \bmod p$	1 + 640
B computes:	$\sigma'_{p_i} = \sigma_{p_i} + s_{p_i} \cdot e_h \bmod q$	1

Delegation Phase Summary: $3 + 2 \cdot WH(e_h) + 2 \cdot 640$

[Proxy Signature Generation Phase]

B randomly chooses k_{1,p_i} and k_{2,p_i}		
B computes:	$e_h^{'} = h(M \| M_w \| T \| K \| y_{VG})$	$WH(e_h^{'})$
	$\xi_{p_i} = k_{2,p_i} - \sigma_{p_i}^{'} \cdot e_h^{'} \bmod q$	1
B computes:	$R = g^{k_1 - k_2} \bmod p$	640
	$Z = y_{VG}^{k_1} \bmod p$	640
B signs:	$S = k_2 - \sigma^{'} \cdot e_h^{"} \bmod q$ $= f^{"}(0) - (f^{'}(0) + f(0) \cdot e_h) \cdot e_h^{"} \bmod q$ *where* $e_h^{"} = h(M \| M_w \| K \| T \| y_{VG} \| R \| Z)$	$1 + WH(e_h^{"})$
B→C:	$S, M, M_w, K, T, y_{VG}, R, Z$	

Proxy Signature Generation Phase Summary: $2 + WH(e_h^{'}) + WH(e_h^{"}) + 2 \cdot 640$

[Nominative Proxy Signature Verification Phase]

C computes:	$y_p = g^{s_p} = r(y_A y_B)^{H(M_w \| T \| K \| y_{VG})} \bmod p$	$2 + WH(e_h) + 640$
C verifies:	$e^{"} = H(M \| m_w \| T \| r \| y_{VG} \| R)$	$WH(e_h^{"})$

Nominative Proxy Signature Verification Phase Summary:

$2 + WH(e_h) + WH(e_h^{"}) + 640$

[Total]

$5771 + 3WH(e_h) + WH(e_h^{'}) + 2WH(e_h^{"})$

3. Distributed nominative proxy signature with t proxy signers & l verifiers

[Proxy Group Share Generation Phase]

B computes:	$r_{p_i} = g^{a_{p_i}} \bmod p$	640×t
	$b_{p_i} \equiv x_{p_i} \cdot r_{p_i} + a_{p_i} \cdot c_{p_i} \bmod q$	2×t
	$\Rightarrow y'_{p_i} = g^{b_{p_i}} = y_{p_i}^{r_{p_i}} \cdot r_{p_i}^{c_{p_i}} \bmod p$	640×t
	$Sign(h(b_{p_i}), r_{p_i}, c_{p_i})$	$WH(b_{p_i})$×t
B computes:	$f_i(j) = b_{p_i} + e_{i,1} j + e_{i,2} j^2 + \cdots + e_{i,t-1} j^{t-1} \bmod q; \forall j = 1,...,n; j \neq i$; send them to other proxy signers	
B broadcasts:	$g^{b_{p_i}}, g^{e_{i,1}}, ..., g^{e_{i,t-1}} (\forall j = 1,2,...,n; and\ j \neq i)$	640×(t-1)×t
B verifies:	$g^{f_j(i)} = g^{b_{p_j}} \cdot (g^{e_{j,1}})^i \cdot (g^{e_{j,2}})^{i^2} \cdots (g^{e_{j,t-1}})^{i^{t-1}} \bmod p$	(640+t-1)×(t-1)×t

Proxy Group Share Generation Summary: $2t + t \cdot WH(b_{p_i}) + t \cdot (t-1)^2 + 2t^2 \cdot 640$

[Verifier Group Share Generation Phase]

C computes:	$r'_{v_j} = g^{a'_{v_j}} \bmod p$	640×l
	$b'_{v_j} \equiv x_{v_j} \cdot r'_{v_j} + a'_{v_j} \cdot c'_{v_j} \bmod q$	2×l
	$\Rightarrow y''_{v_j} = g^{b'_{v_j}} = y_{v_j}^{r'_{v_j}} \cdot (r'_{v_j})^{c'_{v_j}} \bmod p$	640×l
	$Sign(h(b_{p_i}), r_{p_i}, c_{p_i})$	$WH(b'_{v_j})$×l
C computes:	$\psi_j(i) = b'_{v_j} + e_{j,1} i + \cdots + e_{j,l-1} i^{l-1} \bmod q; \forall i = 1,...,m; i \neq j$; send them to other verifiers	
C broadcasts:	$g^{b'_{v_j}}, g^{e_{j,1}}, ..., g^{e_{j,l-1}} (\forall i = 1,2,...,m; and\ i \neq j)$	640×(l-1)×l
C verifies:	$g^{\psi_i(j)} = g^{b'_{v_i}} \cdot (g^{e_{i,1}})^j \cdot (g^{e_{i,2}})^{j^2} \cdots (g^{e_{i,l-1}})^{j^{l-1}} \bmod p$	(640+l-1)×(l-1)×l

Verifier Group Share Generation Summary: $2l + l \cdot WH(b'_{v_j}) + l \cdot (l-1)^2 + 2l^2 \cdot 640$

[Proxy Key Generation Phase]

A computes:	$K = g^k \bmod p$	640
	$e_h = h(M_w \| T \| K \| y_{VG})$	$WH(e_h)$
	$\sigma = x_o \cdot e_h + k \bmod q$	1
	$\sigma_{p_i} = f'(i); \forall i = 1,...,n$	640×t
A broadcasts:	$D_j = g^{d_j} \bmod p; \forall j = 1,...,t-1$	640×(t-1)
A→B:	$(M_w \| T \| K \| y_{VG})$ and σ_{p_i}	
B will check:	$e_h = h(M_w \| T \| K \| y_{VG})$	$WH(e_h)$ ×t
	$g^{\sigma_{p_i}} \stackrel{?}{=} y_o^{e_h} \cdot K \cdot \prod_{j=1}^{t-1} D_j^{i^j} \bmod p$	(t+640)×t
B computes:	$\sigma'_{p_i} = \sigma_{p_i} + s_{p_i} \cdot e_h \bmod q$	t

Proxy Key Generation Summary: $t \cdot (t+1) + 1 + (t+1) \cdot WH(e_h) + 3t \cdot 640$

[Proxy Signature Generation and Issuing Phase]

B computes:	k_{1,p_i} and k_{2,p_i} , and share them as in the same way as y_{PG}	
	2×((640+640+t-1)×(t-1)×t)	
B computes:	$\xi_{p_i} = k_{2,p_i} - \sigma'_{p_i} \cdot e'_h \bmod q; where\ e'_h = h(M \| M_w \| T \| K \| y_{VG})$, send	
	ξ_{p_i} to the other proxy signers	$(1+WH(e'_h))$×t
B verifies:	$g^{\xi_{p_j}} \bmod p = g^{k_{2,p_j} - \sigma'_{p_j} \cdot e'_h} \bmod p$	
	$g^{\xi_{p_j}} \bmod p = (y_{k_{2,p_j}} \cdot \prod_{i=1}^{t-1} Q_i^{j^i}) \cdot [y_o^{e_h} \cdot K \cdot \prod_{j=1}^{t-1} D_j^{i^j} \cdot (y_{PG} \cdot \prod_{i=1}^{t-1} A_i^{j^i})^{e_h}]^{-e'_h} \bmod p$	
	(3×t+1+640)×t	
B computes:	$R = g^{k_1 - k_2} \bmod p$	640
	$Z = y_{VG}^{k_1} \bmod p$	640
	$S = k_2 - \sigma' \cdot e''_h \bmod q$	
B signs:	$= f''(0) - (f'(0) + f(0) \cdot e_h) \cdot e''_h \bmod q$	$1+WH(e''_h)$
	$where\ e''_h = h(M \| M_w \| K \| T \| y_{VG} \| R \| Z)$	
B→C:	$S, M, M_w, K, T, y_{VG}, R, Z$	

Proxy Signature Generation and Issuing Summary:

$$2t^3 - t^2 + 4t + t \cdot WH(e'_h) + WH(e''_h) + [4t^2 - 3t + 2] \cdot 640$$

[Verification Phase]

C verifies:	$(g^S \cdot y_{ps}^{e''_h} \cdot R)^{b'_0} \bmod p \stackrel{?}{=} Z \bmod p$	
	$where\ y_{ps} = g^{f'(0) + f(0) \cdot e_h} \bmod p$	
	$WH(e_h)$ +640+2+ $WH(e''_h)$ +640	

Verification Phase Summary: $2 + WH(e_h) + WH(e''_h) + 2 \cdot 640$

[Total]

640×(6t^2 + 2l^2 + 4t) + (t+2) $WH(e_h)$ + t×

$WH(e_h^{'}) + 2WH(e_h^{''}) + t \cdot WH(b_{p_i}) + l \cdot WH(b_{v_j}^{'}) + 3t^3 - 2t^2 + 8t + l^3 - 2l^2 + 3l + 3$

在此針對各個方法的 computing time 做個總結

表 4- 3 Computing Time 比較表

	Zuo-Wen Tan & Zhuo-Jun Liu	Distributed－1 個 proxy signer 與 verifier	Distributed－t 個 proxy singers 與 l 個 verifiers
所需耗費計算量	O(1)	O(1)	O(t^2+l^2)

4.3 比較結果

透過上述分析，我們大致可以了解這兩個演算法的差異。在表 4-1 中放的是 Zuo-Wen Tan & Zhuo-Jun Liu 與 Distributed nominative proxy signature 採用只有一個 proxy singer 與 verifier 的情形，及擁有 t 個 proxy signers 與 verifiers 的情況。由於在新的方法中，我們將舊演算法裡的一些參數重新拼湊，以增加攻擊者試圖破解所需耗費的時間，增加安全性，而因為採用了分散式的方法，使得新的演算法更具有彈性。

而在表 4-3 裡，是比較兩個方法在 computing time 的差距。很明顯的可以看到因為採用分散式的處理，所以需要的總運算大幅上昇，呈現指數性的成長，這也是本方法的一大缺點。不過也提昇了安全性以及彈性，更符合 ad-hoc network 本身的特性。

第五章 結論與未來研究方向

5.1 結論

隨著科技的進步，越來越多資訊都被數位化，而數位簽章更是其中一項重大的發明。藉由數位簽章，我們可以更加快速的處理文件及資訊。而許多學者在數位簽章這方面亦有許多貢獻，各式各樣的方法都被陸續提出來，來符合各種不同的需求。

然而雖然已有許多演算法被提出，但針對 ad-hoc network，卻沒有一個較為適用的方法來配合其多變的群體與網路拓樸特性。因此本論文在研究了許多現有的簽章技術後，採用 Zuo-Wen Tan & Zhuo-Jun Liu 兩位學者提出的方法做為根基，再搭配 threshold proxy signature 裡所用到的方法，試圖以分散式的方式來配合 ad-hoc network 多變的情況。

在 4.3 的比較表中，可以明顯看到採用新演算法時，彈性性及安全性都有了提昇，但在所需耗費的計算中亦多出了不少，尤其是在數個 proxy signers 及 verifiers 時，更是呈現指數成長，需要龐大的運算，這是為了符合 ad-hoc network 網路特性所需付出的代價。

5.2 未來發展

在本篇論文中，我們大量採用了 threshold proxy signature 分散 share 的方法來達到分散運算及配合 ad-hoc network 的目的。但反過來說，這種方法將會比原本的方法需要耗費更多的計算，因此，如果能夠找到替代的演算方法來分散計算或採用別的計算方式來減少每個 node 所需呈擔的計算量，將會是最主要的發展方向。

參考文獻

[1] Ting-Yi Chang, Chou-Chen Yang and Min-Shiang Hwang, "A threshold signature scheme for group communications without a shared distribution center", 2003

[2] Rosario Gennaro, Stanistaw Jarecki,, Hugo Krawczyk and Tal Rabin, "Robust Threshold DSS Signatures", March 3, 1997

[3] Manik Lal Das, Ashutosh Saxena and V P Gulati, "Security Analysis of Lal and Awasthi's Proxy Signature Schemes", Institute for Development and Research in Banking Technology

[4] Masahiro MAMBO, Keisuke USUDA and Eiji OKAMOTO, "Proxy signatures for Delegating Signing Operation", School of Information Science, Japan Advanced Institute of Science and Technology, 1996 ACM

[5] H.-M.Sun, N.-Y.Lee and T.Hwang, "Threshold proxy signatures", IEE Proc.-Computers & Digital Techniques, Vol. 146, No. 5, September 1999

[6] Zuo-Wen Tan and Zhuo-Jun Liu, "Nominative Proxy Signature Schemes", Institute of Systems Science , AMSS, Chinese Academy of Sciences, State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, 2004

[7] Jianhong Zhang, Qianhong Wu, Jilin Wang and Yumin Wang, "An Improved Nominative Proxy Signature Scheme for Mobile Communication", Key Laboratory of the Ministry Education, Xidian University, Proceedings of the 18th International Conference on Advanced information Networking and Application, 2004 IEEE

[8] Jianhong Zhang, Jiancheng Zou, Member, IEEE and Yumin Wang, Member, IEEE, "Two Modified Nominative Proxy Signature Schemes for Mobile

Communication", 2005 IEEE