植基於屬性加密之外掛式郵件系統

彭煜博[1] 陳品中[2] 柯冠宇[3] 王竣輝[4] 范俊逸[5]
國立中山大學資訊工程學系[1, 2, 3, 4, 5]
kmes1234@gmail.com[1]; owen.chen[2]@gmail.com2; garyko0406@gmail.com[3]
noiping2@gmail.com[4]; cifan@mail.cse.nsysu.edu.tw[5]

摘要

電子郵件已成為現代人在商業貿易上不可或缺的工具之一。使用電子郵件非常方便，但也同時衍生出許多資訊安全問題，一旦使用者不具備完善的資安意識，則可能造成嚴重後果，像是商業機密被公開、使用者隱私洩漏等。因此資訊安全在電子郵件中顯得相對重要，除了需防止一般的駭客利用社交工程，非法獲得使用者帳密竊取個人資料外，同時也要具備訊息加密之功能，使處理信件內文的系統能獨立於郵件伺服器外，且不被服務提供商所看見。

電子郵件服務提供商作為通信雙方之間的傳訊者，可能具有一定之權限得知或儲存信件的內文，因此對通信雙方來說，必然冒著資訊洩漏的風險。為了減少此類資安問題，本團隊設計出企業內部使用之郵件系統 Nmail，並且可提供加密信件之搜尋和寄送功能，其中寄送郵件之服務提供商可以是其他公司所提供之服務，例如 Gmail 等，而企業只需提供一內部伺服器負責加密金鑰的儲存和加密信件，再經由 Nmail 系統寄出加密信件，使服務提供者因為無法對信件進行解密從而知悉信件內容，只單純提供寄送與儲存郵件的功能，進而實作出外掛式郵件加密系統。

關鍵詞：屬性加密、密文搜尋、關鍵字搜尋、中文分詞。

1. 前言

網際網路日漸發達，資料取得容易的優點，同時也是資訊安全中的一項威脅，尤其是網路上的機密資訊，常常成為駭客攻擊的首要目標。因此，學會保護自己的資料便是資訊安全中的重要課題。因應密碼學、加密演算法的興起，本團隊的研究也將著重於資料的防護和加密上。

電子郵件作為最常見的網路通訊方式，能夠迅速傳遞訊息給對方。雖然在使用上相當便利，但也不免存在一些有心人士在信件上面進行惡意行為，像是在信件中夾帶木馬或勒索病毒、試圖攔截他人信件以竊取資訊。因為電子郵件的普及與便利性，人們在傳送電子郵件時往往會忽略資安問題，讓電子郵件成為駭客的主要目標之一。因為信件內容以看似不易閱讀的封包在網路上傳播，使人們認為信件傳送的過程是安全的。然而，事實卻正好相反，當使用者寄出信件後，這些封包為完全公開的，任何人只要截取那些封包，透過轉碼器便能得到一份相同的信件內容；而其中服務提供商作為傳遞郵件必經的中間者，更是有機會檢視這些內容。因此，信件是否被偷窺成為眾多使用者對電子郵件抱持安全疑慮的主要原因之一。

承上所述，人們往往會忽視網路世界中資訊的安全性及隱私性的問題，如何確保使用者在網路上傳送或儲存的資料不會被截取、偷窺，便是此研究要探討的議題。因此，本團隊提出一種「外掛式郵件加密系統」，讓處理資料的伺服器端「看不到」使用者寄送的資訊和內容，期望能減少諸如上述的風險，保護使用者的資料。

2. 文獻回顧與探討

2.1 電子郵件運作原理

在收發電子郵件的過程中，有三個主要的 Agents，分別為與用戶端互動的 MUA、轉送郵件的 MTA 與處理郵件存儲的 MDA [1]。接下來將針對此三個 Agent 進行說明。

- **MUA (Mail User Agent)**

用來與 User 進行互動的電子郵件應用程式，主要的功能為接收郵件主機的電子郵件，以及提供 User 瀏覽與編寫郵件，通常安裝在 User 的電腦，像是 Windows 系統上的 Eudora 或是 Microsoft Office 裡的 Outlook Express 等，皆是屬於 MUA。

- **MTA (Mail Transport Agent)**

郵件伺服器端所安裝的軟體，主要功能為將從 MUA 傳送而來的郵件轉送出去，或是接收其他 MTA 所傳送過來的信件，並等待 MUA 來索取該信件。常見的 MTA 如 Sendmail， Postfix 等，或是一般也直接稱此郵件伺服器為 MTA。

- **MDA (Mail Deliver Agent)**

將 MTA 接收到的郵件保存到硬碟或指定地方，通常會進行垃圾郵件及病毒掃描。

而一般的電子郵件傳送流程，則是依靠上述三個 Agents 的分工合作來完成。寄信人(Gary)利用 MUA 撰寫電子郵件，再經過 MTA 把信件轉送到收信人(Owen)的 Mailbox 裡。圖 1 為電子郵件收發流程圖，流程如下所述：

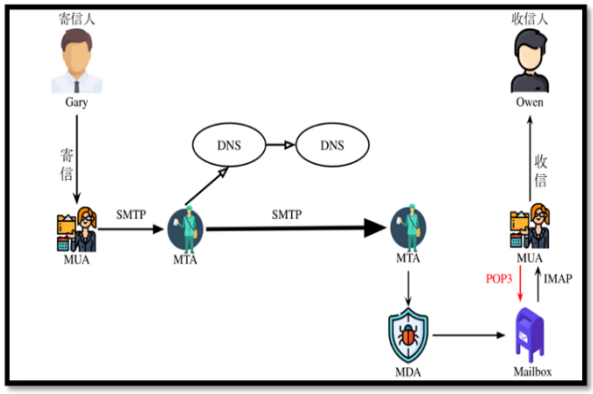

圖 1、電子郵件收發流程圖

I. 使用者向 MTA 註冊取得合法的 Email 帳號及密碼，再利用此合法的帳號及密碼到 MTA 登錄，同時取得 MTA 的認證與授權進行收發信件。

II. 寄信人在 MUA 上撰寫信件，如果收件地址等於寄信人自己的信箱時，則直接將這封信送到這台主機 Mailbox。相反的，MUA 則會將這封信透過 SMTP(Simple Mail Transfer Protocol)傳送至 MTA，透過信件的標頭 (Header)，從 DNS(Domain Name System)取得收信人的 MTA 伺服器地址並使用 SMTP 把郵件傳送過去。

III. 收信人的 MTA 收到信件後，確認為本地端(Local)信件，則轉交到 MDA，MDA 確認信件並無夾帶病毒後，MDA 再將信件放入該收信人的 Mailbox 中，待對方打開信件。

IV. 收信人使用 MUA 程式進行收信，MUA 透過 POP3(Post Office Protocol)或 IMAP（Internet Message Access Protocol）將信件取回，或線上處理。

2.2 密文加密方式

十九世紀，計算機學家認為一個適當的密碼學機制應該具有柯克霍夫原則：「即使敵人知道使用何種演算法加密，只要私鑰未洩漏，它也應是安全的」。而古典加密演算法並不遵從此原則，因此漸漸發展出了金鑰加密的概念。此金鑰加密概念最早在 1974 年由 Ralph C. Merkle [2]提出，並於兩年後由 Whitfield Diffie 與 Martin Hellman [3] [4]兩位學者，提出公開金鑰加密，造就現代密碼學的理論基礎。

現今常見的密碼學方式依照金鑰屬性分為三種，分別為雜湊(Hash)、對稱式金鑰(Symmetric)與非對稱式金鑰(Asymmetric)。其中雜湊法在加密演算上具有不可逆性，不適合用作資料傳輸的用途，因此本文只討論對稱式加密法與非對稱式加密法。

2.2.1 對稱式加密(Symmetric Encryption)

對稱式加密又稱祕密金鑰加密系統 [5]，係指傳送方與接收方的加解密皆使用同一把金鑰(或是使用兩個能夠相互推算的密鑰)。也就是只要雙方都擁有這把私鑰，當傳送方傳送資料時，以該私鑰加密，接收方收到訊息後，再使用相同的私鑰解密即可得到明文。

對稱式加密演算法通常計算量小，因此能夠大量且快速地對資料做加解密。在金鑰長度足夠長的情況下，對於攻擊者的暴力破解具有相當高之加密強度，也就是在計算上之安全性是有保障的。然而對稱式加密最大的缺點在於金鑰數量的管理問題，

金鑰的數量會隨使用者數量呈現 O(n^2)，對系統造成負擔。

2.2.2 非對稱式加密(Asymmetric Encryption)

非對稱式加密又稱公開金鑰加密系統 [4]。系統中的每個使用者都擁有一對金鑰：公開金鑰(Public Key)及私密金鑰(Private Key)。運作方式為傳送方在傳送訊息前，需事先取得接收方的公鑰，接著將訊息以此公鑰加密後，再傳送給接收方。而接收方收到加密訊息後，便可利用自己的私鑰解密。

非對稱式加密最大的特色為公開金鑰是可以公開被所有人知道的，知道公鑰的人並無法由此推得私鑰。因此公開金鑰能被廣泛地流傳與發佈，而私密金鑰必須被妥善的保管。較著名的 RSA [6]加密演算法即為非對稱式加密的一種。

2.3 屬性加密(Attribute-Based Encryption, ABE)

屬性加密與其他加密演算法如 RSA、IBE (Identity-Based Encryption) [7]相比，ABE 最大的特色在於實現一對多的加解密。換言之，ABE 不需要在每次的加密過程中預先知道接收者的身分驗證，而是當使用者擁有符合加密者所描述策略的屬性時，即可完成解密。根據存取結構是嵌在金鑰或者密文之中，又可分為 KP-ABE (Key-Policy ABE)和 CP-ABE (Ciphertext-Policy ABE)兩種。其中 KP-ABE 是一種基於私鑰策略的屬性加密，由 Goyql et al. [8]等人所提出，將存取結構與私鑰連結，密文則與一組屬性作連結。而 CP-ABE 是一種基於密文策略的屬性加密，由 Bethencourt et al. [9]等人所提出，將密文與存取結構連結，私鑰則與一組屬性作連結。

以下將以 KP-ABE 為例，說明 ABE 和傳統公鑰加密法的差異。若以傳統公鑰加密，一個資料擁有者需將一明文加密後發送給 N 個不同接收人，要先保存這 N 個接收人的公鑰，再利用 N 組不同的公鑰加密 N 次，生成 N 份不同的密文再分別發送給這 N 個接收人。

然而若使用 KP-ABE 加密演算法，資料擁有者只需根據接收人將屬性嵌入至密文內，而只要能提供符合的存取規則之接收人即可解密。具體說明如圖 2，對接收人 1 而言，只要密文屬性具有高級人員此一屬性，或是同時具有主任以及專案小組兩種屬性則他的金鑰即可對密文做解密；而接受人 2 之金鑰則要求欲解密的密文中須同時嵌有新進員工及專案小組兩種屬性才可解密，但顯然此密文並不符合，因此接受人 2 之金鑰不可對此密文解密。

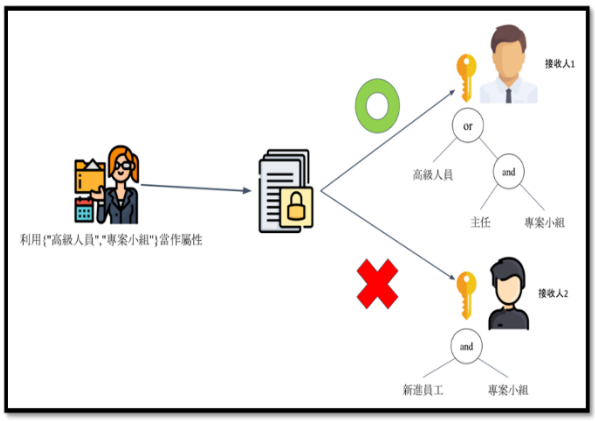

圖 2、屬性加密

3. 研究方法

3.1 設計用戶端介面與網站

本團隊將預先規劃出系統介面之架構圖 [10]，接著架設一個寄收信網站在 Apache 等前端伺服器上，並搭配 JavaScript 建立互動式使用者介面，最後串接 Google 登入功能以及 Google Gmail API 取得 Gmail 資訊，以實現基本的信箱功能。

3.2 建立後端伺服器

前端網頁架設完畢後，接著使用 Amazon EC2 的雲端運算功能以及 Python Flask 開發工具建立後端伺服器(Nmail Server)。在此階段，為了避免信件明文與處理函式都運作在同一個伺服端(Nmail Server)，進而造成資安疑慮，因此額外設計內部伺服器(Private Server)專門處理金鑰存儲及信件的加解密，並將此伺服器(Private Server)架設在私有的主機上以防止外部破解。

- **Nmail Server**

主要以串接 Gmail API 為主，內容包括 Google Account 的 Oauth2.0 授權認證，Gmail 信件的撰寫、標籤和刪除。同時 Nmail Server 作為內部伺服器與

郵件伺服器(Gmail)的接口， 還負責將加密後的信件標頭與使用者的搜尋關鍵字作解密配對，以搜尋相對應之信件。

- **Private Server**

主要以將文字或二進位制的檔案作加解密為主， 此伺服端在接受到密文或明文後，會將所有內容以使用者傳入所欲嵌入之屬性或屬性結構做加密或解密。

3.3 實作 ABE 加密演算法

ABE 加密演算法的部分將依照包含關鍵字多寡的程度來針對內容進行不同的加密處理，目前分為兩大部分：

- **信件內容**

對收發雙方所傳遞的信件內容以及附加檔案進行加密，以保障使用者信件之隱私。使用一般的郵件軟體開啟此信件內容，會呈現亂碼內文，但是透過本團隊所搭建的郵件系統，在關鍵字搜尋到該信件後，便能夠將此密文解密為可閱讀之明文。

- **信件標題**

考慮到標題也可能透漏相關資訊，造成駭客能以標題為參考，暴力破解信件內文，因此將針對標題也進行加密，以達到保護效果。另外，在系統增加搜尋欄功能，能夠對加密過的標題做查詢、分類及刪除。

信件之加解密過程可分為以下五個階段[9]，Setup 階段可先將系統初始化，而 Encrypt 階段將信件以 KP-ABE 的方式加密；TokenGen 階段則生成搜尋權杖(Search Token)，而 Test 階段可在 Nmail Server 端作信件搜尋，最後 Decrypt 階段做密文解密，整體系統架構如圖 3，下列將分別詳細說明各階段執行步驟。

I. Setup 階段

為系統初始化階段，首先會調用 GenKey 函式，生成本系統的公開金鑰 *PK* 和主金鑰 *MK* (Master Key)，存放在內部伺服器(Private Server)。

II. Encrypt 階段

為加密階段，由寄信者制定屬性 *S* (Attribute S)，利用公開金鑰 *PK* 和主金鑰 *MK* 將欲寄送的明文 *M* 加密於定義之屬性下，得到一個只有當收信者提出符合屬性條件的存取結構(Access Structure)才能解密的密文 *CT*。

III. TokenGen 階段

為權杖生成階段，收信者利用主金鑰 *MK* 和以搜尋關鍵字作為存取結構，生成一把搜尋權杖(Search Token)並交由 Nmail Server 做搜尋功用。

IV. Test 階段

在 Nmail Server 中利用搜尋權杖，針對信箱中每封信件標題嘗試解密，若權杖之存取結構符合密文之屬性，則回傳此封信件為搜尋結果，此時內文尚未解密仍呈密文狀態。

V. Decrypt 階段

此部分為解密階段，收信者在 Private Server 中將搜尋結果之密文 *CT* 進行解密得到明文 *M*。

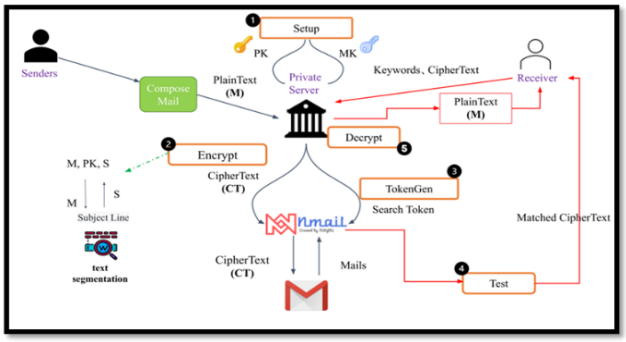

圖 3、系統架構圖

3.4 中文分詞

在屬性加密的過程中，屬性的選擇是由寄信者決定。本團隊預計藉由中文分詞方式[11]，將信件主旨作分詞切割成多個詞句，再由寄信者根據需求選擇作為屬性加密的分詞結果。

本團隊將採用 Jieba 中文分詞，其主要是透過規則斷詞與統計斷詞兩種方法，實現中文的分詞。規則斷詞主要是透過辭典去找對應的詞彙；統計斷詞主要是看如果相連的字出現次數越多，就推斷這相連的字很可能為一個詞。因此，就可以利用字與字相鄰出現的頻率來做統計。當高於某一閾值時，就可以將這個組合視為一個詞。文章若是出現太多不存在於辭典中的專有名詞，也可以透過自定義新

增解決。

4. 研究成果

4.1 加密演算法

本團隊利用 ABE 的 Open Source Library [12]，實作簡易的 KP-ABE 加密。在加密階段以不同的 Sets of Attribute 做加密，再用指定的 Attribute 生成一把解密的 Secret Key，最後用來解回原來的明文。

4.2 Oauth 2.0 驗證

藉由 Gmail API 金鑰就可以取得操作 Gmail API 的權限。取得 API 金鑰後，可藉由設定 Client Id、API Key 和定義可使用的 Scope。來對 Gmail 信件做瀏覽和寄送等功能。

4.3 Demo 展示

圖 4 為登入畫面，當使用者按下登入後，會跳出 Google 的帳戶身分驗證如圖 5，要求使用者同意 Nmail 存取其帳戶權限，也會表明 Nmail 只有使用 Gmail 的信箱相關功能而不會做其他用途。

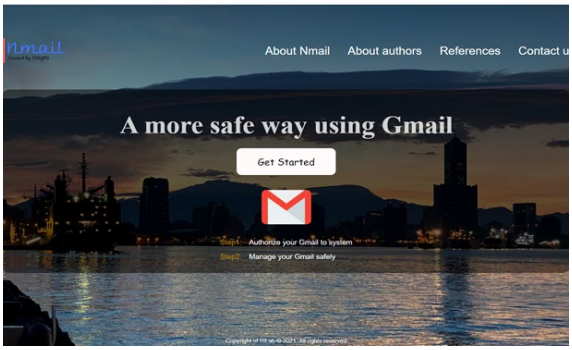

圖 4、登入畫面

圖 5、帳戶身分驗證

圖 6 為登入後的信箱介面，藉由剛剛的 API 驗證後我們可以取得 Gmail 信箱的基本功能，包含標註重要信件、刪除信件等。

圖 6、Nmail 信箱介面

圖 7 為寄信介面，這邊使用「專題指導教授相關問題」作為信件主旨，經由中文分詞後的結果為「專題」、「指導教授」和「相關問題」，可藉由拖曳屬性方塊刪除屬性項目或是自行輸入增加自定義之屬性，如圖中的 "test"。

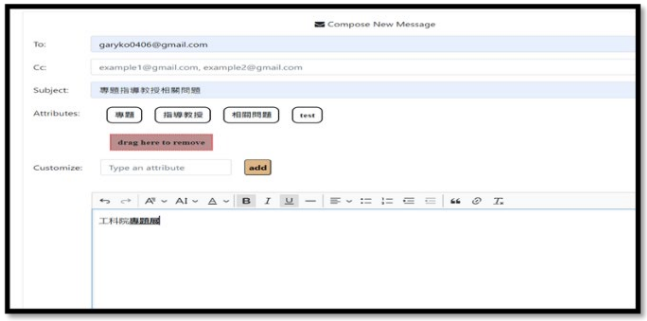

圖 7、寄信介面

圖 8 為經由本系統加密後寄出之信件，在未經過系統搜尋解密前呈現密文狀態，並加註"This mail is sent by Nmail"區分一般信件。

圖 8、密文信件

反之，若是先經由搜尋，以下使用「專題」、「指導教授」作為搜尋關鍵字，系統會將這些關鍵字作為存取結構，生成 ABE 解密之私鑰，若是存取結構符合寄信者所設定之屬性，即可點開信件得到明文，如圖 9、10 所示。

圖 9 關鍵字搜尋

圖 10、明文信件

5. 結論與未來展望

5.1 結論

當需要寄送重要郵件時，便可登入本團隊設計之郵件系統安全地進行收發信件。本系統特色為具有使用者金鑰建立、屬性加密及密文搜尋等功能，並且串接 Gmail API 以讀取使用者的 Gmail 信件。使用者填寫完信件標題後，分詞系統會自動地將其分切成各種屬性，且使用者也可以任意地新增或刪除。

透過本團隊所設計的系統，將可以有效地防止駭客和信件服務提供商(如：Google)惡意地窺視。本團隊所設計之系統，其使用者金鑰建立及加解密過程皆在 Private Server 上，因此即使 Gmail Server 或 Nmail Server 的資料被攻擊者竊取，藉由本系統所寄出的加密信件，因為攻擊者無法破解目前已知的數學難題，所以無法解密以取得信件內容。使用者的信件依然保持安全性，進而達到使用者的隱私保護，避免使用者因為信件內容的外洩而遭受財產損失。

5.2 未來展望

現今，人們以網路傳遞訊息的比例極高，自然是不會希望訊息內容被無關的第三者讀取，但就實際層面而言也不能確定提供訊息傳遞的業者是否可信，因此在日常中的訊息加密上還可以做得更加完善。未來這個加密郵件的概念可套用在公司群組或是醫院的病歷系統.....等注重使用者個人隱私的通訊環境中，落實資訊安全於生活之中。

參考文獻

[1] 范紀鍠, 黃國安, "安全電子郵件系統公鑰自動 取得機制之研製", Diss, 2001.

[2] Merkle Ralph C, "Protocols for public key cryptosystems", *IEEE Symposium on Security and Privacy*, pp. 122-122, 1980.

[3] Diffie Whitfield, and Martin Hellman, "New directions in cryptography", *IEEE transactions on Information Theory*, Vol. 22, No. 6, pp. 644-654, 1976.

[4] Hellman Martin E, "An overview of public key cryptography", *IEEE Communications Magazine*, Vol. 40, Issue 5, pp. 42-49, 2002.

[5] Delfs Hans and Helmut Knebl, "Introduction to cryptography". Vol. 2., Heidelberg: Springer, 2002.

[6] Calderbank Michael, "The RSA cryptosystem: History, Algorithm, Primes", Chicago: math. uchicago.edu, 2007.

[7] Anand Darpan, Vineeta Khemchandani, and Rajendra K. Sharma, "Identity-based cryptography techniques and applications (A Review)", *2013 5th International Conference and Computational Intelligence and Communication Networks*, pp. 343-348, 2013.

[8] Goyal Vipul, Omkant Pandey, Amit Sahai and Brent Waters, "Attribute-based encryption for fine-grained access control of encrypted data", I*n Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89-98, 2006.

[9] John Bethencourt, Amit Sahai, and Brent Waters, "Ciphertext-policy attribute-based encryption", *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007.

[10] Huang Shi-Yuan, "Controllable Predicate Encryption Schemes for Privacy Preserving Search in Cloud Storage", 2013.

[11] Fxsjy, "Jieba chinese text segmentation," in [online] Available: https://github.com/fxsjy/jieba, July 2014.

[12] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems", *Journal of Cryptographic Engineering*, Vol. 3 No.2, pp. 111–128, 2013.