國立政治大學資訊科學系

Department of Computer Science

National Chengchi University

碩士論文

Master Thesis

雙方相等性驗證機制的設計及其應用

A Study on the Design of Two-Party Equality Testing Protocol and Its Applications

研 究 生：吳承峰

指導教授：左瑞麟

中華民國 一〇一年七月

July 2012

雙方相等性驗證機制的設計及其應用

A Study on the Design of Two-Party Equality Testing Protocol and Its Applications

研 究 生：吳承峰　　Student：Cheng-Feng Wu

指導教授：左瑞麟　　Advisor：Raylin Tso

國立政治大學

資訊科學系

碩士論文

A Thesis

submitted to Department of Computer Science

in partial fulfillment of the Requirements

for the degree of

Master

in

Computer Science

中華民國一〇一年七月

July 2011

雙方相等性驗證機制的設計及其應用

摘要

雙方相等性驗證即是在不洩漏任何自身私密資訊的情況下，進行秘密計算來了解彼此的資訊是否相等。然而在大多數的現有協議之中，多數為不公平的協定，也就是說其中的一方(被告知方)只能相信另一方(告知方)所告知的比較結果，而無從驗證。雖然邱等學者在 2011 年提出的〝具隱私保護功能之兩方相等性驗證機制之提案〞已經提供了具雙方驗證的協定，但此方案因為在加密演算法上的限制導致實作較為困難。因此，在本論文中，將利用 ElGamal 的加密機制，提出了一套新的雙方相等性驗證的協議，具備相同的雙方相等性驗證的功能，但對加密演算法的限制較少，實作及運算也較為有效率。另外，搭配模糊傳輸的協定，讓使用者藉由本研究所提出的協定跟伺服器端溝通，來獲得所欲取得的資料，並同時保障使用者以及伺服器端的隱私。同時除了理論的證明安全性及正確性之外，也撰寫程式模擬並證實協定的正確性及討論其效能。

A Study on the Design of Two-Party Equality Testing Protocol and Its Applications

Abstract

Two-party equality testing protocol allows two entities to compare their secrete information without leaking any information except the comparison result. In previous works, the comparison result can only be obtained by one entity (ie. informer) and then the entity informs the result to the other entity (ie. receiver). The receiver has to accept the received result since he has no way to verify its correctness. Ciou et al. in 2011 first mentioned this problem and proposed a new protocol to solve the aforementioned problem. However, their protocol has some specific restrictions which making it unpractical. In this paper, based on the ElGamal encryption, we propose a new two-party equality testing protocol. Our protocol has the same feature (ie. allows the two entries to test the correctness of the comparison result) as Ciou et al.'s protocol but is more efficient and practical than theirs. On the other hand, combining our protocol with an oblivious transfer protocol can let users communicate with servers and to get the data in a private way. It is useful on the issue of privacy protection. Finally, the security and correctness are discussed and proved. The efficiency of the protocol is also provided.

致謝

時光飛逝，歲月如梭。兩年的研究生生活隨著此篇論文的完稿，也到了尾聲。在人生的下個階段開始之前，回顧這兩年來的生活，要感謝的人實在是太多太多了。首先要感謝的人，當然是我的指導教授，左瑞麟老師。左老師在這兩年內的細心指導，讓整個協定以及論文成形、完整。並且在每一次碰到問題與瓶頸的同時，都盡心盡力地協助我通過一道道的難關。再多的言語，也無法表達我對老師的崇敬與感謝。

再來，要感謝實驗室的學長們以及同學們。首先是學長士峰以及致諺，他們在我碩一時與我討論跟給予指教，對我的影響極大。再來是同學凱彬，程式在他的幫忙下，撰寫得更加的順暢，並且感謝他與我的每一次討論，都讓我獲益良多。最後是同學漢光跟欣瑤，有他們的幫忙與競爭，讓論文的完成進度更佳的有效率。

最後，當然要感謝我的父母在這兩年來的支持，感謝有他們做為我強力的後盾。也要感謝女朋友在碩二這一年來的督促以及支持，讓我能夠專心並且具備動力的完成我的論文。

僅以這篇論文，獻給所有在這兩年內支持鼓勵我的所有師長及好友。

目錄

國立政治大學 National Chengchi University

圖目錄

表格目錄

國立政治大學 National Chengchi University

1. 緒論

1.1 研究背景

古代的人怎麼保護自己所珍藏重視或是價值不斐的物品？沒錯，就是放在一個堅固的箱子裡，上鎖後放在一個隱密的地方，除了自己以外沒有任何人有鑰匙，如此一來就完成了保護。隨著時代的進步，這個箱子就進步成了眾人皆知的保險箱，傳統的鎖也進步成了密碼鎖，開箱的鑰匙當然也就從有形的鑰匙轉變成了無形的密碼。但不管箱子再怎麼的堅固，鎖設計的再怎麼精良，能夠保護的始終都被限定在有形的物體，那麼無形的一切呢？

在這個數位化的時代，人們的生活漸漸的離不開電腦，也離不開網路。每當人們在享受電腦與網路所帶來的便利時，同時也不禁眉頭深鎖的擔心自己在電腦上所建立的檔案是否會遭到剽竊？利用網路做溝通是否會被他人竊聽？網路交易的紀錄是否會被他人從中獲得個資？從實體的刷卡動作轉變到網路的信用卡交易，是否安全無虞？這些無形的一切又該用什麼去加以保護呢？有了密碼學的幫助，讓這些重重的顧慮一掃而空。

密碼學一直以來都是資訊安全上相當重要的一個環節，有了密碼學的幫助，電腦及網路上一切活動都有了相當程度的保障。一個良好的密碼設計，基本的保護了資料的安全性；一個完善的加密系統，進一步的保護了資料的隱私性。根據不同的安全性要求，密碼學可以做出千百種的變化跟調整，以達到最嚴密的保護。

網路普及已經多年，隨著網路速度的加快與網路科技的進步，各種網路服務也如雨後春筍般地冒出，並且以驚人的速度蓬勃發展。人們能夠在網路上面完成的事情也愈來愈多，甚至以網路行為取代既有的行為模式。因此，網路安全被重視的程度也與日俱增。人們已經不再滿足於基本的資料加密保護，而更進一步的想保護個人的隱私性，不想讓任何人在未經自身的許可下，得知自己在網路上所做的一切行為。換句話說，人們希望自己在網路上就如同一個被完善保護的祕密。

在現實的世界裡，要保護一個寶物最直觀的方式就是將它鎖在一個設計精良的保險箱中。除此之外，寶物非到必要時刻它是不會從保險箱中被取出的。這樣的想法同樣可以套用到網路的世界。一個設計完善的加密系統，就如同一個設計精良的保險箱，可以加密使用者在網路上的一切,保障了最基本的安全性。而非必要時刻不取出寶物的觀念，便是本文的核心所在。一個寶物如果常常被從保險箱取出，那就大大的增加了它被盜取的風險。同樣的，就算一切都經過加密的處理，但太頻繁的解密動作就會讓祕密處於密文型態的時間大減，被保護的效果大大降低，使得祕密曝光的機率倍增，這絕對不是網路使用者所樂於見到的。但是，之所以使用電腦跟網路就是因為仰賴電腦的運算能力能夠幫助人們在有限的時間內做快速的資料運算，再搭配網路的便利特性，達到事半功倍的目的。如果為了保護安全性及隱私而不解密，電腦要如何在不解密的情況下做運算呢？因此，如何在加密且不洩漏資訊的前提下，仍可藉由特定協議過程達到運算目的，便成為一個在密碼學中被廣為研究的課題。安全的多方計算(secure multi-party computation)即是一個這樣類型的問題，而為了滿足安全多方計算的目的，利用滿足同態加密(Homomorphic Encryption)的演算法是一個相當不錯的選擇，也是本文研究的重點。

安全的多方計算[24]是一個允許兩方或是更多的參與者在不揭露自身祕密資訊的條件底下，獲得最終的運算結果的技術。換句話說，每個參與者所知曉的只有自身資訊以及最後的運算結果，對於其他參與者的資訊以及任何額外資訊一無所知也無從得知。因為如此，在一個有多個使用者且彼此不信任的網路中，藉由安全多方計算可以讓多個使用者完成協同運算，同時也具備安全性以及隱私的保護。

同態加密演算法根據其加密演算法能否同時符合加法與乘法兩種同態性質做為區分。若一個同態加密演算法只能單一滿足加法或乘法的同態運算性質，則稱作部分同態運算加密演算法。反之，若可以同時滿足加法及乘法的同態運算性質，則稱作全同態加密演算法。符合全同態加密的演算法則是在 2009 年首度由 Gentry 在[12]所發表，文中定義了何謂同態加密以及全同態加密，同時說明了使用同態加密系統的意義在於就算密

文不經過解密的步驟，仍可對密文型態的數據做運算、檢索、比較…等等的操作，並且確保其正確性。如此一來，便可以解決將數據委託給第三方做運算時的保密問題。

1.2 研究動機與目的

自 1982 年 Yao 在[24]提出百萬富翁問題以及安全計算協定後，許多專家學者也紛紛地投入這個研究議題，也有許多不錯的研究成果。近年來，因為雲端科技這個新議題的興起，讓祕密計算的研究又再一次的被專家學者們重視。

短短的幾年，雲端科技發展日益成熟，且至今仍在蓬勃的發展當中。無論是國家還是企業，都紛紛投入雲端的研究跟運用，希望能夠搶得先機，開拓市場或是利用雲的幫助提升效率或產能。而在一般個人使用者的層面，也有相當多的服務陸續推出，讓一般民眾在雲端科技的協助下，你我都能利用雲的協助，來達成資料的儲存、運算、比對、著作、編輯、商業行為…等等的目的。像是雲端龍頭 Google 的 Google Cloud，由 Google 的雲端運算平台提供一個由 Gmail、Google Calendar、Google Docs、Google Talk 及 Google Pages 所組成的 Google Apps。此外，Google Chrome 的推出甚至改變了一般大眾對於網頁瀏覽器的認知，讓網頁瀏覽器不再只是瀏覽器，而是一個執行 Apps 的平台，一種線上的作業系統。再者，近幾個月，Google 雲端硬碟(Google Drive)也正式登場，提供了免費的 5GB 的空間讓使用者使用。當然除了 Goolge 雲端硬碟之外，DropBox 早在 2008 即推出了個人的網路雲端硬碟，更在 2011 年提供了 Dropbox for Teams。蘋果也有 iCloud 供蘋果使用者作同步以及備份資料的使用。除了上述的服務以外，大大小小的應用也都正在被開發中，雲端運算實為一個相當熱門的議題以及便利的技術。而在這個熱潮下，相對於雲端科技的便利性，雲的安全性跟隱私性一直以來也是備受人們所關注的，該使用什麼樣的方法才能夠在不洩漏使用者的個人資訊下，達到兼具便利性與隱私性的目標，是許多專家學者致力研究的方向。

本研究就是針對使用者隱私保護的動機，想要設計了一個協定並加以實作。滿足在

不洩漏任何個人私密資訊的條件下,達到雙方相等性驗證的目的。所謂的雙方相等驗證，就是兩方在不洩漏自身私密資訊的情況底下，比較雙方的私密資訊是否相等。藉由雙方相等性驗證，雙方都可以得知比較的結果並且做出驗證。在雙方誠實的情況下，若比較結果不相等，彼此也完全不會知道對方的私密資訊為何。若有其中一方說謊，另一方都可驗證發現。雙方相等性驗證應用的範圍相當的廣泛，本研究就設計了三個應用，其中一個是登入系統，另外兩個則分別是建構出線上購買數位化商品的機制以及結合數位與現實的投票系統。而後述的這兩個都是以雙方相等性驗證搭配模糊傳輸的概念與技術將協定套用在使用者與伺服器(Client & Server)的架構下，讓使用者藉由本研究所提出的協定跟伺服器端溝通，來獲得所欲取得的資料，並同時保障使用者以及伺服器端的隱私。

1.3 研究貢獻

目前的雙方相等性驗證協定，都需要較龐大的計算量或是條件限制較多的加密演算法。因此，本研究利用 ElGamal 加密演算法設計了一個雙方相等性驗證的協定，相對於目前的其他雙方相等性驗證協定，本協定是以較低的運算成本以及較少的條件限制達到目的，而且相較於既有之方案並沒有做任何的安全性的犧牲。除此之外，本研究更以此協定搭配模糊傳輸(Oblivious Transfer)，在使用者與伺服器的架構下，保證不洩漏雙方的任何資訊來達到完成私密資訊查詢(Private Information Query)的目的。最後以撰寫 JAVA 程式加以實作並討論其表現。

1.4 論文架構

本論文共 7 個章節，第 1 章為緒論，接著第 2 章將介紹一些與本研究相關的知識背景以及研究，包含了同態公開金鑰加密、秘密計算、ElGamal 加密系統、模糊傳輸…等。第 3 章將介紹本論文主要的研究方法與協定;第 4 章對本論文所提出的協定做安全性分析，並在第 5 章提出對於協定的相關應用。第 6 章是如何將本論文提出的協定加以實作，並在第 7 章做出結論以及未來展望。

2. 背景知識與相關研究

2.1 同態加密 (Homomorphic Encryption)

同態加密是一種特殊的加密形式，如果一個加密系統具備同態加密的性質，表示這個加密系統滿足對密文進行特定的代數運算得到的結果，會同等於對明文運算後再加密的結果。換句話說，就是要能夠滿足$E(m_1)\odot E(m_2) = E(m_1 \circledast m_2)$，$\odot$與$\circledast$是兩個運算子，而根據運算子$\circledast$的不同，又可以分作加法同態加密(Additive Homomorphic Encryption)與乘法同態加密(Multiplicatively Homomorphic Encryption)，這兩個不同的同態性質會在 2.2.1 與 2.2.2 做較為詳細的介紹。

使用同態加密系統的意義在於就算密文不經過解密的步驟，仍可對密文型態的數據做運算、檢索、比較…等等的操作，並且確保其正確性。如此一來，便可以解決將數據委託給第三方做運算時的保密問題,例如雲端計算的隱私保護問題。不僅僅是雲端計算，同態加密也被應用在許多不同的地方，像是創建一個安全的投票系統[16]、抗碰撞(Collision-resistant)的雜湊函數(Hash function)[15]或是私密資訊擷取(Private Information Retrieval, PIR)[13]。

更進一步來說，同態是一個代數系統(Algebraic system)的對應關係，從一個代數系統對應到同類的代數系統，例如群(Group)、環(Ring)、向量空間(Vector space)的對應關係。在同態的運算過程當中，會保持所有相關的結構不變。換句話說，也就是保持單位元素(Unit element)、反元素(Inverse element)以及二元運算(Binary compute)的屬性不變。以達到我們接下來要說明的加法和乘法的同態性質。

2.1.1 加法同態

若一個加密系統滿足$E(m_1)\odot E(m_2) = E(m_1 + m_2)$，則稱這個加密系統為加法同態加密系統，舉兩個簡單的例子來說：

(1) 若$E(m_1) = km_1, E(m_2) = km_2$則$E(m_1) + E(m_2) = k(m_1 + m_2) = E(m_1 + m_2)$

(2) 若$E(m_1) = e^{m_1}, E(m_2) = e^{m_2}$則$E(m_1) \times E(m_2) = e^{m_1+m_2} = E(m_1 + m_2)$

目前已知滿足加法同態的著名加密系統有： Paillier[21]以及 Benaloh[1]。以下利用 Paillier 的演算法做更詳盡的說明。

Paillier 加密系統分做三個部分，分別是金鑰生成(Key generation)、加密演算法(Encryption)以及解密演算法(Decryption)，以下將逐步做介紹。

金鑰生成：

1. Alice 選取兩個大質數p, q且$gcd\big(pq, (p-1)(q-1)\big) = 1$。
2. Alice 計算$N = p \times q, \lambda = lcm(p-1, q-1)$。
3. Alice 選擇亂數$g \in \mathrm{Z}^*_{N^2}$。
4. 函數$\mathrm{L(u)} = \frac{\mathrm{u-1}}{\mathrm{N}}$，且 Alice 利用是否存在$\mu = \big(L\big(g^{\lambda}\big) mod\, N^2\big)^{-1}\, mod\, N$來確保若$g$的序為 k，則$N|k$。其中，$N|k$表示N為k的因數。
5. Alice 的公鑰為$pk = (N, g)$ ，私鑰為$sk = (\lambda, \mu)$。

加密演算法：

1. Bob 欲加密給 Alice 的訊息為$m \in Z_N$。
2. Bob 選擇亂數$r \in Z_N$。
3. 密文$C = E(m) = g^m \times r^N\, mod\, N^2$並傳送給 Alice。

解密演算法：

1. 密文$C \in Z^*_{N^2}$。
2. Alice 計算明文$m = L\big(c^{\lambda}\, mod\, N^2\big) \times \mu\, mod\, N$。

加法同態：

在介紹完 Paillier 的加解密演算法後，接著說明 Paillier 這個加解密演算法所符合的加法同態性質。首先假設欲加密的兩個變數為m_1, m_2，並且分別在經過加密後得到：

$$C_1 = E(m_1) = g^{m_1} \times r_1^N\, mod\, N^2$$

$$C_2 = E(m_2) = g^{m_2} \times r_2^N \ mod\ N^2$$

接著計算密文乘積為：

$$C_1 \times C_2 = E(m_1) \times E(m_2) = (g^{m_1} \times r^N) \times (g^{m_2} \times r^N)\ mod\ N^2$$

$$= g^{m_1+m_2} \times (r_1 r_2)^N\ mod\ N^2 = g^{m_1+m_2} \times (r')^N\ mod\ N^2 = E(m_1 + m_2)$$

由上述可清楚地看到$C_1 \times C_2 = E(m_1) \times E(m_2) = E(m_1 + m_2)$，因此 Paillier 加解密演算法是具有加法同態性質的。

2.1.2 乘法同態

若一個加密系統滿足$E(m_1) \odot E(m_2) = E(m_1 \times m_2)$，則稱這個加密系統為乘法同態加密系統。目前已知加法同態的著名加密系統有： ElGamal[11] 以及 RSA[22]。首先以簡單的 RSA 加解密演算法做說明。在下一個節會對本論文所使用的 ElGamal 加密演算法做出更詳盡的介紹，並說明其乘法同態性質。

RSA 加密系統分做三個部分，分別是金鑰生成(Key generation)、加密演算法(Encryption)以及解密演算法(Decryption)，以下將逐步做介紹。

金鑰的生成：

1. Alice 隨機選取兩個相異的質數p, q。為了安全性的考量，兩質數的位元長度必須相近。
2. Alice 計算$N = p \times q$與$\varphi(N) = (p-1) \times (q-1)$，$\varphi$為尤拉函數。
3. Alice 選取一個整數e，$1 < e < \varphi(N), \big(e, \varphi(N)\big) = 1$。
4. Alice 根據所選的e計算d，其中$d \equiv e^{-1}\ mod\ \varphi(N)$ 。
5. Alice 的公鑰為(N, e)，私鑰為(N, d) 。

加密演算法：

1. Bob 欲加密明文m給 Alice。
2. 利用 Alice 的公鑰(N, e)計算密文$C = m^e\ mod\ N$並傳送給 Alice。

解密演算法：

1. Alice 收到密文C。

2. Alice 收到密文後利用私鑰(N, d)做解密，計算$m = c^d \ mod \ N$。

乘法同態：

在介紹完 RSA 的加解密演算法後，接著說明 RSA 這個加解密演算法所符合的乘法同態性質。首先假設欲加密的兩個變數為m_1, m_2，並且分別在經過加密後得到：

$$C_1 = E(m_1) = m_1^e \ mod \ N$$

$$C_2 = E(m_2) = m_2^e \ mod \ N$$

接著計算密文乘積為：

$$C_1 \times C_2 = E(m_1) \times E(m_2) = m_1^e \times m_2^e = E(m_1 \times m_2)$$

由上述可清楚地看到$C_1 \times C_2 = E(m_1) \times E(m_2) = E(m_1 \times m_2)$，因此 RSA 加解密演算法是具有加法同態性質的。

2.2 ElGamal 加密演算法

ElGamal 加密演算法有三個部分，分別是金鑰生成(Key generation)、加密演算法(Encryption)以及解密演算法(Decryption)，在這小節將會逐步做介紹。

金鑰的生成：

1. Alice 取一個生成元g，足以生成一個循環群G，內有q個元素。

2. Alice 任選一個變數 $x \in \{0, \dots, q-1\}$。

3. Alice 計算 $y = g^x \ mod \ p$。

4. Alice 以 (G, q, g, y)作為她的公鑰，x作為她的私鑰。

加密演算法：

1. Bob 任選一個變數 $r \in \{0, \dots, q-1\}$，並且計算$C_1 = g^r \ mod \ p$。

2. Bob 加密明文 m，生成 $C_2 = m \times y^r \ mod \ p$。

3. 傳送密文 (C_1, C_2)給 Alice。

解密演算法：

1. Alice 計算 $C_1^x = g^{rx} = y^r$。
2. 接著計算 $C_2 \times (C_1^x)^{-1} = (m \times y^r) \times (y^r)^{-1} = m$ 即可還原明文。

在了解 ElGamal 加密系統之後，在下一小節 2.2.4 將介紹 ElGamal 加密系統的乘法同態性質。

2.2.1 ElGamal 的乘法同態性質

這小節我們將介紹 ElGamal 加密系統的乘法同態性質。首先利用 ElGamal 加密系統分別以對明文m_1, m_2做加密，產生$(C_1, C_2), (C_1', C_2')$兩組密文，其中：

$$(C_1, C_2) = (g^{r_1}, m_1 \times y^{r_1}) \; mod \; p$$

$$(C_1', C_2') = (g^{r_2}, m_2 \times y^{r_2}) \; mod \; p$$

接著對兩組密文(C_1, C_2)與(C_1', C_2')做乘法運算得到$(C_1'', C_2'') = (C_1, C_2) \times (C_1', C_2')$，其中：

$$C_1'' = C_1 \times C_1' = g^{r_1} \times g^{r_2} \; mod \; p = g^{r_1 + r_2} \; mod \; p$$

$$C_2'' = C_2 \times C_2' = (m_1 \times y^{r_1}) \times (m_2 \times y^{r_2}) \; mod \; p = (m_1 \times m_2) \times y^{r_1 + r_2} \; mod \; p$$

最後做解密：

$$C_2'' \times \left(C_1''^{x}\right)^{-1} = (m_1 \times m_2) \times y^{r_1 + r_2} \times (g^{r_1 + r_2})^{-1} = (m_1 \times m_2) \times y^{r_1 + r_2} \times \left(g^{x(r_1 + r_2)}\right)^{-1}$$

$$= (m_1 \times m_2) \times y^{r_1 + r_2} \times (y^{r_1 + r_2})^{-1} = (m_1 \times m_2)$$

由以上的加解密過程及運算結果，我們可以清楚地看到，對兩組利用 ElGamal 加密系統所加密的密文做乘法運算之後，其解密結果會同等於對明文做乘法運算。因此，ElGamal 加密系統是一個符合乘法同態性質的加密系統，本研究將利用 ElGamal 加密系統符合乘法同態的性質，設計並提出一個可以達到隱私保護的雙方相等性驗證協定。

2.3 祕密計算 (Secret Computation)

在介紹祕密計算之前，先來想像一個情境：期中考過後，甲、乙、丙三人都拿到了考卷也得知了自己的分數，三人想要知道平均是多少，所以想要先算三人分數的加總，卻礙

於面子關係不想讓別人知道自己的成績。有沒有一個辦法可以去解決這個問題呢？其實解決的方法相當的簡單，可以用以下的方法輕鬆達到。

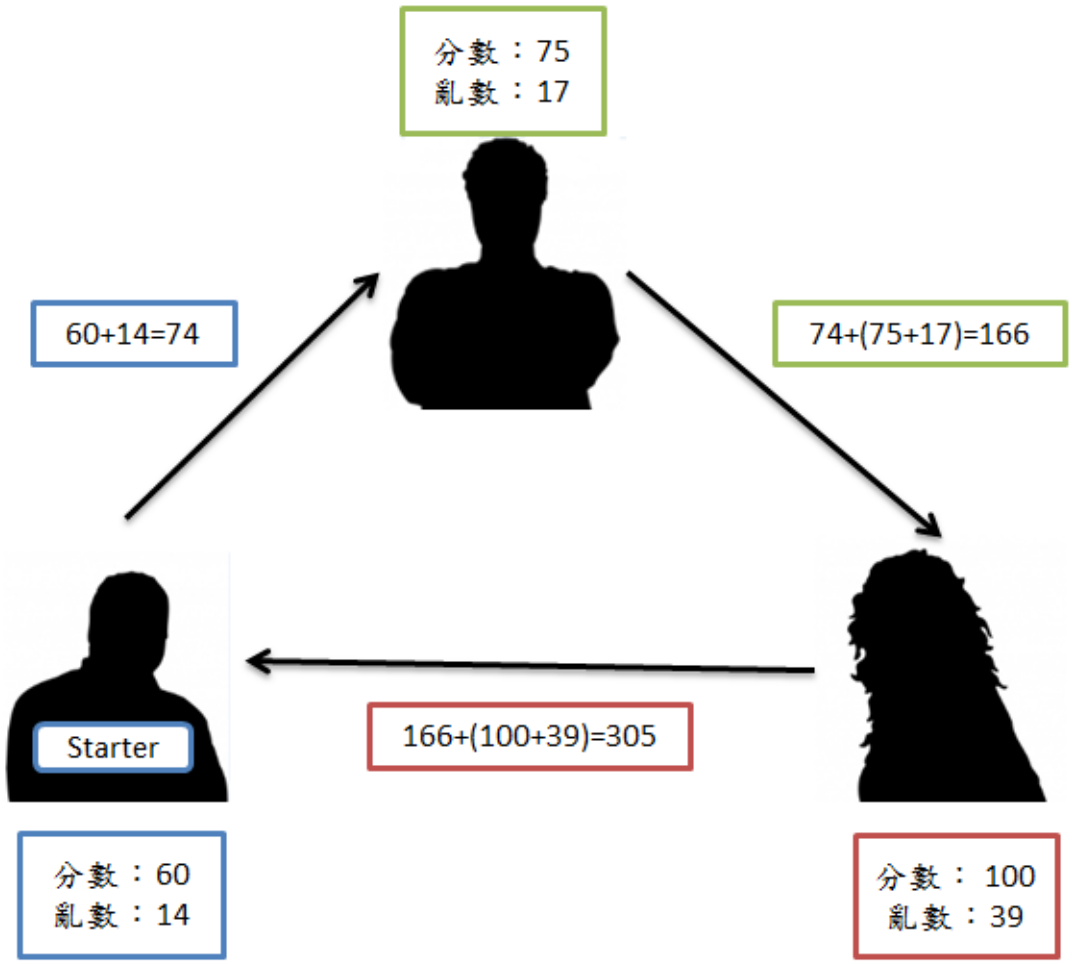

圖 1 簡單祕密計算例圖〈一〉

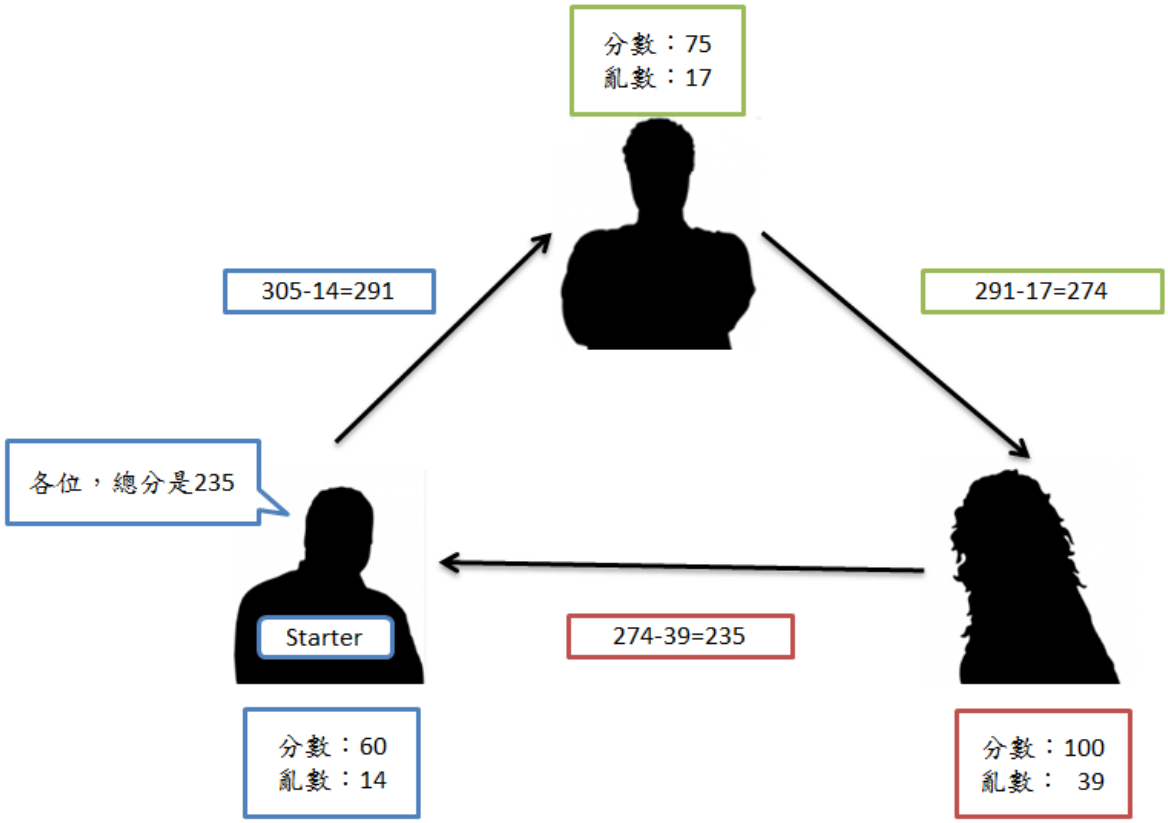

圖 2 簡單祕密計算例圖〈二〉

以上介紹的其實就是個簡單的祕密計算概念，一個安全的祕密計算必須可以保障被運算資料的私密性，在不公開資料的情況下達到正確運算的目的。而 1982 年 Yao 在[24]所提出的百萬富翁問題更是個在祕密計算上相當著名的問題。百萬富翁問題是在探討能否在不透漏兩個富翁的真實財產情況下，透過祕密計算的協定去比較哪一個富翁較為富有。本論文將在接下來的小節，分別介紹雙方架構下的祕密計算協定與藉由半誠實(semi-trust)第三方的祕密計算協定，也將特別對 Yao 以及邱姓學者在雙方架構下所提出的安全祕密計算協定分別作介紹。

2.3.1 Yao 的協定(雙方架構)

假設 a, b 分別為富翁 Alice 與富翁 Bob 的財產(私密資訊)，Alice 跟 Bob 想比較彼此財產的多寡。透過 Yao 提出的協定，可以在不洩漏 a 與 b 給對方的情況下，達到目的。協定

流程如下：

1. Bob 挑一個任意N-bit 的整數x，計算$k = E_{pk_A}(x)$，其中E_{pk_A}是以 Alice 的公鑰pk_A採用 RSA 加密系統的加密演算法。接著 Bob 將$k - b + 1$傳給 Alice。
2. Alice 在收到 Bob 傳來的資訊後，對$k - b + u, u = 1, \dots, m$這$m$個數值作解密，得到$Y_u = D_{sk}(k - b + u), u = 1, .., m$，其中$D_{sk_A}$是以 Alice 的私鑰$sk_A$採用 RSA 加密系統的解密演算法。
3. 接著 Alice 挑選一個任意的質數p，其長度為$\frac{N}{2}$-bit，且必須慎選p使得p與Z_u至少相差 2，計算$Z_u = \begin{cases} Y_u \bmod p, & u = 1, \dots, a \\ Y_u \bmod p + 1, & u = a+1, \dots, m \end{cases}$，並且將$p$與$Z_u$傳送給 Bob。
4. Bob 在收到 Alice 傳來的資訊後，計算$G = x \bmod p$，若$\begin{cases} Z_b = G \Rightarrow a \geq b \\ Z_b \neq G \Rightarrow a < b \end{cases}$，最後 Bob 告知 Alice 判斷結果。

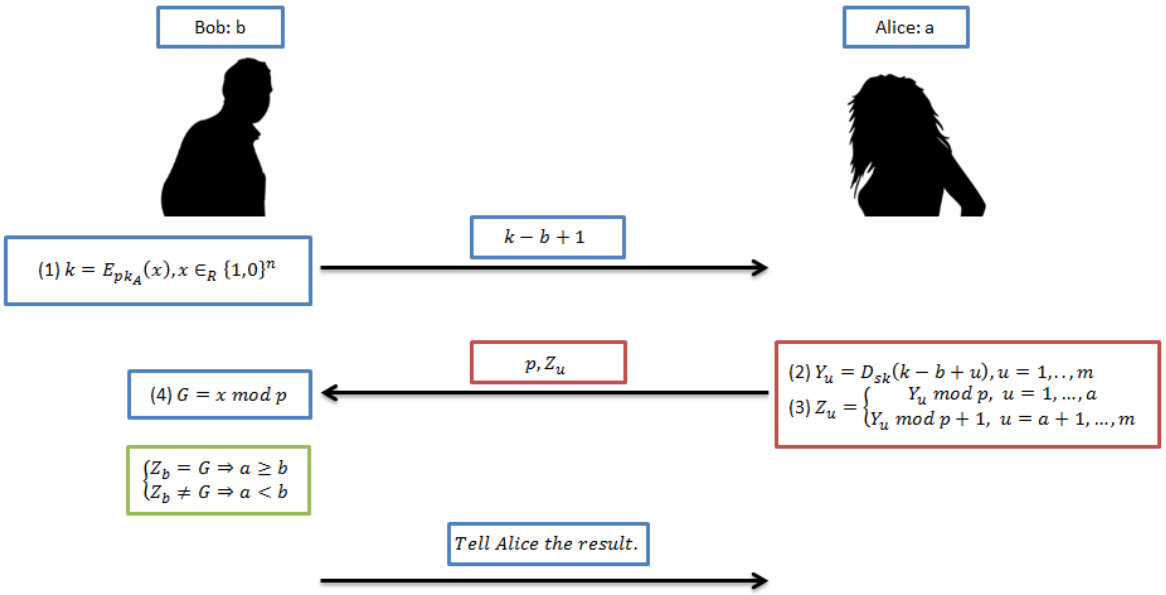

圖 3 Yao 協定流程圖

接著從邱學者在[6] [7]所中舉的簡單例子便能更清楚的了解 Yao 所提出的協定運作以及是否可以利用祕密計算解決百萬富翁的問題。首先，假設 Alice 有 500 萬，Bob 有 600 萬，則$\mathrm{a} = 5, \mathrm{b} = 6$。Alice 的公鑰為$(e, N) = (5,119)$，私鑰為$(d, N) = (77,119)$。

1. Bob 挑一個任意N-bit 的整數$x = 234$，計算$k = E_{pk_A}(x) = 234^5 \ mod\ 119 = 47$，接著 Bob 將$k - b + 1 = 47 - 6 + 1 = 42$傳給 Alice。
2. Alice 在收到 Bob 傳來的資訊後，對$k - b + u, u = 1, \dots, m$這$m$個數值作解密，得到$Y_u = D_{sk}(k - b + u), u = 1, .., m$，下表為解密結果：

表格 1 Yao 協定簡例〈一〉

u	k-j+u	$D_{sk}(k - b + u)$	Y_u
1	42	42^{77} mod 119	77
2	43	43^{77} mod 119	8
3	44	44^{77} mod 119	11
4	45	45^{77} mod 119	75
5	46	46^{77} mod 119	65
6	47	47^{77} mod 119	115
7	48	48^{77} mod 119	90
8	49	49^{77} mod 119	70
9	50	50^{77} mod 119	50
10	51	51^{77} mod 119	102

3. 接著 Alice 挑選一個任意的質數$p = 11$，計算$Z_u = \begin{cases} Y_u \ mod\ p, & u = 1, \dots, a \\ Y_u \ mod\ p + 1, & u = a + 1, \dots, m \end{cases}$，並且將$p$與$Z_u$傳送給 Bob。

表格 2 Yao 協定簡例〈二〉

u	k-j+u	$D_{sk}(k - b + u)$	Y_u	Z_u
1	42	42^{77} mod 119	77	0
2	43	43^{77} mod 119	8	8
3	44	44^{77} mod 119	11	0

4	45	$45^{77} \bmod 119$	75	9
5	46	$46^{77} \bmod 119$	65	10
6	47	$47^{77} \bmod 119$	115	6
7	48	$48^{77} \bmod 119$	90	3
8	49	$49^{77} \bmod 119$	70	5
9	50	$50^{77} \bmod 119$	50	7
10	51	$51^{77} \bmod 119$	102	4

4. Bob 在收到 Alice 傳來的資訊後，計算 $G = x \, mod \, p = 234 \, mod \, p = 5$，若 $\begin{cases} Z_b = G \Rightarrow a \geq b \\ Z_b \neq G \Rightarrow a < b \end{cases} \Rightarrow \begin{cases} Z_6 = 5 \Rightarrow a \geq b \\ Z_6 \neq 5 \Rightarrow a < b \end{cases} \Rightarrow Z_6 = 6 \neq 5 \Rightarrow a < b$，最後 Bob 告知 Alice 判斷結果。

從上述的這個簡單例子，可以知道當 Alice 的財產少於 Bob 的財產時，是可以藉由一次的協定便能得知結果。但假如是 Alice 的財產多於 Bob 的財產或是雙方財產相等時，便需要交換腳色再執行一次協定，才能判斷財產多寡。儘管能夠在不洩漏雙方的資訊下比較雙方數值的大小，卻有效率上的缺點。除此之外，從協定的最後一個步驟可以看出 Alice 作為被告知的一方是沒有辦法去驗證告知方 Bob 是否有欺騙行為，只能相信 Bob 所傳達的結果，對於 Alice 而言是相當不公平的。在介紹完雙方的架構後，接著在下一小節將介紹李姓及武姓學者在[18]中利用半誠實的第三方所提出的協定。

2.3.2 李姓與武姓學者的協定(半誠實第三方架構)

在介紹協定之前，先解釋何謂半誠實的第三方。假如存在一個完全誠實的第三方 C，A 跟 B 便只需要將欲比較的資訊 a,b 利用安全通道傳給 C，C 在比較過雙方大小後分別告知 A 跟 B 結果即可。若是不存在安全通道，則可以利用 C 的公鑰加密 a 跟 b，C 在接收到這兩個密文後，解密比較結果再分別告知 A 跟 B 即可。但上述的兩個方式都是極度理想的狀態，在現實環境中基於安全性的考量我們不能假設有一個完全誠實的第三方。

半誠實的第三方 D 顧名思義並不是個完全誠實的第三方，D 只保證會按照協定的規則進行通訊和計算，但是會試著從協定的缺失獲取額外的資訊。換句話說，假使協定是沒有漏洞的，D 不會從協定中獲得任何額外的資訊。

在李姓及武姓學者所提出的協定中，除了想要比較雙方資訊的 Alice 跟 Bob 外，存在一個半誠實的第三方 Cathy。Cathy 在協定開始前會生成自己的公鑰跟私鑰(pk, sk)，而 Alice 跟 Bob 共同決定一個亂數$r_{ab} \in [0,2^k - 1], k \geq 80$，整個協定中的加解密採用 Paillier 加解密演算法，接著協定開始：

1. Alice 以 Cathy 的公鑰加密自身的私密訊息 a，$x = E_{pk}(r_{ab}a)$，並將x傳送給 Bob。
2. Bob 以 Cathy 的公鑰加密自身的私密訊息 b，$y = E_{pk}(-r_{ab}b)$，並且計算$y' = E_{pk}(r_{ab}a) \times E_{pk}(-r_{ab}b)$，後將$y'$傳送給 Cathy。
3. Cathy 以自己的私鑰對y'做解密，由於 Paillier 的加解密演算法是符合加法同態性質，因此得到 $z = D_{sk}(y') = D_{sk}\left(E_{pk}(r_{ab}a) \times E_{pk}(-r_{ab}b)\right) = D_{sk}\left(E_{pk}(r_{ab}a - r_{ab}b)\right) = D_{sk}\left(E_{pk}(r_{ab}(a-b))\right)$，若$\begin{cases} z = 0,\ a = b \Rightarrow z' = 0 \\ z \neq 0,\ a \neq b \Rightarrow z' = 1 \end{cases}$，接著 Cathy 將$z'$傳送給 Alice 跟 Bob。

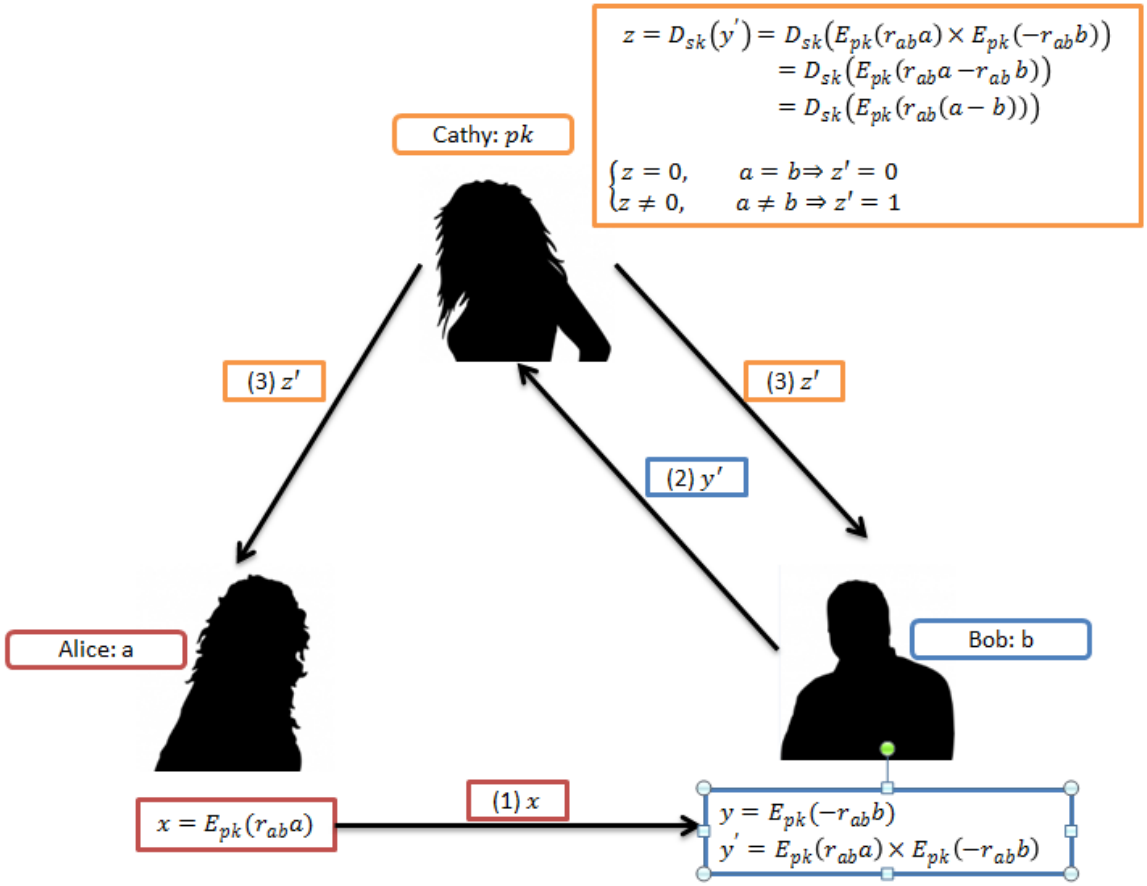

圖 4 李姓及武姓協定流程圖

這個協定雖然可以以一個相當簡單的步驟比較雙方資訊是否相等，但是存在一個只要有第三方的協定就會產生的問題，那就是共謀(collusion)的顧慮。假使 Cathy 跟 Alice 或 Bob 其中一個人產生共謀，那對於另一方來說就顯得相當不公平，除非協定中存在能夠阻止共謀的因子，否則多了第三方便是多了風險，於是本論文還是以雙方架構下的協定為主軸。於是將在下一個小節，對邱姓學者在[6][7]所提出的雙方相等性驗證協定做詳細的介紹。

2.3.3 邱姓學者的協定(雙方架構)

由於 Yao 的協定無法達到雙方驗證，而李姓及武姓學者所提出的協定則需要半誠實的第三方幫助雙方做相等性的判斷。於是邱姓學者在 2012 年提出了具隱私保護功能之兩方相等性驗證機制之提案[6][7]，所提出的協定可以在不需要第三方的條件底下，讓雙方

都能夠驗證彼此的私密資訊是否相同。由於邱姓學者在此提案中所使用的加密演算法須符合兩個較特別的性質，一個是可換式加密(Commutative Encryption)，另一個則是雙重同態加密(Doubly-homomorphic Encryption)。於是在介紹提案之前，先介紹這兩個相關的知識，以便於進一步了解邱姓學者的提案。

■ **可換式加密(Commutative Encryption)**

若一個加密演算法對一個訊息m加密兩次以上，並滿足下列三個條件，則稱作可換式加密：

1. 假設 Alice 的公私鑰為$(\mathrm{pk_A}, \mathrm{sk_A})$，Bob 的公私鑰為$(\mathrm{pk_B}, \mathrm{sk_B})$，則$\mathrm{E_{PK_A}}\left(\mathrm{E_{PK_B}}(\mathrm{m})\right) = \mathrm{E_{PK_B}}\left(\mathrm{E_{PK_A}}(\mathrm{m})\right)$
2. 加密所用公鑰$\mathrm{pk_i}$所對應的解密私鑰$\mathrm{sk_i}$在多項式時間內可計算。
3. 利用$\mathrm{pk_A}$或$\mathrm{pk_B}$所加密產生的密文，其值域相同。

■ **雙重同態加密(Doubly-homomorphic Encryption)**

若一個加密系統同時滿足加法同態性質以及乘法同態性質，即可稱此加密系統為雙重同態加密。換句話說，一個符合雙重同態加密的加密演算法，給定$E(x)$跟$E(y)$，在不知道x, y的情況下，任何人都可以計算出$E(x+y)$和$E(x \times y)$。

在介紹完可換式加密與雙重同態加密之後，接著開始對邱姓學者所提出的雙方相等性驗證機制協定做介紹。首先，先介紹一些協定中所用的符號：a, b分別為 Alice 與 Bob 的私密資訊。E_A, D_A分別為使用 Alice 公鑰PK_A與私鑰SK_A所做的加解密演算法，E_B, D_B分別為使用 Bob 公鑰PK_B與私鑰SK_B所做的加解密演算法。邱姓學者所採用的是 Gentry 的演算法，但由於此演算法只符合雙重同態加密，並沒有滿足可換式加密，所以邱姓學者

在不改變演算法安全性的條件下，對 Gentry 的加解密演算法做了些為調整，使之同時滿足雙重同態加密以及可換式加密。在修改過後的加密演算法為：$c = m + 2^N r' + pq$，並且規定$SK_A \gg SK_B \gg 2^N \gg p'$，$p'$是個由 Alice 與 Bob 共同決定的質數，且$p' \in Z^*, r \in Z_{p'}$。協定流程如下：

1. Alice 生成亂數$r = r_x + r_y, r_x, r_y, r \in Z^*$，接著計算$a \times r_x$並且用$PK_A$加密得到$E_A(a \times r_X)$，再將$r_y$和$E_A(a \times r_X)$傳送給 Bob。
2. Bob 在收到r_y和$E_A(a \times r_X)$之後，計算$b \times r_y$並且用PK_A加密得到$E_A(b \times r_y)$，並挑選一個隨機的亂數$q \in Z^*$加密後得到$E_A(b \times r_y)$。接著利用雙重同態性質運算得到$E_A\left(q\left(ar_X + br_y\right)\right)$，最後 Bob 以自己的公鑰$PK_B$加密得到$E_B E_A\left(q\left(ar_X + br_y\right)\right)$傳送給 Alice。
3. Alice 收到 Bob 傳來的資訊後，利用可交換式加密的特性，先用自己的私鑰SK_A對$E_B E_A\left(q\left(ar_X + br_y\right)\right)$解密，$D_A E_B E_A\left(q\left(ar_X + br_y\right)\right) = E_B\left(q\left(ar_X + br_y\right)\right)$。接著計算$E_B(r^{-1})$，利用乘法同態加密性質運算$E_B(r^{-1}) \times E_B\left(q\left(ar_X + br_y\right)\right) = E_B\left(\frac{q(ar_X + br_y)}{r}\right)$並將運算結果傳給 Bob。
4. Bob 在收到$E_B\left(\frac{q(ar_X + br_y)}{r}\right)$之後用自己的私鑰$SK_B$解密並且$mod\ p'$得到$D_B E_B\left(\frac{q(ar_X + br_y)}{r}\right)\ mod\ p' = \frac{q(ar_X + br_y)}{r}$，接著將$q$除掉。假如$\begin{cases} \frac{(ar_X + br_y)}{r} = \text{b} \Rightarrow \text{a} = \text{b} \\ \frac{(ar_X + br_y)}{r} \neq \text{b} \Rightarrow \text{a} \neq \text{b} \end{cases}$。Bob 將判斷結果連同$\frac{q(ar_X + br_y)}{r}\ mod\ p'$與$E_A\left(q\left(ar_X + br_y\right)\right)$傳送給 Alice。
5. Alice 在收到 Bob 傳遞來的資訊後，以自己的密鑰SK_A將$E_A\left(q\left(ar_X + br_y\right)\right)$解密得到$M = D_A E_A\left(q\left(ar_X + br_y\right)\right) = q\left(ar_X + br_y\right)$，並計算$N = \frac{q(ar_X + br_y)}{r}\ mod\ p' \times r$。此時 Alice 便能夠比較$q\left(ar_X + br_y\right)$是否等於$\frac{q(ar_X + br_y)}{r}\ mod\ p' \times r$，再進一步的判斷若

$$\begin{cases} a \mid \frac{q(ar_X+br_y)}{r} \Rightarrow a = b \\ otherwise \Rightarrow a \neq b \end{cases}$$。

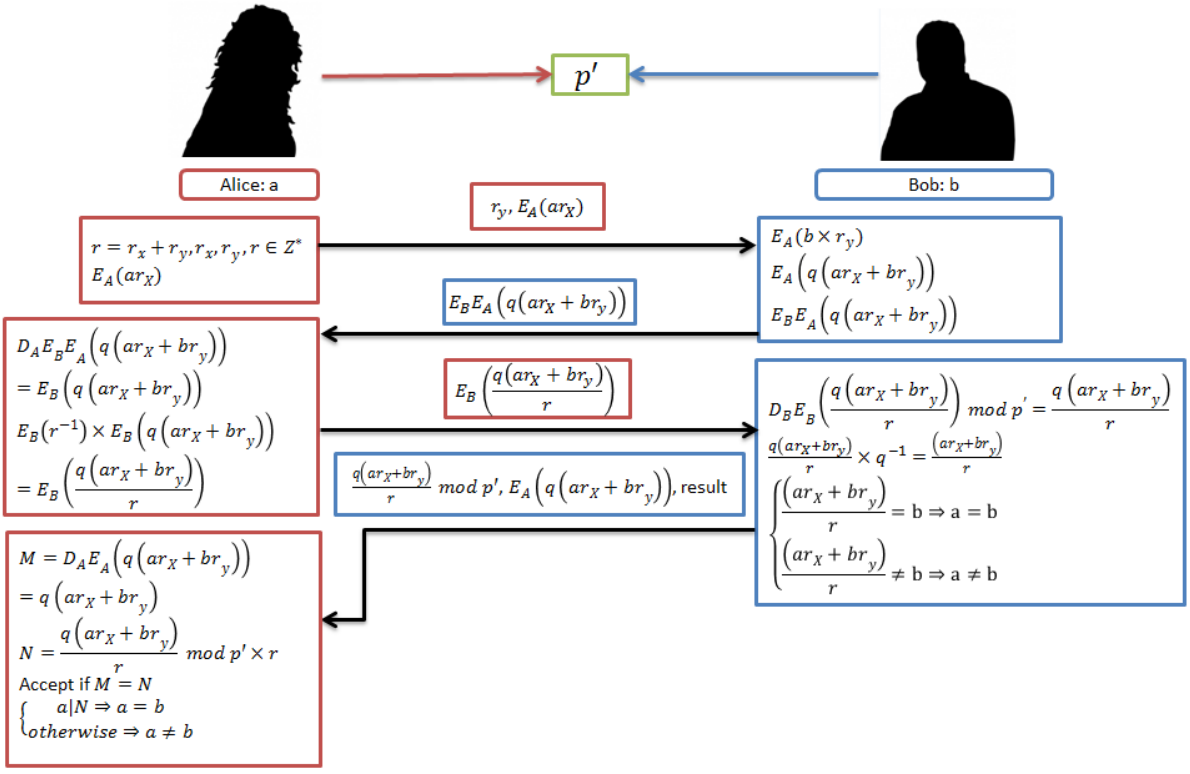

圖 5 邱姓協定流程圖

在邱姓學者的方案中，Alice 跟 Bob 可以利用協定在不洩漏彼此私密資訊的情況下，達到雙方相等性的驗證，而且不需要任何第三方的協助。如此一來，便不需要擔心第三方與 Alice 或 Bob 其中任何一方做出共謀的動作。儘管如此，但邱姓學者的協定卻需要使用到滿足可交換式與雙重同態的加解密演算法。

對於雙重同態的加解密演算法來說，2005 年 Boneh 在[2]所提到的加解密演算法可以做多次的加法運算跟一次的乘法運算，所以並不能算是完整的雙重同態加密。而邱姓學者所採用的演算法，是 Gentry[12]在 2009 年所發表的全同態加密演算法，雖然可以做任意次數的加法與乘法同態，但是所需的時間成本太高，在實作上有一定的困難性。而對可交換式加解密演算法來說，目前的文獻中除了邱姓學者提到的 Gentry 加密演算法外，並沒有其他符合可換式加密的演算法，多半都是更改既有的加密演算法使其符合可換式

加密性質，但往往都導致安全性下降、無法正確解密。

綜合上述，雖然邱姓學者的協定確實可以達到雙方相等性驗證的目的，但所採用的加密演算法的限制較多，且時間成本較高。本研究也是基於這樣的原因，想要設計一個協定來改善加密演算法的限制以及時間的成本。在本研究的協定中，所採用的是 ElGamal 加密演算法，不但不需要額外的限制也同時降低所需的時間成本，增加了協定可被實作以及被應用在實務上的可能性。本研究所提出的協定將會在第 3 章作更詳細的介紹。

2.4 模糊傳輸 (Oblivious Transfer)

因為本研究在最後完整的協定中會加入模糊傳輸，利用模糊傳輸所符合的特殊性質達到資料傳輸的私密性，因此將在 2.4.1 對模糊傳輸做簡單的介紹，並於 2.4.2 介紹在本研究中所使用的有效率的模糊傳輸做出更詳盡的介紹。

2.4.1 模糊傳輸簡介

自從 1981 年 Rabin[23]提出模糊傳輸的主要概念之後，模糊傳輸在近十年來被視為密碼學上相當重要的一項訊息交換技術。基於模糊傳輸的特性，這個協定被套用在相當多的應用上面，例如電子商務、祕密訊息交換或是傳輸認證過的電子郵件或是線上文件等等。在 Rabin 所提出的模糊傳輸概念中，有一個傳送資料的傳輸者(Sender)，跟一個接收資料的接收者(Receiver)，我們可以將傳輸者視為伺服器端(Server)，而接收者視為客戶端(Client)，或稱做使用者端(User)。此概念中，傳輸者任意傳送一個訊息給接收者，而接收者只有二分之一的機會能夠得到這個訊息，另外二分之一的機會什麼都得不到，而傳送者對於接收者是否收到訊息一無所知。

在這個概念被提出之後，Rabin 在中提出了二選一(2-out-of-1)的模糊傳輸協定，在這個協定中，傳送端擁有兩個位元b_1, b_2，而接收端可以挑選他所想要從傳送端得到的位元，但他只有二分之一的機會可以得到他所想要的位元，另外二分之一的機會得到他不想要的位元。在這個協定中，傳送端是無法得知接收端收到的位元為何，而接收端只會

收到一個位元而已。

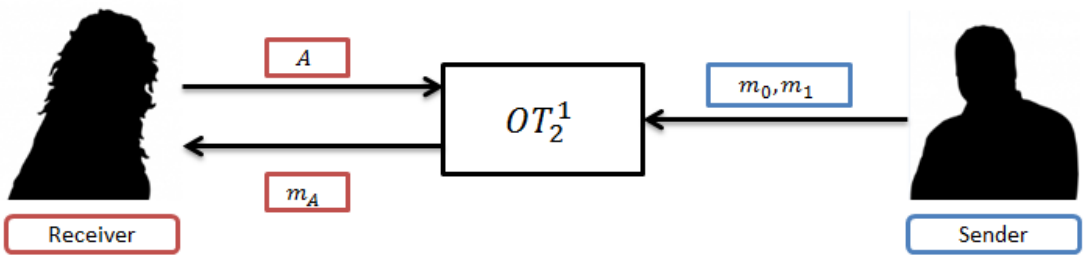

圖 6 二選一模糊傳輸

而在二選一的模糊傳輸協定之後，Brassard 跟 Cre'peau[4]將二選一的協定擴展成 n 選一的模糊傳輸協定，顧名思義就是傳送端擁有 n 個訊息$b_1, \ldots, b_n$，而接收端可以挑選他所想要從傳送端得到的其中一個訊息，而他只有 n 分之一的機會可以得到他所想要的訊息，在這個協定中，傳送端一樣是無法得知接收端收到的訊息為何，而接收端只會收到一個訊息而已。

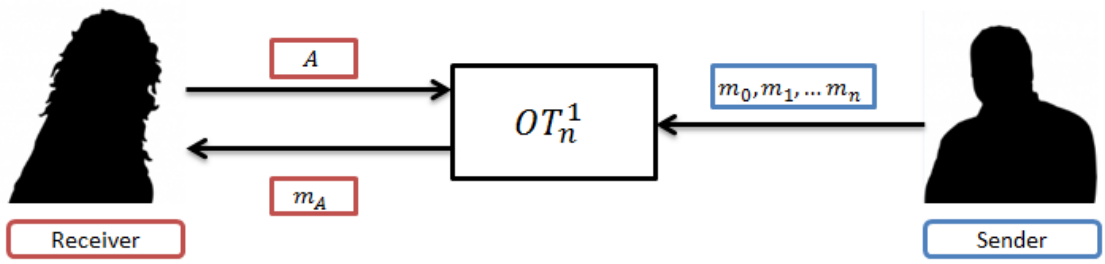

圖 7 n 選一模糊傳輸

在這之後，Chu 和 Tzeng[8]在 2005 年發表了 n 選 t 的模糊傳輸協定，在 n 選 t 的協定裡，傳送者握有 n 個訊息，接收者可以選擇索取其中 t $(t < n)$個訊息，同樣的，傳送端一樣是無法得知接收端收到的訊息是哪 t 個訊息，而接收端只能獲得他所索取的 t 個訊息，而不能夠獲得其他的$n - t$個訊息。另外，在 n 選 t 的模糊傳輸協定中，只需將傳送者所握有的訊息量以及接收者所索取的訊息量，即可將協定簡化並滿足 n 選一或是二選一的模糊傳輸。

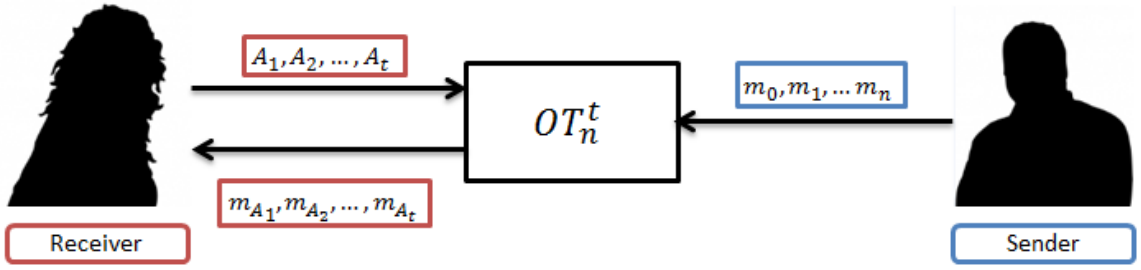

圖 8 n 選 t 模糊傳輸

除了上述所提到的模糊傳輸協定之外，尚有許多基於基本協定所提出的模糊傳輸協定被專家學者所發表出來，像是非交談式模糊傳輸[5]、可重複使用的模糊傳輸[17]、可驗證的模糊傳輸[19]等等。另外，在 1999 年更有由 Crescenzo、Ostrovsky 與 Rajagopalan [10]所提出的條件式模糊傳輸協定(Conditional Oblivious Transfer, COT)，這種條件式的模糊傳輸利用術語判斷式(predicate)，假如判斷式成立，接收者便可以從傳送者端收到索取的訊息，否則將無法獲得任何的訊息。另外，傳送者在整個協定過程都不會知道判斷式的成立與否。之後更有許多專家學者提出改良過的 COT 協定，像是強健型條件式模糊傳輸協定(Strong Conditional Oblivious Transfer, SCOT)[3]、條件式模糊群播協定(Conditional Oblivious Cast, COC)[9]。

綜合上述，無論是任何一種模糊傳輸協定，都必須要滿足以下的三個安全需求：

1. 傳輸的正確性：只要傳送者跟接收者按照協定步驟進行，接收者就可以得到精確地得到他所索取的訊息。
2. 接收者的隱私：對接收者而言，他可以選擇他所要索取的訊息，無論是二選一、n 選一還是 n 選 t，傳送者都沒有辦法得知接收者所索取的訊息項目為何，以保障接收者的隱私。
3. 傳送者的隱私：在協定執行完成後，接收者只能獲得他所索取的訊息，對於其他沒有索取的訊息是一無所知，縱使接收者有無窮的計算能力(unconditional computation power)也沒辦法得知，以保障傳送者的隱私。

在本論文中所使用的模糊傳輸方案，是由 Naor 與 Pinkas 在 Efficient Oblivious Transfer

Protocols[20]中所提出的方案(以下簡寫為 EOT)，分別有二選一以及 n 選一的方案，將在以下的小節做出詳細的介紹。

2.4.2 有效率的模糊傳輸協定(Efficient Oblivious Transfer, EOT)

首先，在提到 EOT 中提到的模糊傳輸協定是在Z_q這個群做運算，這個群的序為質數(prime order)，更特別的是G可以是一個Z_p^*的子群，其序為$q, q|p-1$。定義g為生成元，並具有 Computational Diffie-Hellman assumption(CDH assumption)。而這個協定利用了一個函數 H，這個函數被假設為一個隨機預言(random oracle)。

2.4.2.1 二選一模糊傳輸 (1-out-of-2)

1. 傳送者隨機選取一個亂數$r \in Z_q$，以及公布一個亂數$C \in Z_q$，同時計算C^r與g^r。
2. 接收者隨機選取一個亂數$1 \leq k \leq q$，計算公鑰$PK_\sigma = g^k, PK_{1-\sigma} = \frac{C}{PK_\sigma}, \sigma = \{0,1\}$，接著傳送$PK_0$給傳送者。
3. 傳送者在接收到PK_0之後，計算$(PK_0)^r$以及$(PK_1)^r = \frac{C^r}{(PK_0)^r}$，接著傳送$g^r$與對兩個明文$M_0, M_1$分別加密後的資訊$E_0 = H((PK_0)^r, 0) \oplus M_0$以及$E_1 = H((PK_1)^r, 1) \oplus M_1$給接收者。
4. 接收者收到傳送者傳來的資訊之後，計算$H((g^r)^k, \sigma) = H((PK_\sigma)^r, \sigma)$並與$E_0, E_1$做$\oplus$運算即可得到索求的明文。

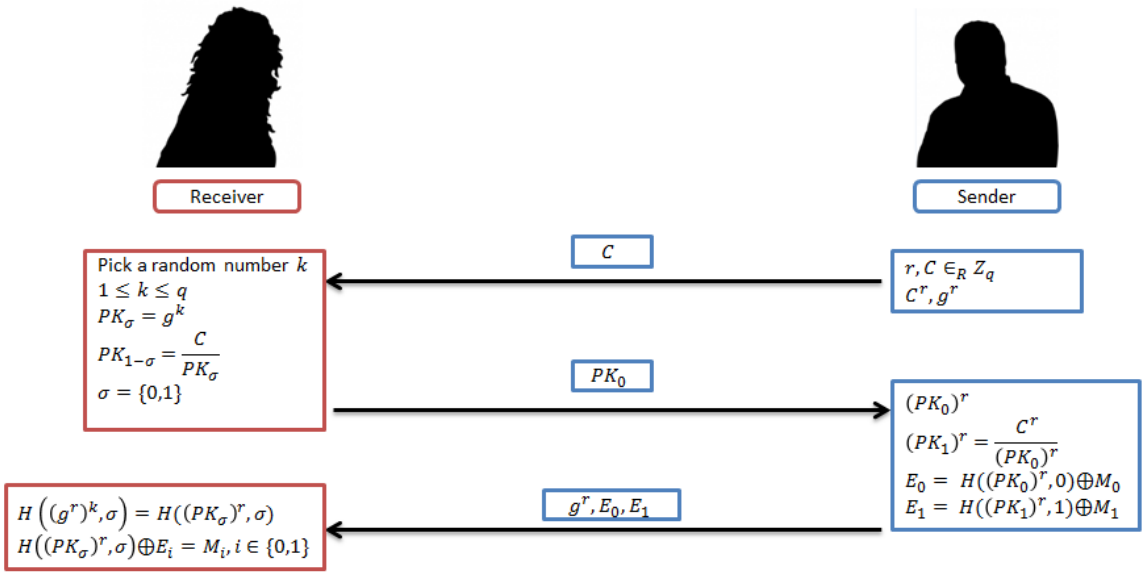

圖 9 二選一模糊傳輸協定流程圖

2.4.2.2 N 選一模糊傳輸 (1-out-of-N)

1. 傳送者隨機選取一個亂數$r \in Z_q$，以及公布$N-1$個亂數$C_1, C_2, \dots, C_{N-1} \in Z_q$，同時計算$(C_i)^r, 1 \le i \le N-1$與$g^r$。接著將$C_1, C_2, \dots, C_{N-1}$與$g^r$做完傳送者的公鑰傳送給接收者。
2. 接收者隨機選取一個亂數$1 \le k \le q$，計算$PK_\sigma = g^k$。如果$\sigma \neq 0$則計算$PK_0 = \frac{C_\sigma}{PK_\sigma}$，接著傳送$PK_0$給傳送者，並且同時也運算$(g^r)^k = (PK_\sigma)^r$作為解密的金鑰。
3. 傳送者在接收到PK_0之後，計算$(PK_0)^r$以及$(PK_i)^r = \frac{(C_i)^r}{(PK_0)^r}, 1 \le i \le N-1$，並且選取一個隨機的字串$R$，這個字串的長度必須夠長，必須大於$2\log n$位元。接著對於每個明文$M_0, M_1 \dots M_{N-1}$分別加密得到$E_i = H((PK_i)^r, R, i) \oplus M_i$，最後傳送$E_i$以及字串$R$給接收者。
4. 接收者收到傳送者傳來的資訊之後，計算$H((PK_\sigma)^r, R, \sigma)$並與$E_i$做$\oplus$運算即可得到索求的明文。

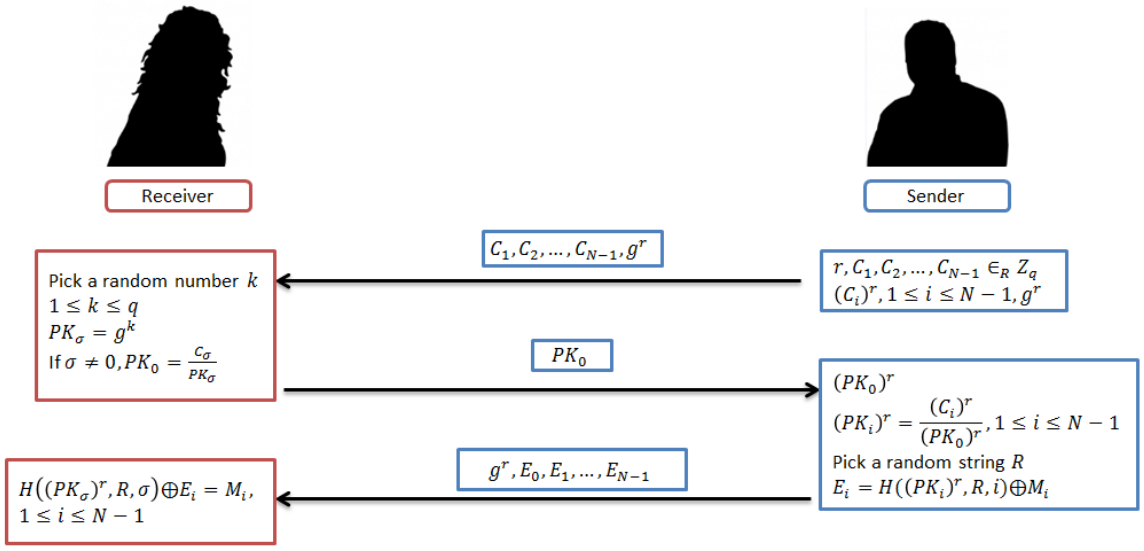

圖 10 N 選一模糊傳輸協定流程圖

2.5 離散對數問題 (Discrete Logarithm Problem, DLP)

離散對數問題因為其求解的困難性而被廣泛利用在密碼學的領域中，而 ElGamal 加密系統主要也是基於求解離散對數這個難問題，換句話說，要破解 ElGamal 加密系統，以目前所知的手法而言，唯一的方法就是去破解離散對數問題，然而離散對數問題又是一個什麼樣的難問題，接著我們會有詳細的定義與介紹。

首先定義一個乘法群(multiplicative group)$(G,\cdot)$，接著對於一個元素$\alpha \in G$，α的序(order)為p，定義$\langle\alpha\rangle = \{\alpha^i \bmod p : 0 \le i \le p-1\}, \beta \in \langle\alpha\rangle$，而從這個定義很清楚可以看出來$\langle\alpha\rangle$是一個乘法循環群，其序為$p$。問題為是否可以有效率的找一個整數$a, 0 \le a \le p-1$，使得$\alpha^a \bmod p = \beta \bmod p$。由上述式子推得$a \bmod p - 1 = \log_\alpha \beta \bmod p - 1$，因此將此問題稱作β的離散對數問題，而到現今為止都被視為一個相當困難求解的難問題。

2.6 計算性迪菲-赫爾曼問題 (Computational Diffie-Hellman Problem, CDHP)

令p為一個大質數，$G \subseteq Z_p^*$為一個循環群其序為q，g是循環群G的生成元且$p = 2q + 1$，

在已知$g, g^a \bmod p, g^b \bmod p$，求解$g^{ab} \bmod p$的問題即稱作 CDH 難問題。

2.7 決定性迪菲-赫爾曼問題 (Decisional Diffie-Hellman Problem, DDHP)

令p為一個大質數，$G \subseteq Z_p^*$為一個循環群其序為q，g是循環群G的生成元且$p = 2q + 1$，在已知$g, g^a \bmod p, g^b \bmod p, g^c \bmod p$，判斷是否$c = ab$的問題即稱作 DDH 難問題。

2.8 語意安全 (Semantic Security)

2.8.1 語意安全介紹

語意安全是一個被廣泛利用在非對稱式金鑰加密(Asymmetric Key Encryption, AKE)演算法的安全定義。對於一個具備語意安全的非對稱式金鑰加密演算法來說，它必須能夠抵抗一個具備有限計算能力(computationally bounded)的攻擊者所發動的攻擊。這個攻擊者就算擁有密文以及與其對應的加密公鑰，也沒有辦法從密文獲得有關明文的任何資訊，換句話說，也就是能夠抵抗選擇明文攻擊(Chosen Plaintext attack, CPA)。

語意安全的概念最早是由 Goldwasser 和 Micali[14]在 1982 年所提出，而通常證明語意安全時會設計一個遊戲，假使能夠成功地贏得遊戲，則表示具備了破解語意安全的能力。以下說明這個遊戲的步驟：

1. 攻擊者知道加密方法與公鑰，並允許攻擊者在有限的多項式時間內產生任何數量的密文。
2. 攻擊者任意生成兩個長度相同的訊息m_0, m_1，並且將這兩個訊息傳給挑戰者(challenger)。
3. 挑戰者投擲一個公平的硬幣，若正面則加密m_0，反面則加密m_1。加密完之後將密文回傳給攻擊者。
4. 攻擊者猜中密文所對應的明文機率是$\frac{1}{2} + \varepsilon$，若$\varepsilon$的值微小到可以忽略(negligible)，則稱攻擊者無法破解語意安全。反之，若ε的值無法忽略不計，則攻擊者攻擊成功的破解了語意安全。

本論文的加密系統是採用 ElGamal 的加密系統，在下一個小節將會以 ElGamal 為例，說明 ElGamal 是符合語意安全的。

2.8.2 ElGamal 的語意安全

為了說明 ElGamal 符合語意安全，我們如上個小節所說創建一個遊戲：

1. 攻擊者知道在這個遊戲中我們都使用 ElGamal 加密系統以及所使用的公鑰(G, q, g, y)，並允許攻擊者在有限的多項式時間內產生任何數量的密文。
2. 攻擊者任意生成兩個長度相同的訊息m_0, m_1，並且將這兩個訊息傳給挑戰者(challenger)。
3. 挑戰者投擲一個公平的硬幣，若正面則加密m_0，反面則加密m_1。此時回顧 2.2 的 ElGamal 加密演算法，挑戰者加密時會選取一個變數$r \in \{0, \dots, q-1\}$，並且計算$C_1 = g^r \bmod p$。接著選擇明文$m_i, i \in \{0,1\}$，生成 $C_2 = m \times y^r \bmod p$。加密完之後將密文$(C_1, C_2)$回傳給攻擊者。
4. 此時由於挑戰者在加密過程中，任意選取了一個亂數$r \in \{0, \dots, q-1\}$，使得攻擊者並沒有辦法從已知條件(G, q, g, y)計算出密文(C_1, C_2)究竟是由m_0, m_1哪一個訊息所加密的，確保了 ElGamal 的語意安全。

基於 DDHP，ElGamal 可被證明滿足語意安全並且有效的抵抗選擇明文攻擊(CPA)。另外，儘管沒有嚴格的證明，但一般推測 ElGamal 是能夠抵抗選擇密文攻擊(CCA) 的。以下便證明如何以 DDHP 證明 ElGamal 符合語意安全。假設攻擊 ElGamal 演算法的攻擊者為AT_1，攻擊 DDH 的攻擊者為AT_2，以及一個挑戰者C(challenger)。

1. C將$g, g^a \bmod p, g^b \bmod p$以及$T = g^{ab}$傳送給$AT_2$，$AT_2$將$g, g^a \bmod p$傳給$AT_1$，其中$g^a$作為 ElGamal 加密的公鑰。
2. AT_1選擇訊息M_0, M_1，並且將M_0, M_1傳送給AT_2。
3. AT_2在收到M_0, M_1後，任意選擇一個訊息做加密得到$C_1 = g^b, C_2 = M_\gamma T, \gamma \in \{0,1\}$，並將$(\mathrm{C}_1, \mathrm{C}_2)$傳回給$AT_1$。

4. AT_1收到(C_1, C_2)後，猜測被加密的訊息是$M'_\gamma, \gamma \in \{0,1\}$回傳給$AT_2$。
5. 若$M'_\gamma = M_\gamma$，表示AT_1挑戰成功，成功的破解了 ElGamal 的語意安全，而同時AT_2從AT_1能夠猜出正確的訊息推論出當初所加密的(C_1, C_2)為合法的 ElGamal 加密，因此得知$C_2 = M_\gamma \times (g^a)^b = M_\gamma T \Rightarrow T = g^{ab}$。如此一來$AT_2$便利用了$AT_1$破解了 DDHP。但已知 DDHP 為難問題，因此由逆否命題可知：若 DDHP 無法被破解，則 ElGamal 無法被破解。所以證明 ElGamal 是符合語意安全的。

國立政治大學 National Chengchi University

3. 研究方法

本論文的研究主要目的是為了解決在章節 2.3.3 所提到的幾個問題點，重新設計一個雙方祕密計算為架構的協定。本研究提出的協定，採用的是 ElGamal 加密演算法，移除了邱姓學者在之前文章中需要對加密演算法的限制，以及減少了原本因為使用 Gentry 的加密演算法所造成的時間成本。同時搭配模糊傳輸協定，讓本研究的相等性驗證協定能夠應用在更多的實務面上。接下來的章節將從基礎的協定開始介紹，一步步探討，最後深入到完整的協定。

3.1 基礎協定

3.1.1 基礎協定介紹

為了解決比較雙方私密資訊相等性的問題，本研究先設計並提出了一個基礎的協定，在不揭露雙方私密資訊的條件下，比較 Alice 跟 Bob 兩個人的私密資訊是否相等。假設 Alice 擁有私密資訊 a，Bob 擁有私密資訊 b 以及私鑰x_b，選取一個夠大的質數p與生成元g，配合私鑰計算出公鑰$PK_B = y_b = g^{x_b} \bmod p$，接著以下列步驟比較 Alice 跟 Bob 的 a 與 b 是否相等。由於文中將會有許多的模(mod)運算，之後文章中的模運算都將以$[x]_y$代表 $x \bmod y$。另外$[(x_1, x_2)]_y$代表 $(x_1 \bmod y, x_2 \bmod y)$。

1. Alice 任選一個亂數 t，並將 a 做$t \times a \bmod p = [t \times a]_p$之後傳送給 Bob。
2. Bob 接收到資訊後，計算$\left[\frac{[t\times a]_p}{b}\right]_p = \left[\frac{ta}{b}\right]_p$，並且用 Bob 自己的公鑰$PK_B$以 ElGamal 加密系統加密運算後，得到$\left[E_{PK_B}\left(\frac{ta}{b}\right)\right]_p = (C_1, C_2) = \left[\left(g^{r_1}, \frac{ta}{b} \times {y_b}^{r_1}\right)\right]_p$，再將$C_2$回傳給 Alice。
3. Alice 收到 (C_1, C_2) 後，對所收到的密文做以下兩個步驟的運算。

   (1) $C_2' = [C_2 \times t^{-1}]_p = \left[\frac{a}{b} \times {y_b}^{r_1}\right]_p$

(2) $(C_1'', C_2'') = (g^l, C_2'^l) = \left[\left(g^l, \left(\frac{a}{b}\right)^l \times {y_b}^{r_1 \times l}\right)\right]_p$

接著再將結果回傳給 Bob，其中l為 Alice 自選的亂數。

4. Bob 在收到(C_1'', C_2'')之後，先利用私鑰x_b與之前自選的亂數r_1計算$[(g^{r_1 \times l})^{x_b}]_p = [{y_b}^{r_1 \times l}]_p$，接著計算$\left[\left(\left(\frac{a}{b}\right)^l \times {y_b}^{r_1 \times l}\right) \times ({y_b}^{r_1 \times l})^{-1}\right]_p = \left[\left(\frac{a}{b}\right)^l\right]_p$，若$\left[\left(\frac{a}{b}\right)^l\right]_p = 1$，則表示$\left(\frac{a}{b}\right) = 1$，也就推論出$a = b$，否則$a \neq b$。最後，Bob 在得知相等與否之後，發送訊息告訴 Alice 告知雙方私密資訊是否相等。

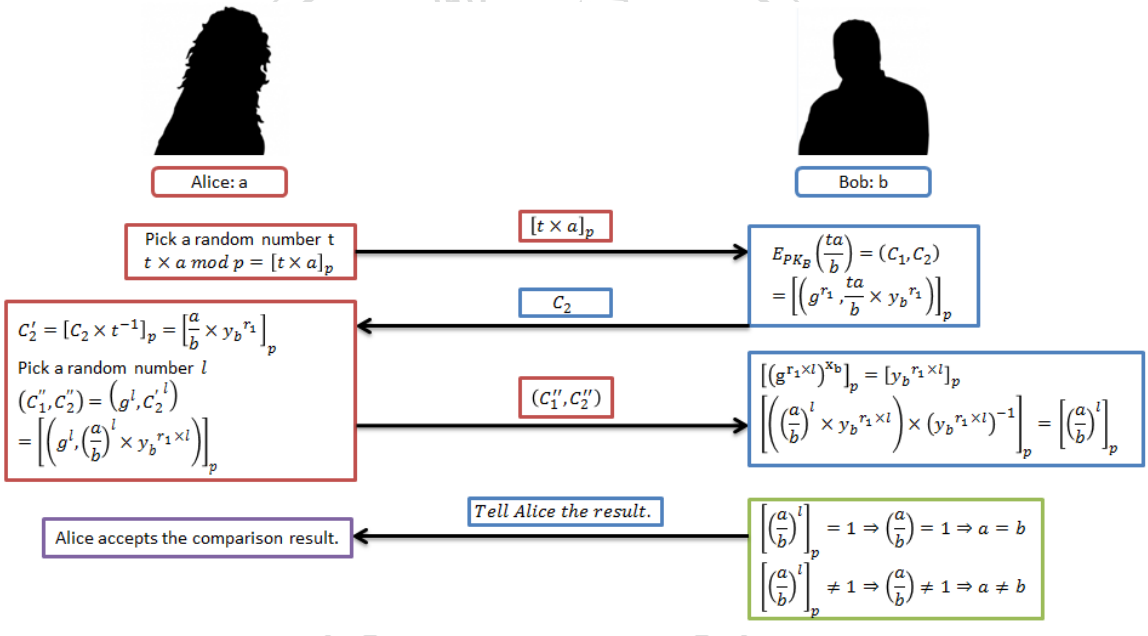

圖 11 基礎協定流程圖

3.1.2 基礎協定討論

在這個基礎的協定裡，只要 Alice 跟 Bob 找著步驟完成協定，便可以達到判斷雙方私密資訊是否相等的目的。而這樣的基礎協定，可以被利用在用戶端與伺服器(client & server)這樣的環境。舉例來說，假設 Alice 為用戶端，Bob 作為伺服器端，Alice 要登入伺服器時，Bob 去驗證 Alice 所傳送來的通行碼(password)是否與當初註冊的相同，如果相同，則允許登入，否則拒絕登入。若使用這個協定所架構出來的環境，則每個用戶所註冊的

資訊就都可以以密文的形式儲存在伺服器端，而伺服器端並不需要解密即可驗證登入者的身分。就算有任何駭客成功破解伺服器的防護，獲取到的用戶資料也全都是密文，比起傳統的密碼儲存方式來的安全有保障。

儘管基礎的協定已經達到判斷雙方私密資訊是否相等的目的，但是仍然存在一個問題。問題在於基礎的協定中，最後一個步驟是由 Bob 來告訴 Alice 是否相等，Alice 只能夠相信 Bob 所告知的最後結果，並沒有辦法驗證 Bob 是否有說謊，這對 Alice 來說是不太公平的。為了解決這個問題，便在基礎的協定中增加了一些驗證的機制，讓 Alice 也能夠驗證 Bob 所告知的最後結果是否正確無誤，達到雙方相等性驗證，以確保雙方的公平性。

3.2 雙方相等性驗證之協定

3.2.1 雙方相等性驗證協定介紹

為了解決上述所提到的雙方驗證問題，本研究接著設計了一個具備雙方驗證功能的協定，除了用來比較 Alice 跟 Bob 兩個人的私密資訊是否相等之外，更可以達到雙方都能夠驗證結果是否正確無誤。

1. Alice 任選一個亂數 t，並將 a 做$t \times a \bmod p = [t \times a]_p$之後傳送給 Bob。
2. Bob 接收到資訊後，計算$\left[\frac{[t\times a]_p}{b}\right]_p = \left[\frac{ta}{b}\right]_p$，並且用自己的公鑰$PK_B = y_b = g^{x_b} \bmod p$以 ElGamal 加密系統加密運算後，得到$\left[E_{PK_B}\left(\frac{ta}{b}\right)\right]_p = (C_1, C_2) = \left[\left(g^{r_1}, \frac{ta}{b} \times {y_b}^{r_1}\right)\right]_p$，再將$(C_1, C_2)$回傳給 Alice。
3. Alice 收到 (C_1, C_2) 後，在這個步驟又分成三個部分：
   I. 將$[t^{-1}]_p$以 Bob 的公鑰加密，並選取一個亂數l_1根據 ElGamal 加密系統所具備的乘法同態性質做以下兩個步驟的運算：

   (1) $E_{PK_B}(t^{-1}) \times E_{PK_B}\left(\frac{ta}{b}\right) = E_{PK_B}\left(\frac{a}{b}\right)$

(2) $\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1} \bmod p = \left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p$

II. 將從 Bob 接收而來的原始(C_1, C_2)，並選取一個亂數l_2做以下兩個步驟的運算。

(1) $C_2' = [C_2 \times t^{-1}]_p = \left[\frac{a}{b} \times {y_b}^{r_1}\right]_p$

(2) $(C_1'', C_2'') = \left(g^{l_2}, {C_2'}^{l_2}\right) = \left(g^{l_2}, \left(\frac{a}{b}\right)^{l_2} \times {y_b}^{r_1 \times l_2}\right)$

III. Alice 對i個 1 與任選的i個不等於 1 的變數k_i並以 Bob 的公鑰加密，將$\left(E_{PK_B}(1_1), \dots, E_{PK_B}(1_i), E_{PK_B}(k_1), \dots, E_{PK_B}(k_i), \left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p, (C_1'', C_2'')\right)$回傳給 Bob。其中，除了$(C_1'', C_2'')$以外的$2i+1$個數值順序是由 Alice 自訂的，這個順序的安排將會被利用在之後的驗證步驟。

4. Bob 在收到密文數列之後，運算分做兩大部分：

I. 在固定位置將(C_1'', C_2'')取出，先利用私鑰x_b與之前自選的亂數r_1計算$[(g^{r_1 \times l_2})^{x_b}]_p = [{y_b}^{r_1 \times l_2}]_p$，接著計算$\left[\left(\left(\frac{a}{b}\right)^{l_2} \times {y_b}^{r_1 \times l_2}\right) \times ({y_b}^{r_1 \times l_2})^{-1}\right]_p = \left[\left(\frac{a}{b}\right)^{l_2}\right]_p$，若$\left[\left(\frac{a}{b}\right)^{l_2}\right]_p = 1$，則表示$\left(\frac{a}{b}\right) = 1$，也就推論出$a = b$，否則$a \neq b$。

II. 對數列中 Alice 隨機擺放$2i+1$個元素做解密，其中$\left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p$解密之後得到$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$，對$E_{PK_B}(1_1), \dots, E_{PK_B}(1_i), E_{PK_B}(k_1), \dots, E_{PK_B}(k_i)$解密會得到$i$個 1 跟$i$個變數$k_i$，解密後的結果劃分為兩種：

(1) 如果1的個數多於變數的個數，表示$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p = 1 = \left(\frac{a}{b}\right)$，由此可以推論得到$a = b$。Bob 在知道雙方私密資訊相等後，接著回傳$(1_1, \dots, 1_i, k_1', \dots, k_i', 1)$給 Alice。

(2) 如果1的個數少於變數的個數，表示$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p \neq 1 \neq \left(\frac{a}{b}\right)$，由此可以推論得到$a \neq b$。Bob 在知道雙方私密資訊不相等後，接著回傳$\left(1_1, \dots, 1_i, k_1', \dots, k_i', \left[\left(\frac{a}{b}\right)^{l_1}\right]_p\right)$給

Alice。

最後，Bob 在經過雙重判斷得知相等與否之後，發送訊息告訴 Alice 告知雙方私密資訊是否相等。無論是上述兩種情況中的哪一種，Bob 都不會去更動數列內元素的順序，否則 Alice 將會在驗證步驟時發現 Bob 更動過數列。同時，Bob 在回傳數列的時候，會將原本數列中的$k_1, \dots, k_i$替換成$k_1', \dots, k_i'$。這麼做的原因是為了避免當$a \neq b$時，Alice 能夠就她自訂的數列位置找到$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$，進而得知 Bob 的私密資訊。

5. 當 Alice 收到 Bob 回傳的數列後，由於只有 Alice 知道當初傳送資料的順序，Alice 針對回傳的數列，檢查所有的 1 與變數位置是否正確，以檢驗 Bob 是否更改了數列中元素的位置或是企圖以竄改元素的方式欺騙 Alice。接著，Alice 再去對應的位置找到$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$，檢查是否為 1。若數列中元素位置皆正確而且沒有被竄改，且$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p = 1$，則表示 Bob 沒有企圖以更改資訊的方式欺騙 Alice，而且雙方的私密資訊是相等的。

國立政治大學 National Chengchi University

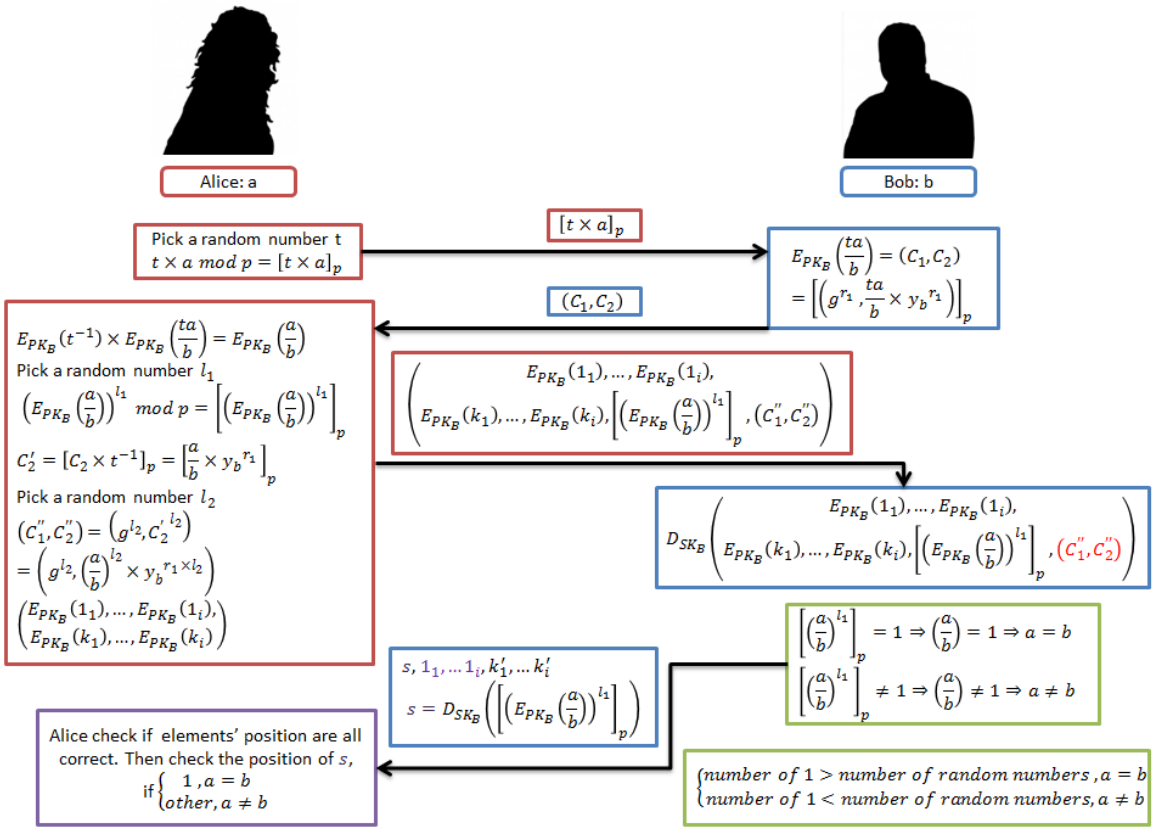

圖 12 雙方相等性驗證協定流程圖

3.2.2 雙方相等性驗證協定討論

在這個改良的方案中，不僅僅是雙方都知道私密資訊是否相等，同時也讓 Alice 可以驗證 Bob 是否誠實地告知結果，達到了雙方的相等性驗證。接著用一個簡單的例子來說明假如一個不誠實的 Bob 想要欺騙 Alice，成功機會是相當微小的。

在這個例子裡，假設 Alice 以 Bob 的公鑰加密了三個 1 以及三個變數k_1, k_2, k_3，並且以$\left(E_{PK_B}(1), E_{PK_B}(k_1), E_{PK_B}(k_2), E_{PK_B}(1), E_{PK_B}(k_3), \left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p, E_{PK_B}(1)\right)$的順序送出給 Bob。Bob 解密後會得到$\left(1, k_1, k_2, 1, k_3, \left[\left(\frac{a}{b}\right)^{l_1}\right]_p, 1\right)$，如同上小節所述，根據雙方私密資訊相等性與否，將分成兩種情況：

(1) 若雙方私密資訊相等，Bob 必須傳回$(1, k_1', k_2', 1, k_3', 1, 1)$給 Alice，此時若 Bob 想欺騙 Alice 說雙方資訊不相等，則必須將四個 1 當中的一個換成任意變數k_4，

所以有$\left(\frac{1}{3+1}\right)$的機率能夠成功欺騙 Alice。

(2) 若雙方私密資訊不相等，Bob 必須傳回$\left(1, k_1', k_2', 1, k_3', \left[\left(\frac{a}{b}\right)^{l_1}\right]_p, 1\right)$給 Alice，此時$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$是個不為 1 的數值，若 Bob 想欺騙 Alice 說雙方資訊相等，則必須將四個不為 1 的數值當中的一個換成 1，所以有$\left(\frac{1}{3+1}\right)$的機率能夠成功欺騙 Alice。

從上述這個簡單的例子中，可以推得當 Alice 選擇的數值為 n 個 1 跟 n 個變數時，Bob 成功欺騙 Alice 的機率就是$\frac{1}{\mathrm{n}}$。如此一來，Alice 可以利用 1 與變數 k 的數量來降低 Bob 欺騙成功的機率。換句話說，愈多的 1 與變數，Bob 欺騙成功的機會就愈小。

3.3 雙方相等性驗證搭配模糊傳輸之協定

3.3.1 雙方相等性驗證搭配模糊傳輸之協定介紹

在有了雙方相等性驗證的協定之後，這小節將介紹將雙方相等性驗證的協定與模糊傳輸做搭配，以應用於具資料庫的用戶端與伺服器(client & server with database)這樣的環境之下。利用關鍵字的雙方的相等性驗證判斷資料庫中是否有用戶端(接收者)需要的資料，若有用戶端(接收者)需要的資料，用戶端(接收者)與伺服器(傳送者)可利用 2.4.2.2 所提到的 n 取一模糊傳輸協定傳送及獲取資料，以保障用戶端(接收者)與伺服器(傳送者)的隱私。以下協定的步驟將以 Alice 作為用戶端(接收者)，Bob 作為伺服器(傳送者)，DB 表示資料庫，資料庫中有資料的關鍵字$b_0, \dots b_{n-1}$。而加解密金鑰的部分，Bob 的私鑰為x_b，公鑰為$y_b = g^{x_b} \bmod p$。

1. Alice 想要知道 DB 中是否有所想要的資料 a，於是任選一個亂數 t，並將 a 做$t \times a \bmod p = [t \times a]_p$之後傳送給 Bob。
2. Bob 接收到資訊後，對於每個資料庫中的關鍵字$b_0, \dots b_{n-1}$，分別計算$\left[\frac{[t\times a]_p}{b_i}\right]_p = \left[\frac{ta}{b_i}\right]_p, 0 \le i \le n-1$，並且用 Bob 的公鑰$PK_B = y_b = g^{x_b}$以 ElGamal 加密系統加密運

算後的結果，得到$\left[E_{PK_B}\left(\frac{ta}{b_0}\right)\right]_p,\ldots,\left[E_{PK_B}\left(\frac{ta}{b_{n-1}}\right)\right]_p=\left(C_{1_0},C_{2_0}\right),\ldots,\left(C_{1_{n-1}},C_{2_{n-1}}\right)=e_0,\ldots,e_{n-1}$，再將$e_0,\ldots,e_{n-1}$回傳給 Alice。

3. Alice 收到$e_0,\ldots,e_{n-1}$之後在這個步驟又分成三個部分：

   I. 先將$[t^{-1}]_p$以 Bob 的公鑰PK_B加密，並選取一個亂數l_1根據 ElGamal 加密系統所具備的乘法同態性質對所有的$e_j,0\le j\le n-1$做以下兩個步驟的運算：

   (1) $E_{PK_B}(t^{-1})\times E_{PK_B}\left(\frac{ta}{b_j}\right)=E_{PK_B}\left(\frac{a}{b_j}\right)$

   (2) $\left(E_{PK_B}\left(\frac{a}{b_j}\right)\right)^{l_1} mod\ p=\left[\left(E_{PK_B}\left(\frac{a}{b_j}\right)\right)^{l_1}\right]_p$

   II. 將從 Bob 接收而來的原始$e_0,\ldots,e_{n-1}=\left(C_{1_0},C_{2_0}\right),\ldots,\left(C_{1_{n-1}},C_{2_{n-1}}\right)$取出$C_{2_0},\ldots,C_{2_{n-1}}$並選取一個亂數$l_2$做以下兩個步驟的運算。

   (1) $C'_{2_j}=\left[C_{2_j}\times t^{-1}\right]_p=\left[\frac{a}{b}\times {y_b}^{r_j}\right]_p,0\le j\le n-1$

   (2) $\left(C''_{1_j},C''_{2_j}\right)=\left(g^{l_2},{C'_{2_j}}^{l_2}\right)=\left(g^{r_j\times l_2},\left(\frac{a}{b}\right)^{l_2}\times {y_b}^{r_j\times l_2}\right),0\le j\le n-1$

   III. Alice 對i個 1 與任選的i個不等於 1 的變數k_i並以 Bob 的公鑰加密，將

   $$\begin{pmatrix}E_{PK_B}(1_1),\ldots,E_{PK_B}(1_i),E_{PK_B}(k_1),\ldots,E_{PK_B}(k_i),\\ \left[\left(E_{PK_B}\left(\frac{a}{b_0}\right)\right)^{l}\right]_p,\ldots,\left[\left(E_{PK_B}\left(\frac{a}{b_{n-1}}\right)\right)^{l}\right]_p,\\ \left(C''_{1_0},C''_{2_0}\right),\ldots,\left(C''_{1_{n-1}},C''_{2_{n-1}}\right)\end{pmatrix}$$

   回傳給 Bob。其中，除了$\left(C''_{1_0},C''_{2_0}\right),\ldots,\left(C''_{1_{n-1}},C''_{2_{n-1}}\right)$外，其他$2i+n$個數值的順序是由 Alice 自訂的，這個順序的安排將會被利用在之後的驗證步驟以及模糊傳輸上。

4. Bob 在收到密文數列之後，運算分做兩大部分：

   I. 在固定位置將$\left(C''_{1_0},C''_{2_0}\right),\ldots,\left(C''_{1_{n-1}},C''_{2_{n-1}}\right)$取出，利用私鑰$x_b$與先前自選的亂數$r_j$計算$\left[\left(g^{r_j\times l_2}\right)^{x_b}\right]_p=\left[{y_b}^{r_j\times l_2}\right]_p,0\le j\le n-1$，接著計算$\left[\left(\left(\frac{a}{b_j}\right)^{l_2}\times {y_b}^{r_j\times l_2}\right)\times\right.$

$\left(y_b{}^{r_j \times l_2}\right)^{-1}\Big]_p = \left[\left(\frac{a}{b_j}\right)^{l_2}\right]_p, 0 \le j \le n-1$，若$\left[\left(\frac{a}{b_j}\right)^{l_2}\right]_p = 1$，則表示$\left(\frac{a}{b_j}\right) = 1$，也就推論出$a = b$，否則$a \neq b$。

II. 對數列中的每個元素做解密，其中$\left[\left(E_{PK_B}\left(\frac{a}{b_j}\right)\right)^{l_1}\right]_p, 0 \le j \le n-1$解密之後得到$\left[\left(\frac{a}{b_j}\right)^{l_1}\right]_p, 0 \le j \le n-1$，對$E_{PK_B}(1_1), \dots, E_{PK_B}(1_i), E_{PK_B}(k_1), \dots, E_{PK_B}(k_i)$解密會得到$i$個 1 跟$i$個變數$k_i$，解密後的結果劃分為兩種：

(1) 如果有$i+1$個 1 與$i+n-1$個變數，表示$\left[\left(\frac{a}{b_0}\right)^{l_1}\right]_p, \dots, \left[\left(\frac{a}{b_{n-1}}\right)^{l_1}\right]_p$中的其中一項$\left[\left(\frac{a}{b_\sigma}\right)^{l_1}\right]_p = 1 = \left(\frac{a}{b_\sigma}\right)$，由此可以推論得到$a = b_\sigma$。此時 Bob 得知資料庫中有 Alice 所搜尋的關鍵字，接著 Bob 便根據之前在 2.4.2.2 所提到的 n 取一模糊傳輸協定的步驟(1)，產生亂數$C_1, \dots, C_{n-1}$與亂數r，計算g^r以及$(C_i)^r, 1 \le i \le n-1$，並回傳$\left(1_1, \dots, 1_i, k'_1, \dots, k'_i, \left[\left(\frac{a}{b_0}\right)^{l_1}\right]_p, \dots 1, \dots, \left[\left(\frac{a}{b_{n-1}}\right)^{l_1}\right]_p, C_1, \dots, C_{n-1}, g^r\right)$給 Alice。

(2) 如果有i個 1 與$i+n$個變數，表示$\left[\left(\frac{a}{b_0}\right)^{l_1}\right]_p, \dots, \left[\left(\frac{a}{b_{n-1}}\right)^{l_1}\right]_p$中，每一項都不等於 1，由此可以推論得到$a \neq b_j, 0 \le j \le n-1$。此時 Bob 得知資料庫中沒有 Alice 所搜尋的關鍵字，接著回傳$\left(1_1, \dots, 1_i, k'_1, \dots, k'_i, \left[\left(\frac{a}{b_0}\right)^{l_1}\right]_p, \dots, \left[\left(\frac{a}{b_{n-1}}\right)^{l_1}\right]_p\right)$給 Alice。

最後，Bob 在得知相等與否之後，發送訊息告訴 Alice 告知雙方私密資訊是否相等。無論是上述兩種情況中的哪一種，Bob 都不會去更動數列內元素的順序，否則 Alice 將會在驗證步驟時發現 Bob 更動過數列。同時，Bob 在回傳數列的時候，會將原本數列中的$k_1, \dots, k_i$替換成$k'_1, \dots, k'_i$。這麼做的原因是為了避免當$a \neq b$時，Alice 能夠就她自訂的數列位置找到$\left[\left(\frac{a}{b_j}\right)^{l_1}\right]_p, 0 \le j \le n-1$，進而得知 Bob 的私密資訊。

5. 當 Alice 收到 Bob 回傳的數列後，由於只有 Alice 知道當初傳送資料的順序，Alice 針對回傳的數列，檢查所有的 1 與變數位置是否正確，以檢驗 Bob 是否更改了數列

中元素的位置或是企圖以竄改元素的方式欺騙 Alice。接著，Alice 再去對應的位置找到所有的$\left[\left(\frac{a}{b_j}\right)^{l_1}\right]_p, 0 \le j \le n-1$，將他們按照相對位置重新排列編號，接著分成兩個情況：

(1) 若取出的$\left[\left(\frac{a}{b_0}\right)^{l_1}\right]_p, \ldots, \left[\left(\frac{a}{b_{n-1}}\right)^{l_1}\right]_p$中沒有任何一個是 1，表示 DB 內沒有 Alice 所要的資訊，協定完成。

(2) 若取出的$\left[\left(\frac{a}{b_0}\right)^{l_1}\right]_p, \ldots, \left[\left(\frac{a}{b_{n-1}}\right)^{l_1}\right]_p$中的其中一項$\left[\left(\frac{a}{b_\sigma}\right)^{l_1}\right]_p = 1$，表示 DB 中有 Alice 所搜尋的關鍵字，也就是有 Alice 所想要的資訊。Alice 便從數列中取得相對應的C_σ以及選取一個亂數k，如果$\sigma \neq 0$則計算$PK_0 = \frac{C_\sigma}{PK_\sigma}$，並傳送$PK_0$給 Bob，同時也運算$(g^r)^k = (PK_\sigma)^r$作為解密的金鑰，協定繼續。

6. Bob 在接收到PK_0之後，計算$(PK_0)^r$以及$(PK_i)^r = \frac{(C_i)^r}{(PK_0)^r}, 1 \le i \le n-1$，並且選取一個隨機的字串$R$，接著對於每個資料庫中的資料$M_0, M_1 \ldots M_{N-1}$分別加密，若依照有效率模糊傳輸協定的步驟，將得到$E_i = H((PK_i)^r, R, i) \oplus M_i$，但是這會造成一個相當嚴重的問題。由於$H((PK_i)^r, R, i)$的值是固定長度，因此在跟$M_i$做⊕的運算後，所能夠隱蔽的資料只有低位元的部分，高位元並沒有被隱蔽到，導致 Alice 可以得知其它非她所選取資料的部分資訊，這是違背模糊傳輸所需保障的〝傳送者的隱私〞。因此，本研究將以乘法運算代替⊕的運算，犧牲少量的運算時間，以確保傳送者的隱私。所以 Bob 最後是傳送$E_i = H((PK_i)^r, R, i) \times M_i$以及字串$R$給 Alice。
7. Alice 收到 Bob 傳來的資訊之後，計算$[H((PK_\sigma)^r, R, \sigma)]^{-1}$並與$E_i$做乘法運算即可得到所想要的資訊，協定完成。

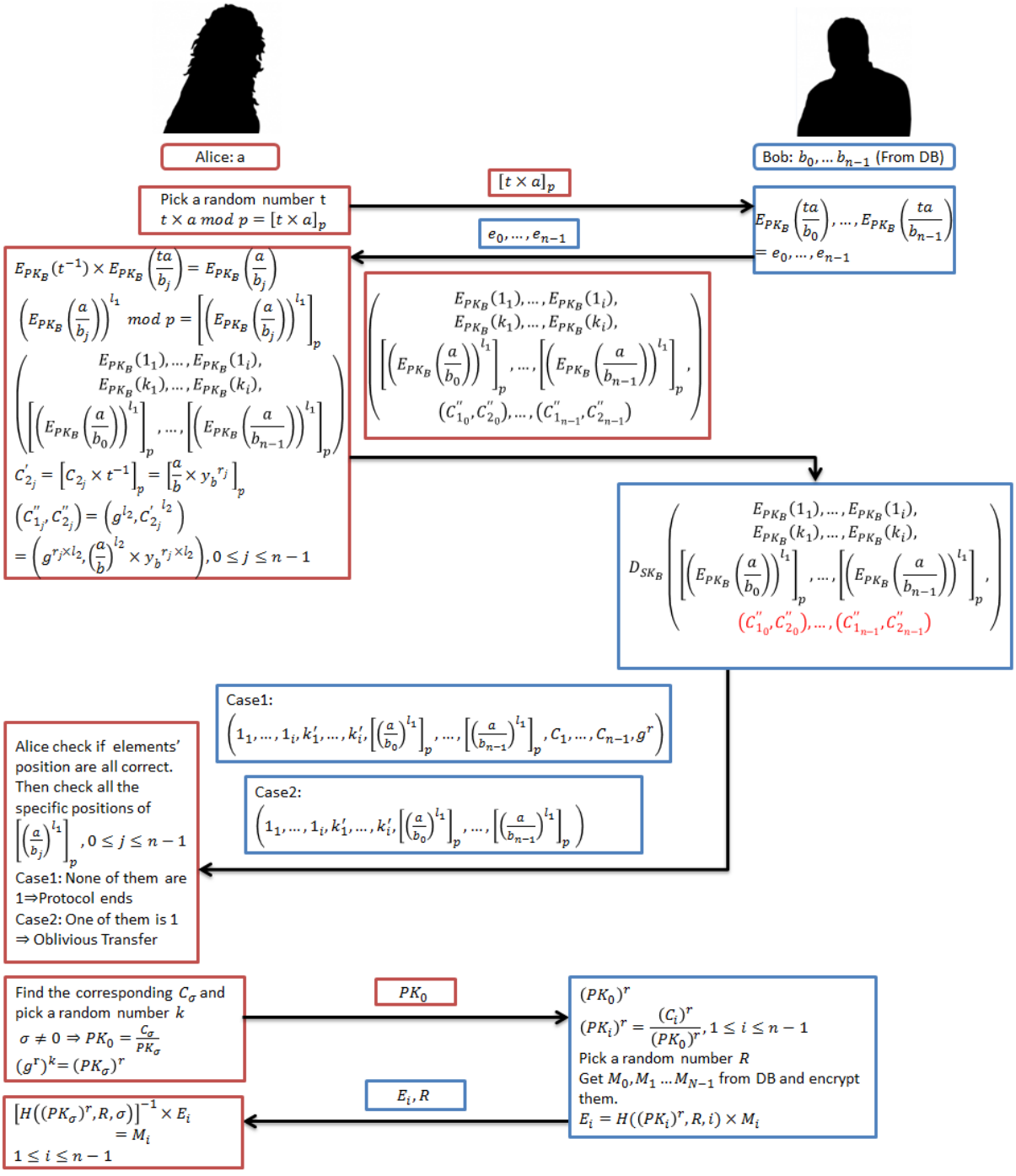

圖 13 雙方相等性驗證搭配模糊傳輸之協定流程圖

3.3.2 雙方相等性驗證搭配模糊傳輸之協定討論

在 3.2 雙方相等性驗證之協定中，已經介紹了雙方如何利用協定來達到比較雙方私密資訊是否相等的目的。在 3.3 中，情境已經從單純的比較雙方私密資訊是否相等擴展到私密資訊查詢的層級。在這個情境底下，Alice 跟 Bob 所扮演的角色已經不再只是想要比

較雙方私密資訊是否相等的二人。Bob 成了一個擁有一個以上資料(檔案)的一方，每份資料(檔案)都有相對應的關鍵字，而 Alice 想在不洩漏任何資訊的狀況下，利用關鍵字搜尋，得知 Bob 是否擁有她所想要的相關資料，若 Bob 有 Alice 想要的資料，Alice 可以利用模糊傳輸獲得她所搜尋的資料。相對的，若 Bob 沒有 Alice 所搜尋的資料，協定也能夠確保 Alice 無法獲得任何額外的資訊，以保護 Bob 擁有的所有資料(檔案)安全及隱私。

更仔細一點的探討，在雙方相等性驗證搭配模糊傳輸之協定中，比較關鍵字利用的是雙方相等性驗證協定，因此 Alice 跟 Bob 雙方都可以判斷並驗證關鍵字的相等性。換句話說，若將 Bob 視為伺服器端，Alice 視為用戶端，表示伺服器端可以知道下關鍵字搜尋的用戶端，是否有在資料庫中找到想要的資料。而對於用戶端來說，也可以驗證伺服器端是否有欺騙或是不誠實的行為，是一個相當公平環境。更進一步來說，當關鍵字比較不相等時，伺服器端是完全無法得知用戶所下的關鍵字為何，保障了用戶端的搜尋隱私。

接著，當用戶端得知關鍵字搜尋的結果後，可以選擇是否跟伺服器端拿取資料，大致上可以分成以下四種狀況：

表格 3 關鍵字判斷與用戶端索取資料與否配對結果

關鍵字是否相等	用戶端是否拿取資料	結果
是	是	利用模糊傳輸，用戶正確獲得資料
是	否	協定結束
否	是	模糊傳輸，確保用戶無法獲得任何資訊
否	否	協定結束

因為模糊傳輸的把關，用戶端只能在關鍵字比對相等且想要拿取資料的情境下獲得關鍵字所對應的資料，而且無法獲得其他額外的訊息。若是另外三種情況，用戶端是完全無法得知資料庫中任何資料的資訊，以保護伺服器端資料庫中資料的安全性及私密性。而

伺服器端在用戶拿取資料的前後，都無法得知用戶端想要的資料是哪一筆，確實的保障了用戶端獲取資料的隱私。

國立政治大學 National Chengchi University

4. 安全性分析

在介紹完一個又一個繁瑣的協定後，這個章節將對本論文所提出的協定，逐步的作安全性分析。由於本論文所提出的協定是結合了雙方相等性驗證與模糊傳輸，所以將分作三大部分作安全性的分析：在 4.1 對於基礎協定作安全性的分析，在基礎的協定的安全性分析之後，在 4.2 對由基礎協定所擴充的雙方相等性驗證作安全性的分析，最後 4.3 則是對模糊傳輸作安全性的說明。

4.1 基礎協定安全性分析

在本節將對本研究提出的基礎協定做出安全性的分析，以下將分別在章節 4.1.1 假設 A 為攻擊者與章節 4.1.2 假設 B 為攻擊者做出安全性的分析。

4.1.1 假設 A 為攻擊者的情況

這個小節假設 A 為攻擊者，針對協定中 A 有可能攻擊成功的步驟做出安全性的分析，說明本研究提出的協定是滿足了保障 B 的安全性與隱私性的目的。

1. B → A：C_2

   由協定步驟二可以得知，$C_2 = \frac{ta}{b} \times {y_b}^{r_1}$，因為$r_1$是 B 自選的亂數，因此 A 在收到$C_2$之後，無法得知${y_b}^{r_1}$為何，且對於所有的$\frac{ta}{b} \in \mathrm{Z}_\mathrm{p}^*$都存在一個相對應的 ${y_b}^{r_1}$滿足上式，因此 A 無法反推$\frac{ta}{b}$，進而保障 A 無法從C_2得到任何關於b的資訊。

2. A → B：(C_1'', C_2'')

   這個步驟比較特殊，要探討的是 A 是否可以藉由傳送更動過的假訊息達到欺騙 B 的目的。由協定的步驟三可以得知$(C_1'', C_2'') = \left(g^l, {C_2'}^l\right) = \left[\left(g^l, \left(\frac{a}{b}\right)^l \times {y_b}^{r_1 \times l}\right)\right]_p$，B 在收到$(C_1'', C_2'')$之後，若要解密必須先計算$[(g^{r_1 \times l})]_p$，此時$g^{r_1 \times l}$便同等於一把由 A 跟 B 共構的一把迪菲－赫爾曼密鑰(Diffie–Hellman key)。接著計算$[(g^{r_1 \times l})^{x_b}]_p =$

$[{y_b}^{r_1 \times l}]_p$，並利用所得到的結果解密計算得到$\left[\left(\left(\frac{a}{b}\right)^l \times {y_b}^{r_1 \times l}\right) \times ({y_b}^{r_1 \times l})^{-1}\right]_p = \left[\left(\frac{a}{b}\right)^l\right]_p$，由於是使用 B 的私鑰$x_b$與 B 之前自選的亂數$r_1$，所以如果 A 要試圖將$\left[\left(\frac{a}{b}\right)^l \times {y_b}^{r_1 \times l}\right]_p$竄改成$\left[1^l \times {y_b}^{r_1 \times l}\right]_p$，就必須知道當初 B 所選擇的亂數$r_1$為何，否則竄改後的密文就不會是個合法的密文，讓 B 可以藉由驗證發現 A 的不法行為，進而結束協定。

4.1.2 假設 B 為攻擊者的情況

這個小節假設 B 為攻擊者，針對協定中 B 有可能攻擊成功的步驟做出安全性的分析，說明本研究提出的協定是滿足了保障 A 的安全性與隱私性的目的。

1. A → B：$t \times a \bmod p = [t \times a]_p$

   協定的一開始 A 將自己的私密資訊a藉由乘上亂數t以及p的模運算，很明顯的可以知道對於所有的$a \in Z_p^*$都存在一個相對應的t滿足上式，因此 B 無法反推a，確保了 A 的私密資訊不會洩漏讓 B 知道，而且任何惡意的第三者從中擷取資訊也無從得知a為何。

2. A → B：(C_1'', C_2'')

   B 在收到$(C_1'', C_2'') = \left(g^l, {C_2'}^l\right) = \left[\left(g^l, \left(\frac{a}{b}\right)^l \times {y_b}^{r_1 \times l}\right)\right]_p$，可以計算$[(g^{r_1 \times l})^{x_b}]_p = [{y_b}^{r_1 \times l}]_p$，並利用求得的結果解密計算得到$\left[\left(\left(\frac{a}{b}\right)^l \times {y_b}^{r_1 \times l}\right) \times ({y_b}^{r_1 \times l})^{-1}\right]_p = \left[\left(\frac{a}{b}\right)^l\right]_p$，此時由於基於離散對數問題 B 是無法得知$l$為何值，在不知道$l$為何值的情況下，B 更不可能知道$\left(\frac{a}{b}\right)$的值為何。換句話說，B 無法藉由 A 回傳的訊息計算得知 A 的私密資訊a為何，確保 A 的私密資訊a不會洩漏讓 B 知道。

4.2 雙方相等性驗證安全性分析

由於在章節 3.3 所提到的雙方相等性驗證搭配模糊傳輸之協定較為複雜，且該協定中的

雙方相等性驗證部份是沿用章節 3.2 所提到雙方相等性驗證協定，因此這小節將只針對章節 3.2 的雙方相等性驗證協定作出安全性的分析。以下便於章節 4.2.1 假設 A 為攻擊者與章節 4.2.2 假設 B 為攻擊者做出安全性的分析。

4.2.1 假設 A 為攻擊者的情況

這個小節假設 A 為攻擊者，針對協定中 A 有可能攻擊成功的步驟做出安全性的分析，說明本研究提出的協定是滿足了保障 B 的安全性與隱私性的目的。

1. B → A：$E_{PK_B}\left(\frac{ta}{b}\right) = (C_1, C_2) = \left[\left(g^{r_1}, \frac{ta}{b} \times {y_b}^{r_1}\right)\right]_p$

   這是協定中的第二步，B 在接到 A 所傳來的資訊後，計算$\left[\frac{[t\times a]_p}{b}\right]_p = \left[\frac{ta}{b}\right]_p$，並且用自己的公鑰$PK_B$以 ElGamal 加密系統加密運算後的結果，得到$\left[E_{PK_B}\left(\frac{ta}{b}\right)\right]_p = (C_1, C_2) = \left[\left(g^{r_1}, \frac{ta}{b} \times {y_b}^{r_1}\right)\right]_p$。簡單的來說，由於是用 B 的公鑰加密，所以 A 無法解密，確保了 B 的私密資訊不會洩漏讓 A 知道。同樣的，沒有 B 的公鑰的任何惡意攻擊者，也無法解密得到任何有關a跟b的資訊。接著，更進一步的設計一個情境，證明若 A 可以藉由這個步驟得到任何關於 B 的資訊，則可以利用 A 破解 ElGamal 的語意安全。假設攻擊 ElGamal 演算法的攻擊者為AT_1，攻擊本研究提案的攻擊者為AT_2，以及一個挑戰者(challenger)。證明分成下面四階段。

   設置(Setup)階段：

   挑戰者依照 ElGamal 的金鑰生成演算法，生成加解密的公私鑰。並將公鑰PK給AT_1，AT_1再將公鑰PK轉傳給AT_2。

   詢問(Query)階段：

   (1) AT_2自選明文m_i，經過 ElGamal 加密演算法加密後得到密文$(\mathrm{C}_{1_i}, \mathrm{C}_{2_i})$，將$(\mathrm{C}_{1_i}, \mathrm{C}_{2_i})$傳給$AT_1$要求其對應的明文，$AT_1$轉傳給挑戰者也要求其對應的明文。

   (2) 挑戰者接到密文後解密，得到明文m_i回傳給AT_1，AT_1得知明文m_i後再回傳給AT_2。

挑戰(Challenge)階段：

(1) AT_2自選兩個明文m_0, m_1傳送給AT_1，AT_1轉傳給挑戰者，挑戰者接到後分別對m_0, m_1作加密得到$E_{pk}(m_0), E_{pk}(m_1)$。

(2) 挑戰者將$E_{pk}(m_0), E_{pk}(m_1)$其中之一傳給AT_1，AT_1轉傳給AT_2。

猜測(Guess)階段：

這是最後的階段，AT_2猜測$m_i', i \in \{0,1\}$並將猜測值傳送給AT_1，AT_1轉傳給挑戰者。若$m_i' = m_i$表示AT_2挑戰成功，成功的破解了本研究提出的協定。同時，也表示AT_1可藉由本協定成功破解 ElGamal 的語意安全。

由上述的結果可以知道，由於 ElGamal 是具備語意安全的加解密演算法，AT_1是不可能破解 ElGamal 的語意安全的。因此根據逆否命題與原命題等價得知，若AT_1無法藉由本協定破解 ElGamal 的語意安全，則AT_2無法破解本協定的安全性。

2. A → B：$\left(E_{PK_B}(1_1), \dots, E_{PK_B}(1_i), E_{PK_B}(k_1), \dots, E_{PK_B}(k_i), \left[\left(E_{PK_B}\left(\frac{a}{b}\right) \right)^{l_1} \right]_p, (C_1'', C_2'') \right)$

這個步驟比較特別，攻擊方式為 A 傳送造假的訊息來欺騙 B，導致 B 的判斷錯誤。從 3.2.1 的協定的步驟可以看出，B 將數列解密後可以利用 1 與亂數的數目判斷是否相等，同時也可以從(C_1'', C_2'')做相等性判斷的依據。同樣都是判斷相等性，為什麼需要判斷兩次？主要原因是因為如果只利用 1 與亂數的數目判斷是否相等，A 可以將$\left[\left(E_{PK_B}\left(\frac{a}{b}\right) \right)^{l_1} \right]_p$直接替換成$E_{PK_B}(1_{i+1})$，如此一來，無論雙方所握有的私密資訊是否相等，B 都會做出雙方相等的判斷。這是相當具威脅性的一個漏洞。舉例來說：若 A 為用戶，B 為伺服器端，A 永遠都能夠讓 B 做出相等的判斷結果，表示 A 這個使用者只要輸入任意的密碼，都可以成功的登入 B 這個伺服器。於是本研究針對這個部分加入了(C_1'', C_2'')作為補強，將 B 判斷是否相等的主要依據改為(C_1'', C_2'')的解密結果，

而把 1 與亂數的數目判斷作為額外的確認，將它的存在目的著重於讓 A 作驗證的依據。

而 B 對(C_1'', C_2'')的解密方式如之前 3.2.1 中的步驟 4 所提到的，解密時先計算$[(g^{r_1\times l_2})^{x_b}]_p = [{y_b}^{r_1\times l_2}]_p$，此時$g^{r_1\times l_2}$便同等於一把由 A 跟 B 共構的一把迪菲－赫爾曼密鑰(Diffie–Hellman key)。接著計算$\left[\left(\left(\frac{a}{b}\right)^{l_2}\times {y_b}^{r_1\times l_2}\right)\times({y_b}^{r_1\times l_2})^{-1}\right]_p = \left[\left(\frac{a}{b}\right)^{l_2}\right]_p$，由於是使用 B 的私鑰$x_b$與 B 之前自選的亂數$r_1$，所以如果 A 要試圖將$\left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p$竄改成$E_{PK_B}(1_{i+1})$，就必須知道當初 B 所選擇的亂數$r_1$為何，否則竄改後的密文就不會是個合法的密文，讓 B 可以藉由驗證發現 A 的不法行為，進而結束協定。

3. B → A：$\begin{cases}(1_1,\dots,1_i,k_1',\dots,k_i',1), a=b\\ \left(1_1,\dots,1_i,k_1',\dots,k_i',\left[\left(\frac{a}{b}\right)^{l_1}\right]_p\right), a\neq b\end{cases}$

B 在回傳數列的時候，會將原本數列中的$k_1,\dots,k_i$替換成$k_1',\dots,k_i'$。原因是因為數列中的元素順序是 A 所決定的，所以若 B 沒有將$k_1,\dots,k_i$替換成$k_1',\dots,k_i'$，當$a\neq b$時，A 可以從 B 回傳的數列中找到$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$，找到之後由於$l_1$是 A 所選取的亂數，A 便只需要計算$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p^{-1} = \left[\left(\frac{a}{b}\right)\right]_p$，就可以輕易得到$b$的值。所以 B 將$k_1,\dots,k_i$替換成$k_1',\dots,k_i'$是為了讓 A 無法在$a\neq b$時從數列中的特定元素得知 B 的私密資訊$b$。

4.2.2 假設 B 為攻擊者的情況

這個小節假設 B 為攻擊者，針對協定中 B 有可能攻擊成功的步驟做出安全性的分析，說明本研究提出的協定是滿足了保障 A 的安全性與隱私性的目的。

1. A → B：$t\times a\ mod\ p = [t\times a]_p$

協定的一開始 A 將自己的私密資訊a藉由乘上亂數t以及p的模運算，很明顯的可以知道對於所有的$a\in Z_p^*$都存在一個相對應的t滿足上式，因此 B 無法反推a，確保了 A 的私密資訊不會洩漏讓 B 知道，而且任何惡意的第三者從中擷取資訊也無從得知a為

何。

2. A → B：$\left(E_{PK_B}(1_1),\ldots,E_{PK_B}(1_i),E_{PK_B}(k_1),\ldots,E_{PK_B}(k_i),\left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p,(C_1'',C_2'')\right)$

Alice 收到 (C_1,C_2) 後，先將$[t^{-1}]_p$以 Bob 的公鑰加密，根據 ElGamal 加密系統所具備的乘法同態性質做以下三個步驟的運算：

(1) $E_{PK_B}(t^{-1})\times E_{PK_B}\left(\frac{ta}{b}\right)=E_{PK_B}\left(\frac{a}{b}\right)$，這個部分為 ElGama 的乘法同態運算，並不會洩漏任何雙方的私密資訊。而且都是以 ElGamal 加密系統所加密的密文在運算，因此對於任何沒有辦法破解 ElGamal 加密系統的惡意攻擊者來說，從密文得知任何有關a跟b的資訊是不可能辦到的。

(2) $\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1} mod\ p=\left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p$，在這一步之所以要$l_1$次方，是為了讓 B 在收到 A 回傳的訊息並解密後，得到的是$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$，基於離散對數問題 B 是無法得知$l$為何值，在不知道$l_1$為何值的情況下，B 更不可能知道$\left(\frac{a}{b}\right)$的值為何。換句話說，B 無法藉由 A 回傳的訊息計算得知 A 的私密資訊a為何，確保 A 的私密資訊a不會洩漏讓 B 知道。這一步a跟b一樣還是受到 ElGamal 加密系統的保護，任何其它惡意的攻擊者一樣無法得知任何有關a跟b的資訊。

(3) A 對i個 1 與任選的i個不等於 1 的變數k_i並以 B 的公鑰加密，將$\left(E_{PK_B}(1_1),\ldots,E_{PK_B}(1_i),E_{PK_B}(k_1),\ldots,E_{PK_B}(k_i),\left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p\right)$回傳給 B。由於 ElGamal 加密演算法具有在章節 2.8 所提到的語意安全，因此在加密i個 1 以及i個不等於 1 的變數k_i後所產生的是共$2i$個不規則的亂數，加上$\left[\left(E_{PK_B}\left(\frac{a}{b}\right)\right)^{l_1}\right]_p$就共有$2i+1$個亂數。B 在對收到的密文數列解密後，若$a=b$則會得到$i+1$個 1 與$i$個

變數，但卻無法從中判斷哪一個是由$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$運算得到的 1，於是除了得知雙方私密資訊相等的結果外，無法試圖從中得到更多的資訊。若$a \neq b$則會得到$i+1$個變數與i個 1，同樣的無法判斷哪個變數是代表$\left[\left(\frac{a}{b}\right)^{l_1}\right]_p$的運算結果。而依據$a$跟$b$是否相等，分別會回傳兩種不同的數列。無論是哪一種情況，B 都不能去更動數列內元素的順序，否則 A 將會在驗證步驟時發現 B 更動過數列，造成 A 對這次協定的結果產生懷疑。

3. A 驗證結果

從章節 3.2.2 中，已經提到過當 A 選擇的數值為 n 個 1 跟 n 個變數時，B 成功欺騙 Alice 的機率就是$\frac{1}{n}$。如此一來，A 可以利用 1 與變數 k 的數量來降低 B 欺騙成功的機率。換句話說，愈多的 1 與變數，B 欺騙成功的機會就愈小。

4.3 模糊傳輸安全性分析

在這個小節中，將對本論文所使用的 N 選一模糊傳輸協定作出安全性的分析，在分析之前先回顧一下章節 2.4.1 中所提到的模糊傳輸所需具備的三個安全需求：

(1) 傳輸的正確性：只要傳送者跟接收者按照協定步驟進行，接收者就可以得到精確地得到他所索取的訊息。

(2) 接收者的隱私：對接收者而言，他可以選擇他所要索取的訊息，無論是二選一、n 選一還是 n 選 t，傳送者都沒有辦法得知接收者所索取的訊息項目為何，以保障接收者的隱私。

(3) 傳送者的隱私：在協定執行完成後，接收者只能獲得他所索取的訊息，對於其他沒有索取的訊息是一無所知，縱使接收者有再強大的計算能力也沒辦法得知，以保障傳送者的隱私。

由於傳輸的正確性已經在章節 2.4.2 中做出了說明，因此在這個章節便將焦點著重

在協定中接收者跟傳送者的隱私上，分析並說明本文所採用的模糊傳輸協定是一個能夠確保接收者跟傳送者隱私的安全協定。

首先，先就接收者的隱私是否被安全的保護作分析。搭配章節 2.4.2.2 所提到的 N 選一模糊傳輸協定可知，接收者在每次的協定中，只會做出一次的傳送，傳送值是 $\begin{cases} PK_0 = g^k, \sigma = 0 \\ PK_0 = \frac{C_\sigma}{PK_\sigma}, \sigma \neq 0 \end{cases}$，而傳送者在接到接收者所送來的$PK_0$後，並無法從$PK_0$判斷接收者在本次協定中所使用的$PK_\sigma$為何。換句話說，傳送者得知接收者所索取的訊息項目為何的機率同等於任意猜測，也就是$\frac{1}{\mathrm{N}}$，所以接收者的隱私的保護在此協定中是有達到的。

接著，傳輸者的隱私保護安全分析就比較複雜，需要利用 Computational Diffie-Hellman (CDH)的難問題作為輔助，因此在開始分析傳輸者的隱私保護前，先回顧之前在章節 2.6 所定義的 CDH 難問題。

在回顧完 CDH 難問題後，接著便可藉由 CDH 難問題的幫助，對傳送者的隱私保護做出安全性的分析。如同之前所提到的，傳送者的隱私保護需要確保每一次的 N 選一協定中，縱使接收者有再強大的計算能力，也無法從傳送者那裏得到兩個以上的訊息。於是先假設接收者擁有一個以上的公鑰，這裡假設是PK_1跟PK_2，此時連結到 CDH 難問題，給定$A = g^a, B = g^b$，並選定兩個亂數r_1, r_2，模擬$C_1 = A^{r_1}, C_2 = A^{r_2}, g^r = g^b$。若接收者能夠同時解出$M_1, M_2$兩個明文，傳送者只需從協定的過程中找出對應的$C_1$以及$C_2$，相除得到$\left(\frac{C_1}{C_2}\right)^r = \left(\frac{A^{r_1}}{A^{r_2}}\right)^r = A^{r(r_1-r_2)} = g^{ar(r_1-r_2)} = g^{ab(r_1-r_2)}$，此時由於$r_1, r_2$是傳送者自選，於是將$\left(g^{ab(r_1-r_2)}\right)^{(r_1-r_2)^{-1}} = g^{ab(r_1-r_2)(r_1-r_2)^{-1}} = g^{ab}$，使得傳送者在只知道$g, g^a, g^b$的條件底下，計算出了$g^{ab}$，破解了 CDH 難問題。也就是說，若接收者可以解出兩個以上的明文，則傳送者就能藉此破解 CDH 難問題。由逆否命題可知，若傳送者不能破解 CDH 難問題，則接收者不可以解出兩個以上的明文。證明了此協定中，接收者除了自己所索取的訊息外，無法獲得非他所索取的任何額外訊息或資訊，確保了傳送者的隱私。

5. 相關應用

在這個章節裡，將對本研究提出的各個協定，根據其符合的特性，做出不同的應用探討。章節 5.1 是以章節 3.2 所提到的雙方相等性驗證協定為主軸，發展出伺服器端與用戶端的登入驗證系統。章節 5.2 則是將章節 3.3 所提到的雙方相等性驗證搭配模糊傳輸之協定應用於線上購買數位化的出版品，如電子書、音樂、影片等。章節 5.3 則是將雙方相等性驗證搭配模糊傳輸之協定應用在一個投票的系統中，模擬出一個結合數位與現實的投票的情境。以下將對每個應用做出更加詳細的說明。

5.1 登入驗證系統

當某個使用者在想要某個網站上註冊一個帳戶，勢必要輸入像是帳號、密碼、真實姓名、生日、電子郵件、連絡電話…諸如此類的一些個人資訊。而這些資訊在使用者按下註冊按鍵的同時，伺服器端便將這些資料存在後端的資料庫裡。每當使用者要以自己的帳號登入該網站時，伺服器就必須比對資料庫中的資料，檢查該帳戶是否存在，若存在則驗證該使用所鍵入的密碼是否正確。這對生活在這個數位化世界的每個人來說是如此的直觀而且幾乎每天都在做的事情。但是，使用者存在資料庫裡的這些資料，有被保護嗎？如果駭客入侵了資料庫，那這些個資不就外洩了嗎？因此，為了保護使用者的個資，許多的儲存在資料庫中的資料是以加密過後的密文型態保存的。既然經過加密，當駭客入侵資料庫，所看到的都是如同亂數般的密文，當然也就無法直接得到關於註冊用戶的任何資料。

接著，駭客如果想知道某個使用者的帳號密碼，可以採用監聽的方式，監聽使用者與伺服器的通訊，並擷取傳輸的內容。因此，使用者與伺服器端都會希望連通訊時都必須要加密，就算使用者輸入明文，但在送出時會經過一道加密的手續，使得在傳輸過程中也是以密文的形態存在。如此一來，就算駭客成功地從通訊的過程中擷取下來訊息，也無法知道使用者與伺服器端所做的溝通為何。

再者，人們依賴網路的程度愈來愈高，網路詐騙與攻擊也跟著出現，其中的一種稱作網路釣魚(Phishing)。這種攻擊主要是使用垃圾郵件、惡意網站、電子郵件及即時通訊來誘騙人們洩漏機密資訊，例如信用卡或是銀行的帳戶。本研究將重點放在使用惡意網站做出的釣魚攻擊。惡意網站多半是製作精良的偽網站，製作的頁面與真正的網站幾乎完全一樣，讓使用者誤以為到了真的網站，藉此盜取使用者在假網站所鍵入的帳號密碼或是其他資訊。因此，使用者也會希望能夠驗證自己是不是登入了真正的網站，更進一步的還希望就算登入的是假的網站，也不會有帳號密碼被盜取的情況。

綜合以上所述，為了抵抗眾多的攻擊手法，無論是在資料儲存還是溝通傳輸，都希望資訊是以加密過的密文型態存在。但如果都是密文，伺服器雙方該怎麼去驗證是否為合法的登入？如果要防治釣魚網站，使用者又該如何驗證伺服器端呢？利用本研究所提出的雙方相等性驗證，可以輕鬆地解決上述的問題，達到安全有效的登入及驗證。以下將舉圖例說明。

國立政治大學 National Chengchi University

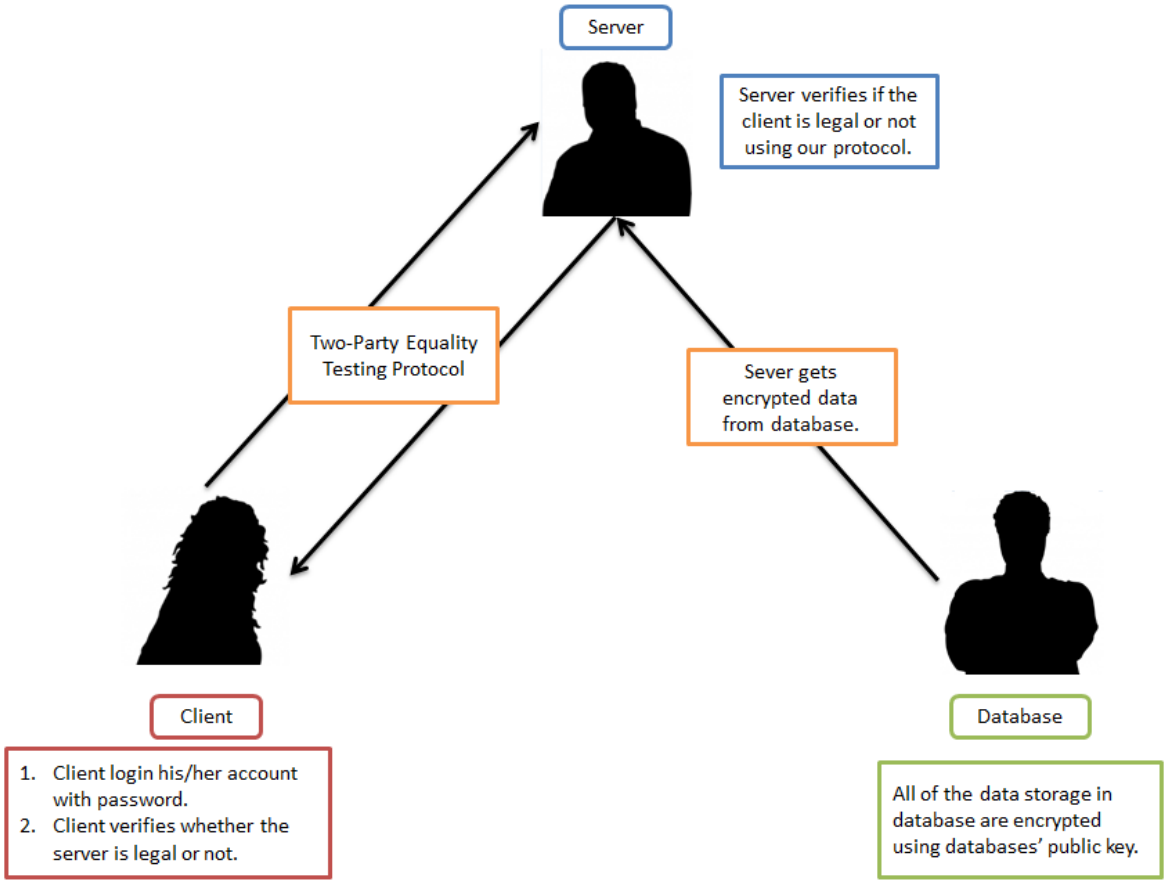

圖 14 雙方相等性驗證應用於登入架構

以本研究所提出的雙方相等性驗證架構出來的登入系統如圖 14 所示，接著將說明這樣架構的登入系統滿足哪些安全性：

1. 使用者登入時的密碼會因為協定的第一步達到密碼隱蔽的效果。
2. 資料庫中所存放的資訊都是以資料庫的公鑰加密過的密文型態存在，當資料庫被入侵時保障了用戶資料的安全。
3. 伺服器從資料庫中讀取資訊並利用協定中的同態運算比較使用者所輸入的帳號密碼是否正確。由於都是密文在做運算，伺服器端也無從得知使用者的任何資訊。
4. 在伺服器端驗證通過後，使用者也可以驗證伺服器的真偽，若是一個釣魚網站，使用者將立刻停止與該網站的互動。而且如第 3 點所提到，協定中的一切運算都是採用密文的同態運算，因此釣魚網站並無法因此得知使用者的真實帳號密碼或是任何

帳戶資訊，若使用者不放心仍可在驗證出伺服器是偽造之後第一時間做更改密碼的動作，減少帳號密碼被濫用的機會與損失。

綜合以上所述，使用本研究提出的雙方相等性驗證協定架構出的登入系統是相當安全而且保護隱私的，更可以有效的防治釣魚網站的攻擊。

5.2 線上購買數位化出版品

隨著出版品數位化的趨勢，許多的書籍、影片、音樂都可以利用網路購買並且下載，改變了傳統到實體店面購物的消費型態。而一旦涉及網路交易行為，第一個要考量的就是交易機制的安全性。消費者們儘管享受著在網路上消費的便利性，卻也擔心不健全的交易機制會使自身的權益受損或是造成金錢上的損失，例如信用卡資訊被盜用。而另一方面，網路上的賣家也會思考該如何設計收費機制，讓使用者付費，確保跟正確的消費者收取金額，並且正確無誤地提供該消費者購買的數位商品。

除了上述所提到的問題點之外，人們對於隱私保護的要求與日漸增。消費者希望在購買數位商品的同時，可以保障自身隱私，不想讓賣家或是任何人知道自己在網路上搜尋或是購買了什麼樣的商品。而賣家當然不願做虧本的生意，希望交易機制能確保消費者不能獲取他未付費的數位商品。這些買賣雙方的需求又該如何去滿足？

為了滿足以上所提到的需求與問題點，於是利用本研究提出的雙方相等性驗證搭配模糊傳輸之協定為主體，建構一個交易的機制，同時滿足交易的正確性以及隱私性，以下將以流程圖說明。

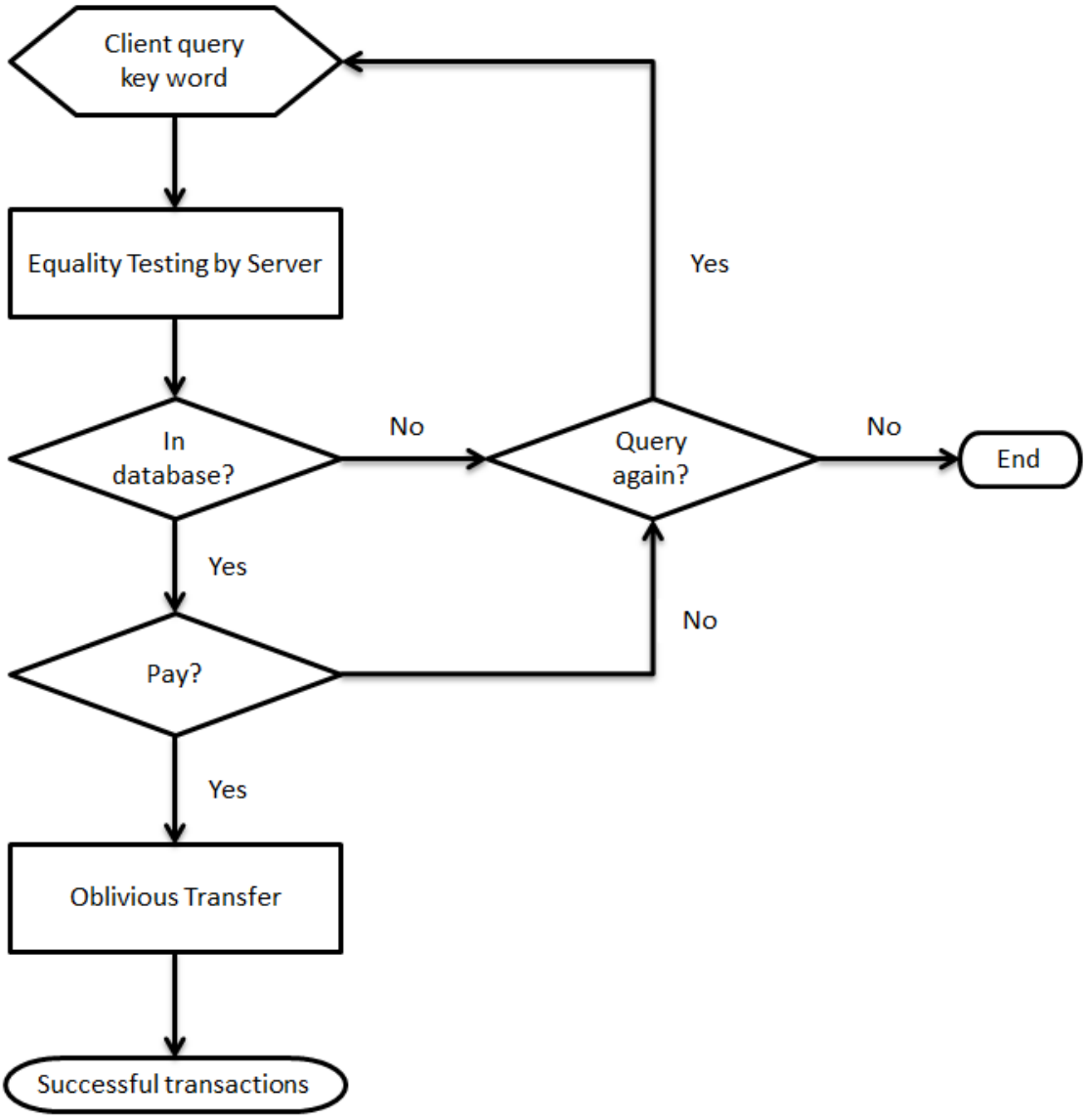

圖 15 雙方相等性驗證搭配模糊傳輸之協定應用於線上購買數位化出版品

以本研究所提出的雙方相等性驗證搭配模糊傳輸之協定所架構出來的交易流程如圖 15 所示，接著將說明這樣系統如何滿足上述所提到的要求：

1. 賣家比對消費者所搜尋的關鍵字與資料庫中的資料時，所使用的是本研究所提出的雙方相等性驗證協定，因此是兩個密文在作比較是否相等，所以如果賣家資料庫中沒有消費者所搜尋的商品，賣家是不會知道消費者搜尋了什麼，保障了消費者的隱私。而在賣家作出回應後，根據本研究所提出的驗證方案，消費者也可以判斷賣家

是否說謊，決定是否繼續交易。這樣的比對跟驗證對於消費者來說是利多的，其原因在於消費者可以在知道賣家是否有自己想購買的商品後，才在該網站註冊跟登錄付費購買的動作，對消費者來說可以避免因為註冊過多不必要的帳號所造成的個資外洩風險。

2. 在知道賣家有消費者所需的商品後，藉由模糊傳輸可以讓消費者拿到正確的商品，也確保了消費者除了自己付費購買的商品外,其他資料庫的資料是完全無法取得的，落實了使用者付費的觀念，保障了賣家的權益。同時因為模糊傳輸的關係，賣家也無從得知消費者購買了哪一件商品，完整保護了消費者的隱私。

綜合以上所述，使用本研究提出的雙方相等性驗證搭配模糊傳輸之協定建構出的交易系統是相當安全而且保護買賣雙方隱私的。

5.3 結合數位與現實的投票情境-核發選票

每當投票日到來，盡責的公民們就會拿著投票通知單、身分證跟印章，起個大早去排隊領選票，選下心目中的理想候選人，並投下神聖的一票。許多投過票的選民都知道，這看似簡單而且愜意的幾個動作,真實的畫面是大排長龍而且各種類型的問題百出,例如：證件印章沒帶齊，對照身分花太久時間、人員調度問題…等等。如果再碰上個大熱天，那簡直會讓每個人都脾氣火氣都上來。

沒錯，電子投票與網路投票是可以解決這些的問題，但為什麼目前為止電子投票以及網路投票並未普及呢？電子投票已經在美國、愛沙尼亞行等國家行之有年，未普及的原因不外乎數位落差，加上人們已經熟悉了傳統紙本的投票模式，對於電子投票的接受度不高。在台灣雖然有在推行電子投票的系統，卻目前仍止於理論研究的階段。而網路投票相對於電子投票來說又更加的難以施行，原因在於網路投票是私下投票，例如在家中上網投票。雖然聽起來相當方便，但也因為是私下透過網路投票，可能會產生有旁人在側脅迫的問題。除了這個問題之外，網路投票目前來說多半以類似自然人憑證的系統

作認證身分的動作，若是密碼遭竊，則無法阻止冒名投票的情況。

因為上述的種種問題，本研究構想了一個數位認證與網路投票或是傳統投票的結合方案。利用本研究提出的雙方相等性驗證協定搭配模糊傳輸協定建構一個核發選票的系統，藉由雙方相等性驗證協定認證選民身分以及模糊傳輸協定發出選票，經過核發的選票會具有有效的簽章證明為有效的選票。而且由於是藉由模糊傳輸的協定發出選票，因此確保了伺服器端不知道選民所選擇的候選人，也保證每個選民只會藉由協定獲得所選候選人的有效票，不會有一張選票可投給兩人的錯誤或是廢票。接著選民再拿著有效票到特定的投票所或是登入投票系統做投票的動作。流程如下圖所示：

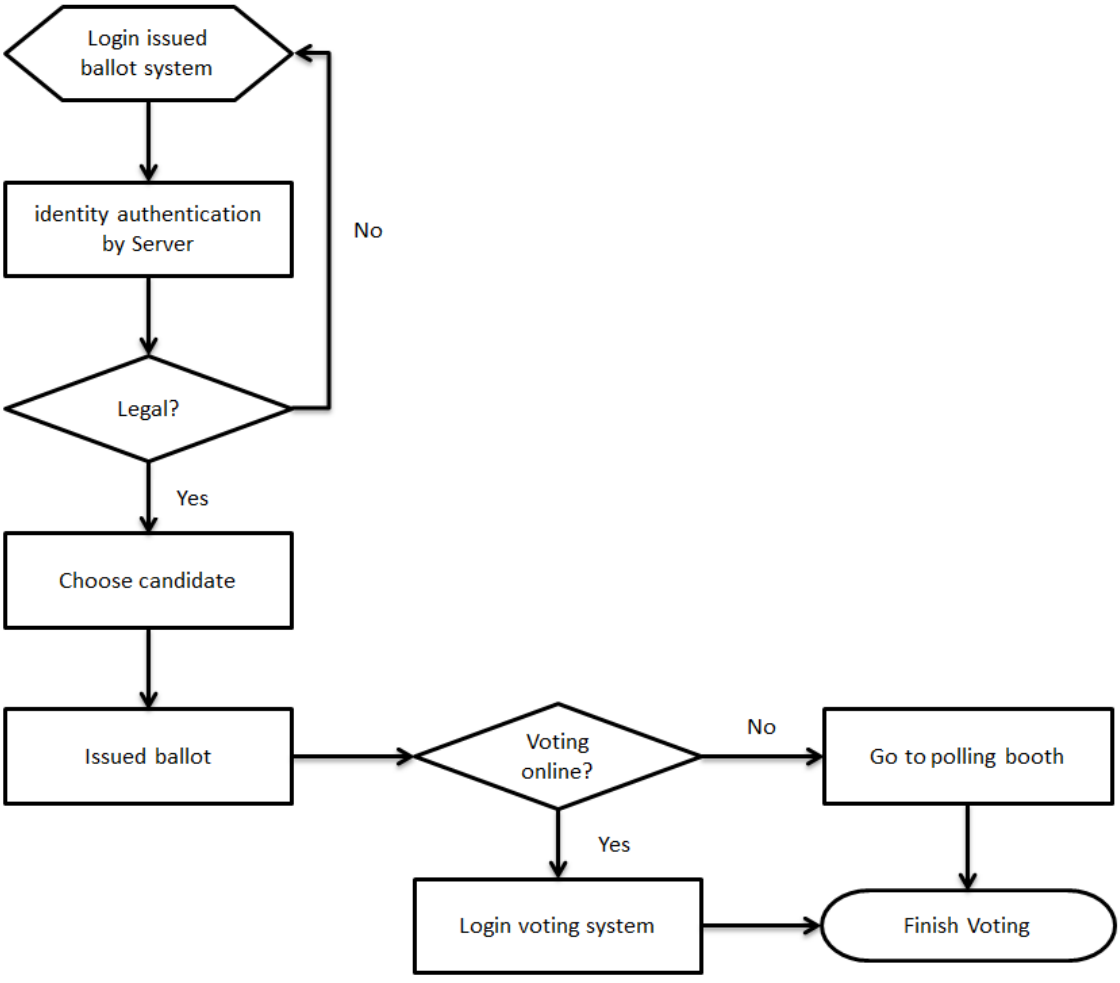

圖 16 雙方相等性驗證搭配模糊傳輸之協定應用於核發選票

6. 實作與效能分析

這個章節將介紹本研究的協定實作，以及對本研究提出的協定效能做出分析。由於之前沒有人做過這樣類型的協定與實驗，於是在這裡並沒有跟其他作者做比較。在實作的部分是利用 JAVA 撰寫程式碼，執行環境是利用個人電腦(2.50 GHz Pentium 4 Dual-Core, 4GB 記憶體)。由於單純的要評估所設計的協定演算法效能，於是排除從資料庫匯入資料的時間，將資料先存取在陣列當中。

模擬狀況為讓使用者輸入全國 163 所大專院校的英文校名縮寫，經過雙方相等性驗證的協定比較伺服器端的陣列中是否有使用者輸入的校名縮寫，若存在則回傳相對應的中文校名，若沒有該值則協定結束。模擬中也讓伺服器企圖欺騙使用者，測試協定是否能按照流程驗證出伺服器的不良行為。以下為流程圖以及執行時的截圖，但由於 163 所大專院校的執行截圖實在太過於冗長，因此先將學校數減為 10 所做正確性測試：

國立政治大學 National Chengchi University

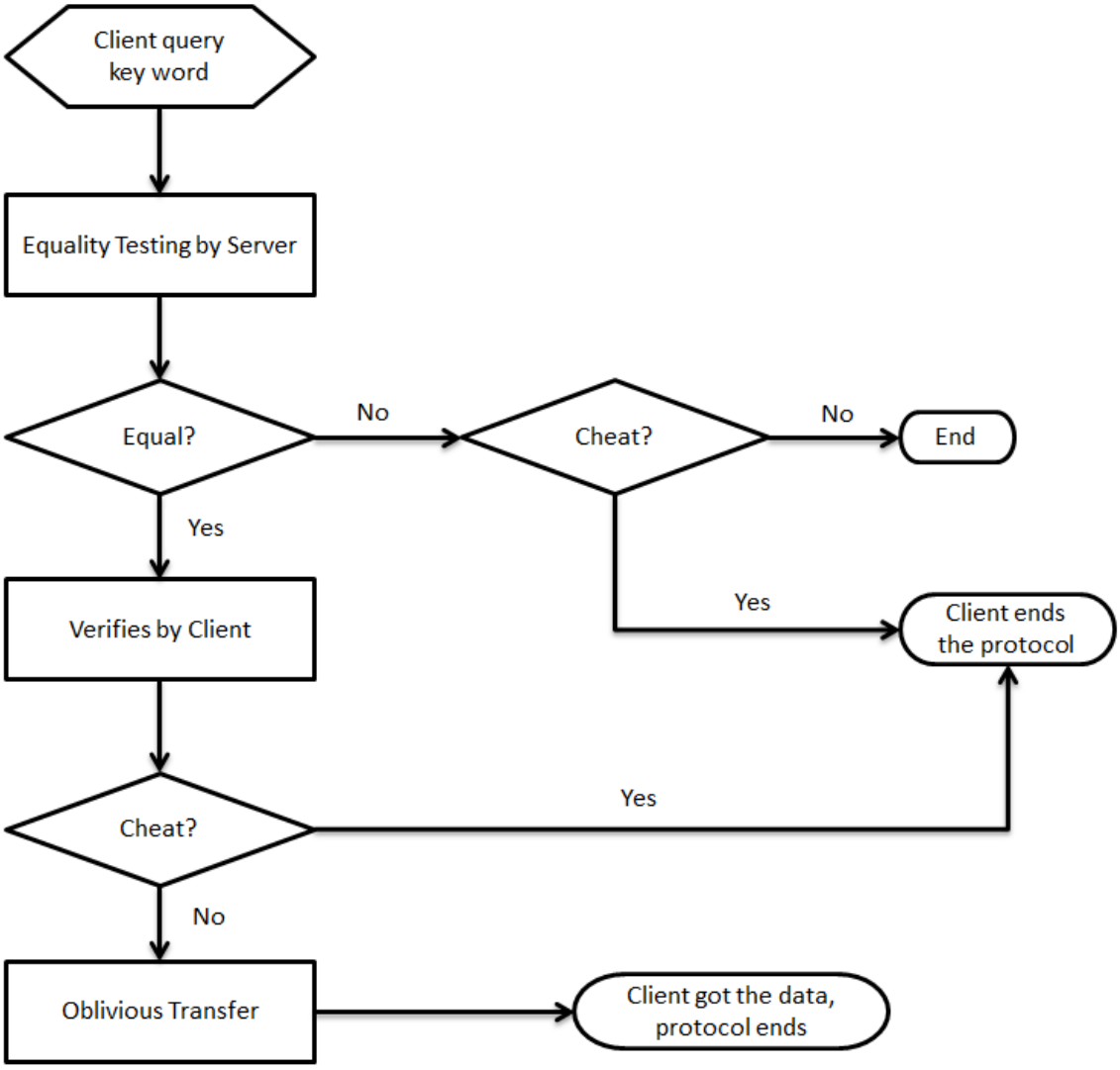

圖 17 協定模擬流程圖

```
String[] b = { "ntu", "fju", "nctu", "nccu", "mcu", "nthu", "scu", "ncku",
        "tku", "ncu" };
String[] d = { "國立台灣大學", "私立輔仁大學", "國立交通大學", "國立政治大學", "私立銘傳大學", "國立清華大學",
        "私立東吳大學", "國立成功大學", "私立淡江大學", "國立中央大學" };
```

圖 18 測試時所選取的 10 所大專院校

```
new_array[c.length] = BigInteger.valueOf(count_k - count_one);
// new_array[c.length] = BigInteger.valueOf(d.length-2);//test
// liar(always the same)
// new_array[c.length] = BigInteger.valueOf(d.length);//test liar(always
// not the same)
```

圖 19 用來控制伺服器是否說謊的程式碼〈一〉-伺服器為誠實

```
Enter something you want to query:
nccu
Enter 10 index numbers(divided by comma):
0,4,6,8,9,10,12,13,14,16
(Server)The same.
(Client)The same.
???昊????倬?繶澵?????????????????相??黯???酪屐熔珊既???????姒????鰋咔?慍詺晷?
     ??慵??   蕡?澓????盒???欖?????聚?經   ??樣??槏??????????伂??溧?呯?????????卍
????cc?????   ???????   義?????????燒??頍???鼬??晲??豬?   堉?嚃??      ?????????
國立政治大學
巃??祚?????腩???   ??????甸?????）???????啐??掄?嚨嘟???冇鰱????   ??A??????蝵
?   ??   ???   ??糶?緧?畐?   ????????   恟妯值?揳???鶖???   瘡??????????一?昴昷   抶????
??痘?簝慛??   ????禹??   屾黃??碰仉?Q?????????????????慡   磢????????????      惿   ??
??????悶??牰   ????   ????攸菩????酬?叵???氪圍??   ?諰???悚?   鄒???????   庵??        ?
?????著?換   ???犡???   ??堐?   蒐??????淨????????撽??婿??   ??   ???????圩琦????
  ?   ??竷降猰??嚼??注??   x?病?櫆   ??窓????   醧?   ???   ????      瘧   ??????檻?????椁矵?
```

圖 20 伺服器誠實且比對找到 nccu 的執行結果

```
Enter something you want to query:
gwjk
Enter 10 index numbers(divided by comma):
0,4,6,8,9,10,12,13,14,16
(Server)Not equal.
(Client)Not the same. The protocol ends
```

圖 21 伺服器誠實且比對無法找到 gwjk 的執行結果

從圖 18 與圖 20 可以清楚的看到，若資料庫中有關鍵字〝nccu〞，且伺服器端沒有做出欺騙動作，則經過相等性驗證後，伺服器告知使用者資料庫中有該關鍵字的相關資料，而使用者也可以經過驗證得知伺服器沒說謊且資料庫中有所需的資料。接著經過模糊傳輸，使用者得到 10 個值中，只有〝政治大學〞能夠成功被解密得到明文，其他非使用者所搜尋的資料是一連串的亂碼，達到了傳輸的正確性以及傳送者的隱私，而接收者的隱私雖然無法從執行結果看出，但可藉由回顧章節 4.2 的證明搭配附錄的程式碼獲得說明。而圖 18 與圖 21 則顯示當輸入的關鍵字並不存在於資料庫的話，則伺服器告知使用者沒有該筆資料，使用者也驗證伺服器沒說謊，協定結束。接著將修改程式碼，讓伺服器說謊，看使用者是否能發現並依照流程中斷協定。

```
//new_array[c.length] = BigInteger.valueOf(count_k - count_one);//honest
 new_array[c.length] = BigInteger.valueOf(d.length-2);//test liar(always the same)
// new_array[c.length] = BigInteger.valueOf(d.length);//test liar(always
// not the same)
```

圖 22 用來控制伺服器是否說謊的程式碼〈二〉-伺服器說謊(永遠說相等)

```
Enter something you want to query:
gwjk
Enter 10 index numbers(divided by comma):
0,4,6,8,9,10,12,13,14,16
(Server)The same.
(Client_type1)B is a liar. The proptcol ends
```

圖 23 伺服器說謊被使用者驗證發現〈一〉

```
// new_array[c.length] = BigInteger.valueOf(count_k -count_one);//honest
// new_array[c.length] = BigInteger.valueOf(d.length - 2);// test
// liar(always the same)
new_array[c.length] = BigInteger.valueOf(d.length);// test liar(always not the same)
```

圖 24 用來控制伺服器是否說謊的程式碼〈三〉-伺服器說謊(永遠說不相等)

```
Enter something you want to query:
nccu
Enter 10 index numbers(divided by comma):
0,4,6,8,9,10,12,13,14,16
(Server)Not equal.
(Client_type2)B is a liar. The protocol ends
```

圖 25 伺服器說謊被使用者驗證發現〈二〉

從圖 22 與圖 23 可以看出，當使用者輸入了資料庫中沒有的關鍵字〝gwjk〞，伺服器卻欺騙使用者資料庫有該值時，使用者可透過協定驗證發現伺服器說謊，並將協定停止。而從圖 24 與圖 25 可以看出，當使用者輸入了確實存在於資料庫中的關鍵字〝nccu〞，伺服器卻欺騙使用者資料庫沒有該值時，使用者同樣可以透過協定驗證發現伺服器說謊，並將協定停止。透過程式執行以及以上的幾個情境的測試，說明了本研究提出的雙方相等性驗證搭配模糊傳輸之協定的正確性。接著，來探討協定從執行到結束所需花費的時間。以下的表為參數長度的設定。

表格 4 參數長度設定

參數	長度(bit)	適用方案或步驟
p	1024	ElGamal、Oblivious Transfer
q	1024	ElGamal、Oblivious Transfer
g	1024	ElGamal、Oblivious Transfer
r	1024	ElGama
t	1024	雙方相等性驗證步驟 1
l	1024	雙方相等性驗證步驟 3

當資料量擴增到 163 所大專院校時，若關鍵字以明文存在於資料庫中，且伺服器誠實，使用者從輸入關鍵字到取得資料，在做 100 次的模擬後，所需耗費的時間平均為 35933 毫秒，也就是 35.933 秒。另外，若關鍵字本來就以密文存放在資料庫中，則省略協定中加密資料庫資料的運算時間，經過 100 次的模擬後，所得到的平均時間為 28907 毫秒，也就是 28.907 秒。由此可知，若資料為 163 筆的明文，協定需花費相當於五分之一的執行時間做加密資料庫中的資料。因此，資料庫內資料最好可以用密文型態儲存，不但節省了協定的執行時間，更重要的是當駭客入侵資料庫，所看到的都是如同亂數般的密文，充分的保障了資料庫中的資料安全。

7. 結論

本研究提出了一個在雙方架構下比較雙方私密資訊是否相等的協定，同時允許協定中的雙方驗證結果的正確性。有別於之前學者所提出的方案，協定所用的加密演算法為常見的 ElGamal 演算法，對加密演算法的選取不需做額外條件的限制，唯一的限制就是所選取的加密演算法須符合語意安全以及乘法同態。換句話說，只要符合語意安全以及乘法同態的加密演算法，都可以利用此協定達到雙方相等性的比較。因為如此，此協定較容易實作，應用的可能性也相對的提高。除此之外，利用本研究提出的協定搭配有效率的模糊傳輸，可以在保護雙方的隱私下，達到正確的資料傳輸。同時也將雙方相等性驗證搭配模糊傳輸之協定應用於數位商品的購買，保障了買賣雙方所需要的隱私，也保障了買賣的正確性以及公平性。

在實作方面，本研究實作了一個簡單的模擬，讓使用者輸入關鍵字搜尋，並回傳相對應的資料。除了理論證明外，更以模擬證實了本研究提出協定的正確性。雖然 163 筆資料量平均仍須 15.524 秒的執行時間，但相信能夠藉由比對演算法的改善或是程式的最佳化來減少執行時間，達到優化的效果。

未來希望能夠進一步的研究如何將此協定延伸，發展出適用於如 RSA 等具備同態性質但非語意安全的加密演算法的方案。此外，也希望將程式改為利用網路溝通互動的模式，計算通訊所需耗費的時間，使其更為貼近生活中所應用的實際情況，以便針對使用者的需求加強及改良。若能將協定的運算時間在降低，相信會有更多的應用能夠發展，使更多層面的資料安全及隱私能夠被充分保護。

參考文獻

[1] J. Benaloh. “Dense probabilistic encryption”, Proceedings of the Workshop on Selected Areas of Cryptography, pp. 120–128, 1994.

[2] D. Boneh, EJ. Goh, and K. Nissim, “Evaluating 2-DNF formulas on ciphertexts”, Proceedings of Thepry of Cryptography (TCC), pp. 325–341, 2005.

[3] I. F. Blake and V. Kolesnikov, “Strong conditional oblivious transfer and computing on intervals”, Proceedings of Advances in Cryptology (Asiacrypt'04), LNCS vol.3329, pp.515-529, 2004.

[4] G. Brassard, C. Cre'peau, and J. M. Robert, “Oblivious transfer and privacy amplification”, Proceedings of Advances in Cryptology (Eurocrypt'97), LNCS vol.1233, pp. 334-346, 1997.

[5] M. Bellare, and S. Micali, “Non-interactive oblivious transfer”, Proceedings of Advances in Cryptology (Crypto'89), LNCS vol.435, pp. 547-557, 1990.

[6] S. F. Ciou, “Two-party equality test with privacy protection”, Master's Thesis, 2011. (in Chinese)

[7] S. F. Ciou, R. Tso: “A privacy preserved two-party equality testing protocol”, Proceedings of ICGEC 2011, pp. 220-223, 2011.

[8] C. K. Chu and W. G. Tzeng, “Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries”, Proceedings of the Public Key Cryptography(PKC '05), LNCS vol.3386, pp.200-212, 2005.

[9] C. K. Chu and W. G. Tzeng, “Conditional oblivious cast”, Proceedings of the Public Key Cryptography (PKC '06), LNCS vol.3958, pp. 443-457, 2006.

[10] G. D. Crescenzo, R. Ostrovsky, and S. Rajagopalan, “Conditional oblivious transfer and time-released encryption”, Proceedings of Advances in Cryptology (Eurocrypt'99), LNCS

vol.1592, pp. 74-89, 1999.

[11] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Trans. Inform. Theory, vol. 31, pp. 469-472, 1985.

[12] C. Gentry, “Fully homomorphic encryption using ideal lattices”, Proceedings of STOC ’09, ACM, pages 169–178, 2009.

[13] C. Gentry and Z. Ramzan, “Single-database private information retrieval with constant communication rate”, Proceedings of ICALP 2005, pp.803-815, 2005.

[14] S. Goldwasser and S. Micali, “Probabilistic encryption & how to play mental poker keeping secret all partial information”, Proceedings of Annual ACM Symposium on Theory of Computing, pp.365-377, 1982.

[15] B. Hemenway and R. Ostrovsky, “Lossy trapdoor functions from smooth homomorphic hash proof systems”, In Electronic Colloquium on Computational Complex-ity, Report TR09-127, 2009.

[16] M. Hirt and K. Sako, “Efficient receipt-Free voting based on homomorphic encryption”, Proceedings of (Eurocrypt'00), LNCS vol.1807, pp.539–556, 2000.

[17] K. Kurosawa and Q. Duong, “How to design efficient multiple-use 1-out-n oblivious transfer”, IEICE Trans. Fundamentals, vol.E87-A, No.1, pp. 141-146, 2004.

[18] R. Li and C.K. Wu, “Co-operative private equality test”, International Journal of Network Security, vol.1, No.3, PP.149–153, 2005.

[19] N.Y. Lee and C.C. Wang, “Verifiable oblivious transfer protocol”, IEICE Trans. Information and Systems, vol.E88-D, No.12, pp. 2890-2892, 2005.

[20] M. Naor and B. Pinkas, “Efficient Oblivious Transfer Protocols”, Proceedings of ACM-SIAM symposium on Discrete algorithms, pp.448-457, 2000.

[21] P. Paillier, “Public-key cryptosystems based on composite degree residuocity classes”,

Proceedings of Advances in Cryptology (Eurocrypt'99), LNCS vol.1592, pp. 223–238, 1999.

[22] R. Rivest, A. Shamir and L. Adleman,” A method for obtaining digital signatures and public-key cryptosystems”, Comm. ACM vol.21, pp. 120-126, 1977.

[23] M. Rabin, “How to exchange secrets by oblivious transfer”, Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981

[24] A. Yao, “Protocols for secure computations”, Proceedings of 21st Annual IEEE Symposium on Foundations of Computer Science, 1982.

國立政治大學 National Chengchi University