Regular Paper

Communications of the CCISA
Vol. 27 No. 2 May 2021

以屬性加密為基礎的輕量化雙向認證

陳以德[1]、陳胤彤[2]、蔡哲民[3*]
[1,2]高雄醫學大學醫務管理暨醫療資訊學系、[3]崑山科技大學資訊傳播系
[1]itchen@kmu.edu.tw、[2]u108572013@kmu.edu.tw、[3]tjm@fhl.net

摘要

隨著物聯網 (IoT)、5G 與 AI 科技的發展，「智慧醫療」一詞其實就是將科技結合醫療，使得醫院開始在慢慢轉型，本研究以「智慧病房」為例，在於病房的布局上，讓所有連網的感測裝置與病患配對後，進行即時偵測，其中包含了量測病患血壓、脈搏、ECG…等相關生理訊號，然而部分數據屬於高度隱私，如果沒有透過好的加密與驗證機制，這些資料將有可能被洩漏出去或是被不正當的利用。

本研究以屬性加密 (Attribute-based Encryption) 為主要架構，使用者可以選擇符合病患的屬性特徵以制定一些存取規則 (Access policy) ，病患數據只能夠被某些符合其存取規則的特定使用者存取，其他無法滿足存取規則的使用者即使竊取到了數據，也因為透過加密而無法取得其明文內容。在完整驗證階段中以屬性加密為基礎，並在輕量化驗證階段，使用病患的生理數據與時間戳記作為 seed，帶入 PRNG 產生隨機數，在這樣的情況下，僅有三方 (病患、醫生、醫院 Server) 知道計算隨機數，使得有心人士就算竊取到了參數，也因為不知道隨機數該如何製作而無法完成驗證。並加入雜湊、XOR 運算，使得整體加密與驗證更為完善。

關鍵詞：密碼學、屬性加密機制、輕量化認證、存取控制

Regular Paper

Communications of the CCISA
Vol. 27 No. 2 May 2021

A Lightweight mutual authentication based on Attribute-based Encryption

I-Te Chen[1], Yin-Tung Chen[2] , Jer-Min Tsai[3*]
[1,2] Department of Health care Administration and Medical Informatics, Kaohsiung Medical University, Kaohsiung, Taiwan
[3] Department of Information Communication, Kun Shan University, Tainan, Taiwan
[1]itchen@kmu.edu.tw、[2]u108572013@kmu.edu.tw、[3]tjm@fhl.net

Abstract

Recently, With the development of Internet of Things, 5G and AI, the term "smart medical" is actually the combination of technology and medical. Our study takes "smart wards" as an example, which lies in the layout of the wards On the above. After pairing all connected sensor devices with the patient, which includes the measurement of the patient's blood pressure, pulse, ECG... and other related physiological signals. However, some of the data is highly private. If there is no effective encryption and verification mechanism, these data may be leaked out or used improperly.

Our study uses Attribute-based Encryption as the main framework. Users can choose to match the patient's attribute characteristics to formulate some access policies. Patient data can only be accessed by certain access rules. Access by a specific user. Even if other users steal the data, they cannot obtain the plaintext content through encryption. In complete verification phase, ABE is used as the basis, and in the lightweight verification phase, the patient's physiological data and timestamp are used as the seed, and the PRNG is used to generate random numbers. In this case, there are only three parties (patients, doctor, hospital server) know to calculate the random number, so that even if the others steal the parameter, they cannot complete the verification because they don't know how to make the random number. In addition, hash function and XOR operations are added to make the overall encryption and verification more complete.

Keywords: Cryptography, Attribute-based Encryption, Lightweight authentication, Access control

壹、前言

近年來，隨著資訊科技的進步、網路的普及之下，使得我們人類的生活越來越便利，電腦與電腦之間可以透過網路進行通訊，但是在早期，所有的訊息都是以明文的方式進行傳遞，有心人士可以直接擷取訊息，並加以偽造、竄改以達到某些特定目的。鑒於上述原因，使得資訊安全的課題也變得越來越重要，其中密碼學便扮演著關鍵的角色，許多學者提出各種不同的加密方法來保證訊息的安全性，使得我們在網路通訊與資料交換時確保是安全的，但是訊息是安全的還並不算是完整，我們還需要透過有效的認證機制，來確保傳送方與接收方是建立在兩方互信的基礎上。但是在醫療資訊領域中，病患的數據是具有高度隱私的，可是卻沒有一個安全的認證機制，舉例來說：醫生向醫院 Server 要求病患健康資料，但是如果沒有一個好的認證機制，也許醫生這個身分會被其他有心人士偽造，也就是說，透過認證的方式來確認雙方的合法性。

隨著物聯網 (IoT) [1-5]、5G 與 AI 科技的發展，「智慧醫療」[1]一詞其實就是將科技結合醫療，使得醫院開始在慢慢轉型，本研究以「智慧病房」如圖 1 為例，在於病房的佈局上，讓所有連網的感測裝置與病患配對後[2][4]，進行即時偵測，其中包含量測病患血壓、脈搏、ECG…等相關生理訊號，然而部分數據屬於高度隱私，如果沒有透過好的加密與驗證機制，這些資料將有可能被洩漏出去或是被不正當的利用。本研究以屬性加密為基礎，透過存取控制使符合規則的使用者取得數據，並再加入輕量化驗證使驗證效率更為提升。

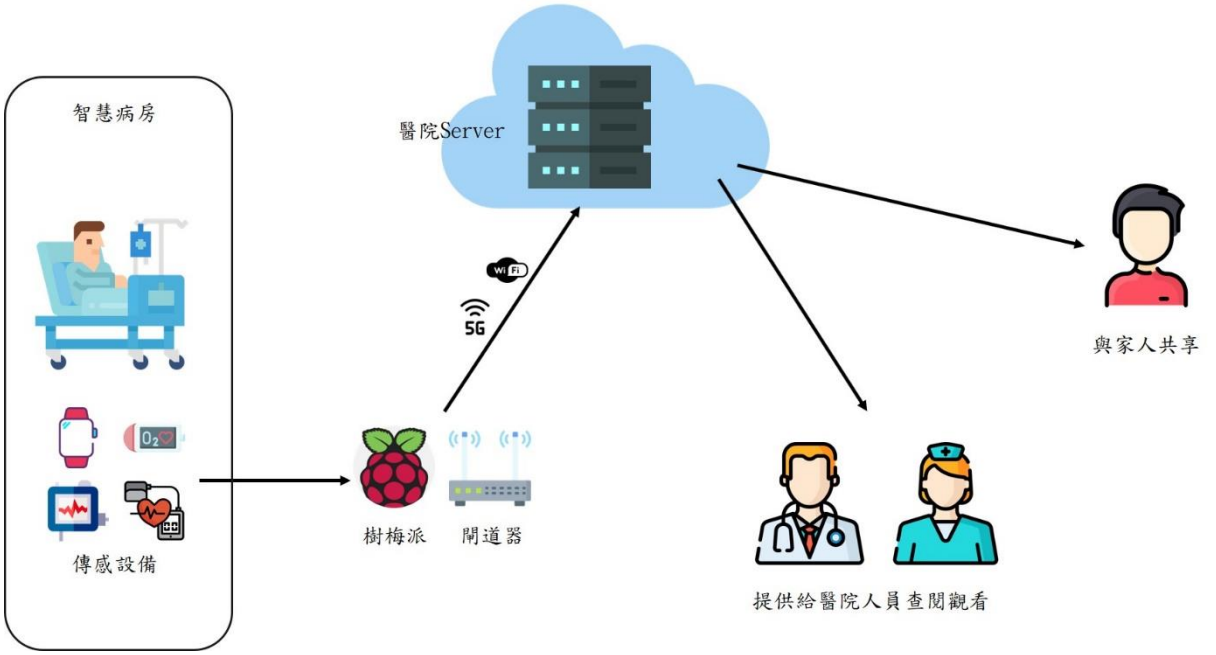

圖 1：智慧病房佈局

有鑑於前言所提到的病患高度隱密資料的存取與驗證問題，在將資料傳送到醫院

Server 儲存之前，得經過一些加密處理與身分驗證。目前主流的加密與驗證系統中，最常見的是以下二種加密方式：

1. 對稱式加密 (Symmetric Key Encryption)
2. 非對稱式加密 (Asymmetric Key Encryption)

上述所提到的二種加密方式及公開金鑰加密基礎設施 (Public Key Infrastructure) 其實都不太適合運用在數據的存取管控上，其中在於面對醫院中大量的病患，每一位病患的資料需根據不同的接收者而擁有不同的存取規則，也就是說，每一份的加密資料只會對應到一把解密的金鑰，如果想要將同一份資料分享給更多的使用者，同一份資料就必須利用各個使用者的屬性特徵或是公開金鑰進行多次的加密，進而使得在金鑰管理上要花很大的成本而且也沒有彈性。

有鑒於此，根據上面的分析，目前最適合運用在本篇論文的加密方法是屬性加密 (Attribute-Based Encryption, ABE) ，使用者可以選擇符合病患的屬性特徵以制定一些存取規則，符合規則的人就允許存取符合該規則的數據，換言之，病患數據只能夠被某些符合其 存取規則的特定使用者存取，其他無法滿足存取規則的使用者即使竊取到了病患數據也因為透過 ABE 加密而無法取得其明文內容。所以，透過 ABE 的加密提供了更富有彈性的控制方法。而 ABE 的計算量過大，所以本研究提出的輕量化驗證來改善 ABE 計算時間太久的問題。

貳、相關研究

2.1 密碼學

密碼學是一種將明文訊息透數學運算來達到隱藏的方法，用於保護訊息在傳遞期間操到竄改或是惡意攻擊。在現代密碼學中，大致將安全性的條件分成了四種：

1. 機密性 (Confidentiality)：資訊內容除了傳送方與接收方外不應該讓第三者獲得。
2. 完整性 (Integrity)：確保接收方得到的資訊內容是完整的。
3. 可驗證性 (Authentication)：接收方可以確認資訊內容為傳送方所送出。
4. 不可否認性 (Non-Reputation)：傳送方不可否認自己所發出的資訊。

由於古典密碼學只達成機密性的安全性要求，因此近代為了滿足這些安全性的密碼系統便不斷的提出不同的加密協定，而在公開金鑰加密系統下，還有「傳統公開金鑰加密系統」、「基於身分認證的加密系統」與「基於屬性認證的加密系統」…等。

2.2 雙線性配對 (Bilinear Pairing)

近年來，雙線性配對函數越來越常在密碼學上廣泛的使用，其使用概念如下，

雙線性配對是指兩個循環群 (Cyclic Group) 之間相對應的線性映射函數 (Bilinear Map) 關係，而橢圓曲線上所有點形成的集合，在代數幾何學中會形成群 (Group) 的關係，因此，雙線性配對的運算恰好能運用在橢圓曲線上。由一個群對應到另外一個群的關係，其相關運算概念如下：其中的 G_1 當中的序 (Order) 為一大質數 q 的循環加法群 (Cyclic Additive Group) ，其生成點 ($genreator$) 是 P，而 G_2 是一個循環式的乘法群 (Multiplicative Group) ， G_2 中的序 (Order) 與G_1皆為一大質數 q，雙線性配對可表示為 $e: G_1 \times G_1 \rightarrow G_2$。在$G_1$和$G_2$中因一大質數 q，使得要解離散對數問題是相當困難的，以下說明其函數之定義與特性：

(1) 雙線性 (Bilinearity)：對於$P, Q \in G_1$ $a, b \in Z_p^*$, $e(aP, bQ) = e(P, Q)^{ab}$
(2) 非退化性 (Non-Degeneracy)：如果 P 是 G_1 中的$genreator$ (生成子)，那 $e(P, P)$也會是 G_2 的 $genreator$ (生成子)，即 $e(P, P) \neq 1$。
(3) 可計算性 (Computability)：$P, Q \in G_1$，存在一演算法可計算 $e(P, Q) \in G_2$

2.3 Access Structure

令 $P = \{P_1, \dots, P_n\}$ 為一個 party 的集合，集和 $\mathbb{A} \subseteq 2^{\{P_1,\dots,P_n\}}$ 是單調性 (monotone) 的，假如$\forall$ $B, C: if$ $B \in \mathbb{A}$ and $B \subseteq C$ $then$ $C \in \mathbb{A}$，則一個存取結構 $\mathbb{A} \subseteq 2^{\{P_1,\dots,P_n\}}$ 是單調性 (monotone) 的。而一個單調性的存取結構在一個非空子集合的 collaction $\mathbb{A}$ 的 $\{P_1, \dots, P_n\}$ ，也就是 $\mathbb{A} \subseteq 2^{\{P_1,\dots,P_n\}} \backslash \{\emptyset\}$。在 $\mathbb{A}$ 的集合則稱作為被授權集合 (authorized sets) ，反之，不在 $\mathbb{A}$ 中的集合則被稱作是非授權集合 (unauthorized sets) 。

通常在屬性加密中，集合中的角色我們將它定義為屬性 (Attribute) ，也就是說在 $\mathbb{A}$ 存取結構中會包含這些授權的屬性，在我們的研究架構中，只考慮了單調的屬性結構。

2.4 屬性加密 (Attribute-Based Encryption)

屬性加密[8]最早是由 Sahai et al.等學者於 1995 年所提出，而其最早的模型概念可以追溯到 Shamir 在 1984 年所提出的 Identity-Based Encryption (IBE) [5]，在 IBE 中，使用者可以選擇一個身分，ex：jerry@，並經由一個可信任的 Private Key Generator (PKG) 產生一個屬於該使用者的身分。然而，不論是 Identity-Based Encryption (IBE) 或是 Public Key Infrastructure (PKI) 當中，兩者加密皆採用一對一的加密，也就是說，每一個的加密資料會對應到一把解密金鑰，在這樣的狀況下，如果想要將同一個資料加密後傳送給其他人，同一份資料就必須根據各個接收者的屬性特徵或是公開金鑰重複加密，如果有 n 個接收者，那就必須加密 n 次，使得在金鑰數目過於龐大，造成管理不易且較沒有彈性。

然而在 Sahai et al.[8]等學者提出了 Fuzzy Identity-Based Encryption，該學者們提出的模型也被認為是 ABE 的前身。使用者在加密資料時定義有閾值的存取規則 (Access

policy)，加密後的密文配對特定的屬性，解密時根據接收者手上所握有的屬性 (attribute) 來決定該接收者是否符合他的存取規則，符合的話接收者則得以將資料解密。所以可以將 ABE 視為 IBE 的延伸，只是原本在 IBE 中是以身分 (Identity) 作為代表；ABE 中則是以屬性 (attribute) 的集合作為代表。因此，ABE 的出現，解決了 IBE 一對一加密的缺點與金鑰管理上的困難度，其中 ABE 與傳統上的加密系統最大的差別在於它可以是一種一對多的加密，在加密時將存取規則 (Access policy) 加入接收者的屬性，使得接收者在收到時可以根據自己私密金鑰所擁有的屬性集合是否符合，如果符其存取規則就能解密。正是基於這樣的特性，使得 ABE 在 IBE 與 PKI 相較之下更富有彈性，同時也降低使用者或是管理者在管理上的複雜度。

目前在屬性加密中，主要分為了 Ciphertext-Policy Attribute Based Encryption (CP-ABE) [6,10-12]與 Key-Policy Attribute-Based Encryption (KP-ABE) [7,13]，兩種 ABE 最大的差別在於一個是將存取規則與私密金鑰配對；另一個是將存取規則與訊息進行加密成密文。

2.4.1 Key Policy Attribute-Based Encryption (KP-ABE)

KP-ABE 是由 Goyql et al.[7] 等學者所提出，在這個加密系統中，制定者設定好屬性後與明文加密後配對，也就是說該密文與一組屬性作連結，而存取規則 (Access policy) 與私密金鑰 (Private key) 做連結，接收者依據自己手上私密金鑰中配對的存取規則是否滿足屬性而決定能否解密，如圖 2 所示。

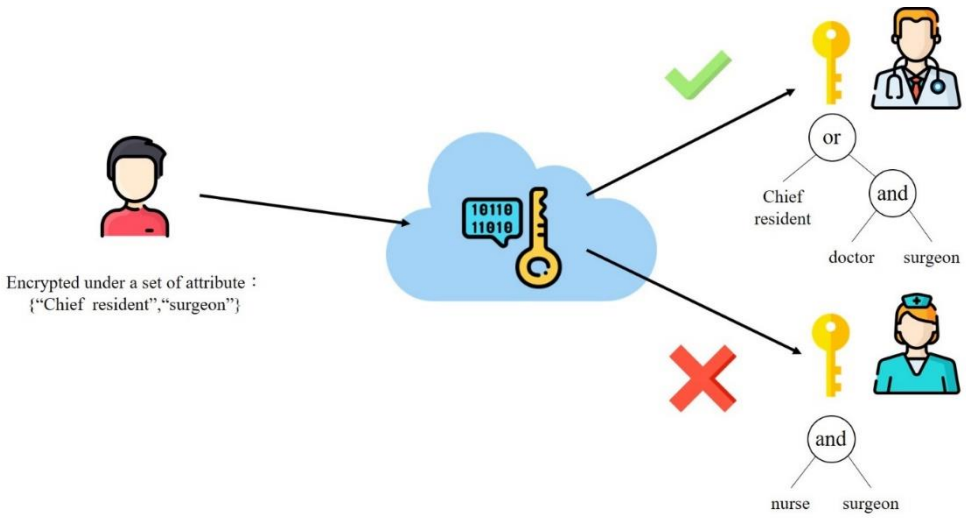

圖 2：KP-ABE

KP-ABE 演算法是建立在 symmetric bilinear pairing，$e: G_0 \times G_0 \rightarrow G_1$，$G_0$的 order 是質數$p$，$g$是$G_0$的一個生成點 $(genreator)$，以及定義一個雜湊函數$H = \{0,1\}^* \rightarrow G_0$。此加密方法主要分為四個階段：系統參數設定 (Setup)、金鑰產生階段 (Key generation)、

加密階段 (Encryption) 、解密階段 (Decryption) 。

Setup

定義一個屬性宇集合 attribute universe $\mathcal{U} = \{1,2,\dots,n\}$，其中 n 為自然數，替每一個在 $\mathcal{U}$ 中的屬性 i 選擇一個值t_i，其中 $t_i \in Z_p^*$，並且選擇一個$y \in Z_p$，並製作公開金鑰為：PK = $(T_1 = g^{t1}, \dots T_n = g^{tn}, Y = e(g,g)^y)$

而主金鑰(Master Key)為：MK = $(t_1, \dots t_n, y)$

Key generation

此演算法會回傳一個私密金鑰給使用者，該金鑰與存取規則 (Access Structure) 做連結。首先會先將存取規則轉成 Access tree，對每一個在 Access tree 中的非業節點x，選定一個多項式p_x，而此多項式的 degree 為$t_x - 1$，對根節點 R 設定$p_R(0) = y$，其他的節點x設定為$p_R(0) = p_{parrent(x)}(index(x))$，再選擇$t_x - 1$個任意常數來完成這個多項式，最後，令$S$為 Access tree 中葉節點的集合，對每一個屬性所連結的葉節點，產生私密金鑰：

$$D_x = g^{\frac{p_x(0)}{t_{att(x)}}} \ for \ x \in S$$

Encryption

選擇一個 secret s，其中$s \in Z_p$，並將一個屬性集合 γ 與訊息 M 做加密，其密文為：$E = (\gamma, E' = MY^S, \{E_i = T_i^S\}_{i\in\gamma})$

Decryption

首先定義一個遞迴演算法DecrypNode(CT, SK, x)，對每一個在 access tree 中的葉節點 x：

$$\text{DecrypNode(CT,SK,x)}=\left\{e(D_x,E_i) = e\left(g^{\frac{p_x(0)}{t_i}}, g^{st_i}\right) = e(g,g)^{sp_x(0)} \ for \ i = att(x) \in \gamma\right\}$$

$$\perp otherwise$$

由於$p_x(0) = p_{parrent(x)}(index(x))$，對於每個節點$x$，利用內插法得到$e(g,g)^{sp_x(0)}$，一直重複計算到根節點，最後可得到這個根結點 R ：

$$\left\{e(D_R,E_i) = e\left(g^{\frac{p_R(0)}{t_i}}, g^{st_i}\right) = e(g,g)^{sp_R(0)} = e(g,g)^{sy} = Y^s\right\}$$

即可得到明文：$M = \frac{E\prime}{Y^s}$

2.4.2 Ciphertext-Policy Attribute Based Encryption (CP-ABE)

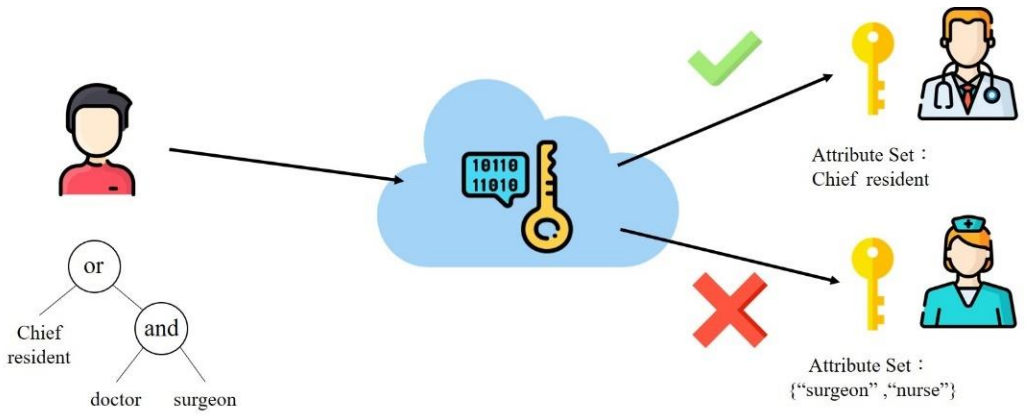

圖 3：CP-ABE

CP-ABE 是由 Bethencourt et al. [6]等學者所提出，與 KP-ABE 正好相反，CP-ABE 是將存取規則 (Access policy) 連結與密文連結，私密金鑰則與一組屬性作連結。例：制定者可以自訂其屬性策略，亦或者是可以將接收者的屬性加入自身的屬性策略中，也就是「授權」，接收者在收到時可以根據自己私密金鑰所擁有的屬性集合是否符合，如果符合該密文配對的存取規則就能將其解密。如圖 3 所示。CP-ABE 演算法是建立在 symmetric bilinear pairing，$e: G_0 \times G_0 \rightarrow G_1$，$G_0$的 order 是質數$p$，$g$是$G_0$的一個生成點 $(genreator)$，以及定義一個雜湊函數$H = \{0,1\}^* \rightarrow G_0$。此加密方法主要分為四個階段：系統參數設定 (Setup) 、金鑰產生階段 (Key generation) 、加密階段 (Encryption) 、解密階段 (Decryption) 。

Setup

選擇變數$\alpha, \beta \in Z_p$，並製作公開金鑰：$\mathrm{PK} = (G_1, \mathrm{g}, \mathrm{h} = g^{\beta}, \mathrm{f} = g^{\frac{1}{\beta}}, e(g,g)^{\alpha})$

而主金鑰(Master Key)為：$\mathrm{MK} = (\beta, g^{\alpha})$

Key generation

此演算法會輸入一組屬性集合 S ，最後輸出一對金鑰 (公開金鑰、私密金鑰) 對應到此集合 S 。首先任意選取一個亂數 $\mathrm{r} \in Z_p$，針對每個屬性 $\mathrm{j} \in S$，任意選取一個亂數 $r_j \in Z_p$，最後運算出私密金鑰為：$SK = (D = g^{\frac{\alpha+r}{\beta}} \ \forall \ j \in S : D_j = g^r H(j)^{r_j}, D'_j = g^{r_j})$

Encryption

此演算法會將訊息 M 加密於存取規則 (Access policy) $\mathbb{A}$ 之下， 首先會先將存取規則轉成 Access tree，針對 tree 中的每一個節點選定一個多項式p_x，由根節點 R 開始，選定一個多項式p_R，而此多項式的 degree 為$t_R - 1$。選擇一個 secret s，其中$s \in Z_p$，並設定$p_R(0) = s$，再選擇$t_x - 1$個任意常數來完成這個多項式。其他的節點x設定為$p_x(0) =$

$p_{parrent(x)}(index(x))$，同樣地，再選擇$t_x - 1$ 個任意常數來完成這個多項式 p_x。最後，令S為 Access tree 中葉節點的集合，則密文為：

$CT = (\mathrm{A}, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s \ \forall \ x \in X : C_x = g^{p_x(0)}, C'_x = H(att(x))^{p_x(0)})$

Decryption

根據接收者手上握有私密金鑰 SK 來解密文 CT ，假如接收者的屬性集合 S 滿足 Access structure，即可將 CT 解密回訊息 M。以下詳細說明其遞迴演算法。

對於每個非葉節點 x ，令 $i = (att(x))$ ，若 $i \in S$，則

$$\mathrm{DecrypNode}(\mathrm{CT}, \mathrm{SK}, \mathrm{x}) = \frac{e(D_i, C_x)}{e(D'_i, C'_x)} = \frac{e(g^r H(i)^{r_i}, g^{p_x(0)})}{e(g^{r_i}, H(i)^{p_x(0)})}$$

$$= \frac{e(g^r H(i)^{r_i}, g^{p_x(0)})}{e(g^{p_x(0)}, H(i)^{r_i})} = \frac{e(g^r g^{p_x(0)}, g^{p_x(0)})}{e(H(i)^{r_i}, H(i)^{r_i})} = \frac{e(g^r g^{p_x(0)}, g^{p_x(0)})}{1}$$

$$= e(g,g)^{rp_x(0)}$$

若 $i \notin S$：

$$\mathrm{DecrypNode}(\mathrm{CT}, \mathrm{SK}, \mathrm{x}) = \perp \ otherwise$$

若 x 為非葉節點，考慮遞迴的情況：

令 z 為 x 的子節點，呼叫$\mathrm{DecrypNode}(\mathrm{CT}, \mathrm{SK}, \mathrm{x})$，輸出 F_z ，令 S_x 為大小 t_x 的任意集合，收集這些子節點 z ，使得 $F_z \neq \perp$，如果 S_x 不存在，則

$$\mathrm{DecrypNode}(\mathrm{CT}, \mathrm{SK}, \mathrm{x}) = \perp \ otherwise$$

$$F_z = \prod_{z \in S_x} F_z{}^{\Delta_{i,S'_x}(0)}, where \ i = index(z), S'_x = \{index(z) : z \in S_x\}$$

$$= \prod_{z \in S_x} (e(g,g)^{rp_z(0)})^{\Delta_{i,S'_x}(0)}$$

$$= \prod_{z \in S_x} (e(g,g)^{p_{parrent(z)}(index(z))})^{\Delta_{i,S'_x}(0)} \prod_{z \in S_x} (e(g,g)^{rp_x(i)})^{\Delta_{i,S'_x}(0)} = e(g,g)^{rp_x(0)}$$

最後一路從葉節點算回至根結點R：$e(g,g)^{rp_R(0)}$，可得$\mathrm{DecrypNode}(\mathrm{CT}, \mathrm{SK}, \mathrm{x}) = e(g,g)^{rp_R(0)} = e(g,g)^{rs}$，最後可算出明文：

$$M = \frac{\tilde{C}}{e(C,D)/e(g,g)^{rs}} = \frac{\tilde{C}}{e(h^s, g^{\frac{\alpha+r}{\beta}})/e(g,g)^{rs}} = \frac{\tilde{C}}{e(g^{\beta s}, g^{\frac{\alpha+r}{\beta}})/e(g,g)^{rs}}$$

$$= \frac{\tilde{C}}{e(g,g)^{s(\alpha+r)}/e(g,g)^{rs}} = \frac{\tilde{C}}{e(g,g)^{s(\alpha+r)}/e(g,g)^{rs}} = \frac{\tilde{C}}{e(g,g)^{\alpha s}}$$

2.4.3 KP-ABE & CP-ABE 小結

由表 1 可看出兩個演算法有對偶的關係,但是在實際應用中,兩個是有很大的差別,其中 CP-ABE 是將 Access policy 與密文配對,這也就代表數據擁有者可以自訂其屬性策略,使得接收者在收到時可以根據自己所擁有的屬性集合是否符合,如果符合 Access policy 就能將密文解開;KP-ABE 正好相反,在加密前先制定好屬性集合,將屬性集合與密文配對,解密時根據自己擁有的密鑰中配對的 Access policy 是否符合,得以解密。因此,根據上述比較,本研究採用 CP-ABE 作為我們完整驗證中的主要演算法。

表 1：兩種屬性加密比較

	KP-ABE	CP-ABE
屬性相關	與密文配對	與密鑰配對
Access policy 相關	與密鑰配對	與密文配對
數據擁有者能否決定授權給誰	否	能
計算能力要求	較低	較高
抵禦同謀攻擊	較低	較高

參、系統架構與實作

3.1 開發環境

表 2：硬體規格

CPU	Intel(R) Xeon(R) CPU E3-1230 v3
GPU	NVIDIA GT 630
RAM	12GB
SSD	500GB

作業系統採用 ubuntu 18.04.5LT，為了實現 CP-ABE 演算法，需安裝 PBC (Pairing-Based Cryptography) 函式庫 (版本：0.5.14) 、GMP 函式庫 (版本：6.2.1) 由 M4、bison、flex 等套件組成，並安裝 openssl (版本：l-1.1.1k)、libbswcpabe (版本：0.9)、cpabe (版本：0.11) 等套件，建構出實驗環境。

3.2 系統架構

在本研究的架構中，主要有四個角色：屬性授權中心 (AA) 、被授權者、授權者與醫院 Server。

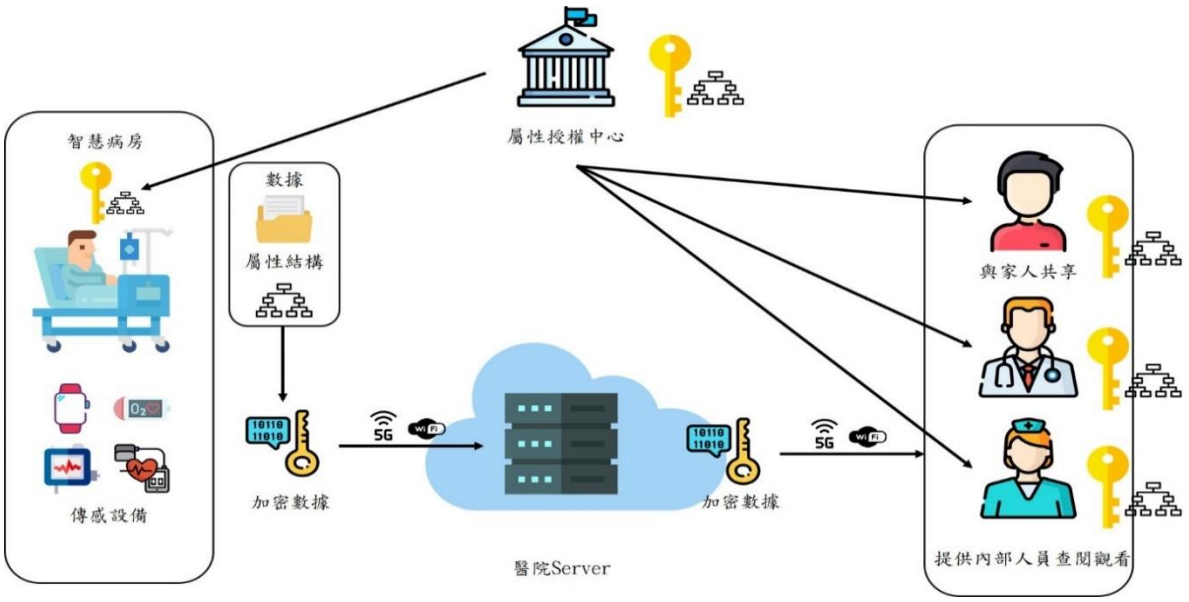

圖 4：系統架構

表 3：角色介紹

名稱	說明
屬性授權中心(AA)	管理授權人的屬性集合，並分配屬性給被授權人
授權人	病人，將數據加密後傳到 Server 儲存
被授權人	醫生或護士，申請數據查閱
醫院 Server	儲存資料，並提供醫生、護士查閱

表 4：參數介紹

名稱	說明	名稱	說明
ID_p, ID_d	Patient Id, Doctor Id	$e(g,g)$	Bilinear Pairing
MK	*master key*	$\oplus$	XOR 運算
PK	*public key*	$\|\|$	串接
SK	*secrey key*	*timestamp*	時間戳記
P_1, P_2	由 Patient 端製作	M	明文
D_1, D_2	由 Doctor 端製作	CT	密文
S_1, S_2, S_3, S_4	由 Server 端製作	$HMAC_{sk_p}, HMAC_{sk_d}$	一次性金鑰雜湊
sk_p, sk_d	*secret value*	$h()$	雜湊演算法
X_1, Y_1, X_2, Y_2	驗證參數	r_1, r_2, r_3, r_4	透過 PRNG
ID_pC_1	密文	*session key*	會議密鑰
M_1, M_2	一次性金鑰雜湊值		

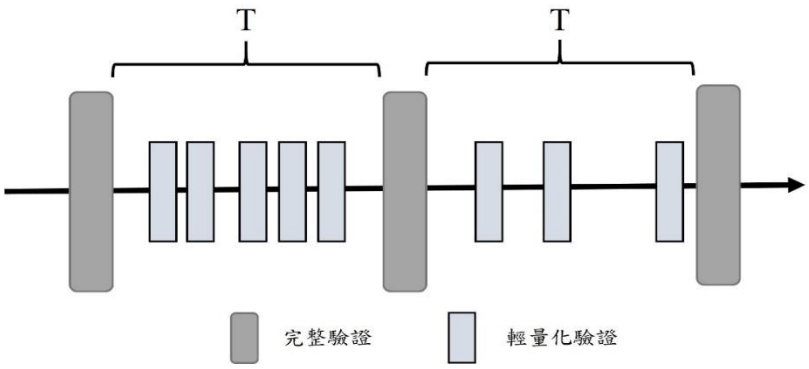

圖 5：驗證示意圖

3.3 完整驗證

採取與 Bethencourt et al [4]等學者所提出之演算法，說明於 2.4.2，並將其演算法加入時間戳記之概念，使得明文資料在傳輸時以$M = (D_{1ID_{p1}}|| \dots ||D_{1ID_{pi}}||timestamp_n)$。完整驗證中分為了五個階段：系統初始階段、病患註冊階段、金鑰產生階段、加密階段、解密階段。

1.系統初始階段：

由屬性授權中心產生PK、SK等參數

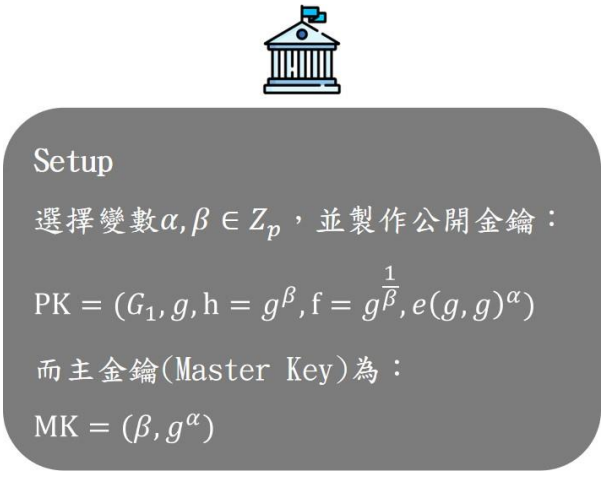

圖 6：系統初始階段

2.病患註冊階段：

當有病患入院時，首先指派ID_p給病人，並秘密決定sk_p，透過 Secure channel 傳給病人與醫院 Server，sk_p僅有病患與醫院 Server 端知道，第三方並無權限取得。再來由屬性授權中心根據病患給予特定屬性特徵，譬如有一個病患來 A 醫院的神經內科看診，

屬性授權中心可賦予該病患"A 醫院"、"神經內科"、"病患姓名"、"性別"、"年齡"...等屬性，倘若需要將病患數據授權給其他醫生或護士查閱，可以再將病患的存取規則 (Access policy) 中加入其他人的屬性，建構出一個新的存取規則，使得對方收到加密資料時可以根據自己私密金鑰所擁有的屬性是否符合得以解密。

圖 7：病患註冊階段

3.金鑰產生階段

在此階段進行時會產生一個私密金鑰(SK)給病患，該金鑰與病患存取規則做配對。首先會先將存取規則轉成 Access tree，對每一個在 Access tree 中的非葉節點x，選定一個多項式p_x，而此多項式的 degree 為$t_x - 1$，對根節點 R 設定$p_R(0) = y$，其他的節點x設定為$p_R(0) = p_{parrent(x)}(index(x))$，再選擇$t_x - 1$個任意常數來完成這個多項式，最後，令$S$為 Access tree 中葉節點的集合，對每一個屬性所連結的葉節點，產生私密金鑰(SK)：$SK = (D = g^{\frac{\alpha+r}{\beta}} \ \forall j \in S : D_j = g^r H(j)^{r_j}, D'_j = g^{r_j})$

再透過 Secure Channel 回傳至病人端。

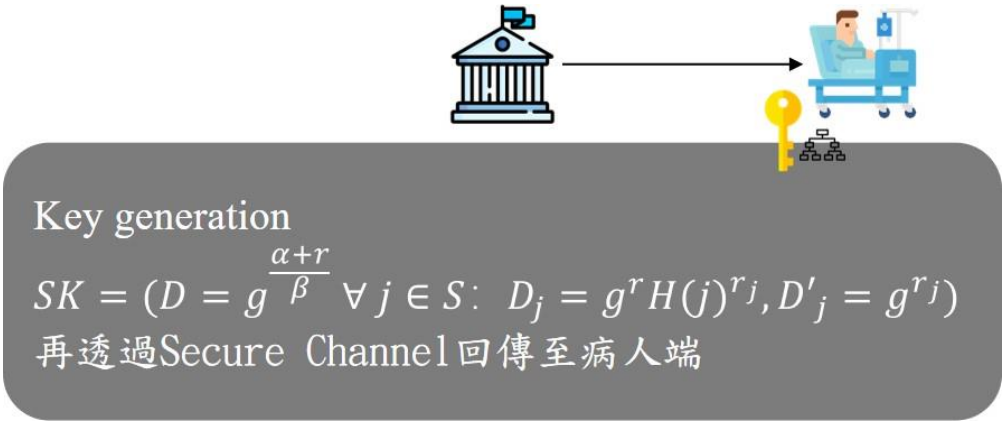

圖 8：金鑰產生階段

4.加密階段

此階段由病患端於一段時間蒐集完數據後，其生理數據可略分為心跳、血壓、血氧、呼吸等四大類。並取得時間戳記，使用 unix time 格式，例：2021 年 6 月 30 日 上午 12:00:00，其值為 1625025600，將數據 $M = (D_{1ID_{p1}}|| \dots ||D_{1ID_{pi}}||timestamp_n)$，使用自己的私密金鑰進行加密，加密 CT再傳到醫院 Server 保存。

Encryption

此演算法會將訊息 M 加密於Access structure A 之下，首先會先將存取規則轉成Access tree

$$\mathrm{CT} = (\mathrm{A}, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s \; \forall x \in X : C_x = g^{p_x(0)}, C'_x = H(att(x))^{p_x(0)})$$

圖 9：加密階段

5.解密階段

醫院 Server 收到加密 CT 儲存至資料庫，提供被授權人 (醫生、護士) 申請查閱。如果需進行輕量化驗證，醫院 Server 使用私密金鑰進行解密，因為病患的 Access policy 有授權醫院 Server 的屬性，解密後將$(D_{1ID_{p1}}|| \dots ||D_{1ID_{pi}}||timestamp_n)$作為 seed 使用。

Decryption

從葉節點算回至根結點R：$e(g,g)^{rp_R(0)}$，

可得$\mathrm{DecrypNode}(\mathrm{CT}, \mathrm{SK}, \mathrm{x}) = e(g,g)^{rp_R(0)} = e(g,g)^{rs}$

最後可算出明文：

$$M = \frac{\tilde{C}}{e(C,D)/e(g,g)^{rs}} = \frac{\tilde{C}}{e(h^s, g^{\frac{\alpha+r}{\beta}})/e(g,g)^{rs}}$$

$$= \frac{\tilde{C}}{e(g^{\beta s}, g^{\frac{\alpha+r}{\beta}})/e(g,g)^{rs}} = \frac{\tilde{C}}{e(g,g)^{s(\alpha+r)}/e(g,g)^{rs}} = M$$

圖 10：解密階段

3.4 輕量化驗證

其架構分為兩個階段，以下將進行詳細說明：授權者 (病患) 與醫院 Server 驗證階段與被授權者 (醫生或是護士) 醫院 Server 驗證階段，兩個驗證皆使用*session key*加密。

3.4.1 授權者與醫院驗證階段

步驟 1：授權者 (病患) 採定期、定時將資料傳送至醫院 Server 儲存，由病患發起驗證要求，透過 Random number 產生r_1參數，其中r_1參數於 3.3.3 小節中介紹如何製作。取得完整驗證中儲存的sk_p，計算$X_1 = h(sk_p \oplus r_1)$，$P_1 = h(r_1 \oplus h(ID_p)) \oplus h(sk_p)$，$P_2 = HMAC_{sk_p}(ID_p, X, P_1)$，最後將$ID_p$、$X_1$、$P_1$、$P_2$傳送至醫院 Server。

步驟 2：當醫院 Server 收到ID_p、X_1、P_1、P_2四個參數時，首先根據ID_p找到對應的sk_p，接著透過一次性金鑰雜湊值計算$P'_2 = HMAC_{sk_p}(ID_p, X, P_1)$，比對$P'_2 = P_2$，如果不相等，表示在傳輸參數時遭到竄改或偽造，拒絕此次驗證要求。如果相等，則繼續計算$X'_1 = h(sk_p \oplus r_1)$，$P'_1 = h(r_1 \oplus h(ID_p)) \oplus h(sk_p)$，如果$X'_1 = X_1$且$P'_1 = P_1$，表示以確認病患端身分真實性。接下來將由醫院 Server 製作參數傳回至病患端做驗證。

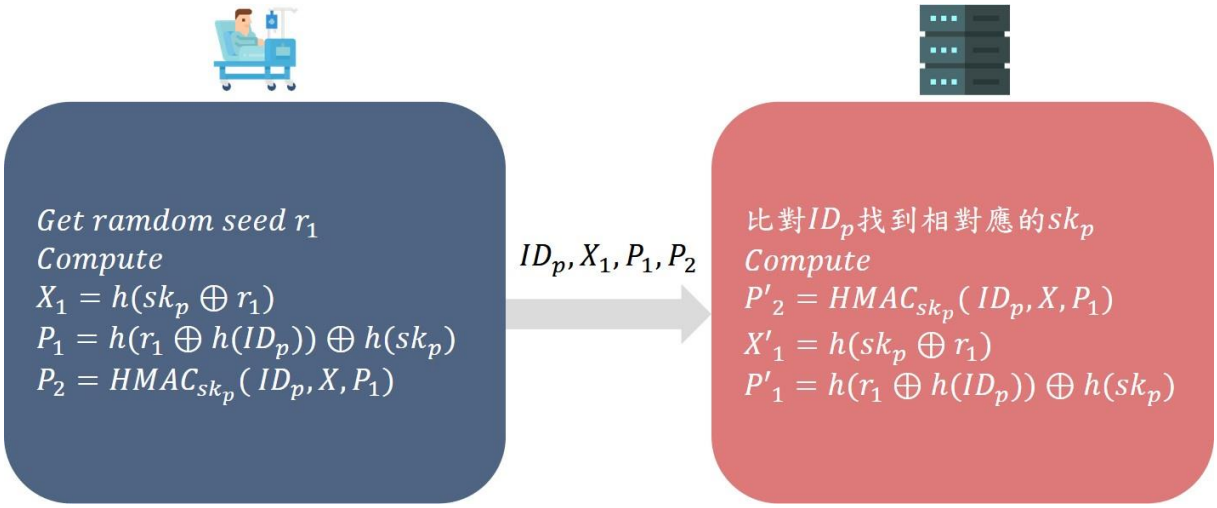

圖 11：病患驗證階段

步驟 3：醫院 Server 透過 Random number 產生r_2參數，其中r_2參數於 3.3.3 小節中介紹如何製作。計算$Y_1 = h(r_1) \oplus h(sk_p \oplus r_2)$，$S_1 = h(Y \oplus ID_p) \oplus (sk_p)$，$S_2 = HMAC_{sk_p}(r_1, r_2, Y, S_3)$，並決定會議金鑰$session\ key = h(r_1||r_2||sk_p||timestamp_n)$，其中$timestamp_n$是前一次驗證時所記錄下來的時間戳記，最後將$Y_1, S_1, S_2$傳送至病患端。

步驟 4：當病患端收到Y_1, S_1, S_2三個參時，透過一次性金鑰雜湊值計算$S'_2 = HMAC_{sk_p}(r_1, r_2, Y, S_3)$，比對$S'_2 = S_2$，如果不相等，表示在傳輸參數時遭到竄改或偽造，拒絕此次驗證要求。如果相等，則繼續計算$S'_1 = h(Y \oplus ID_p) \oplus (sk_p)$，$Y'_2 = h(r_1) \oplus h(sk_p \oplus r_2)$，如果$S'_1 = S_1$且$Y'_1 = Y_1$，表示以確認醫院 Server 端身分真實性。並決定會

議金鑰$session\ key = h(r_1||r_2||sk_p||timestamp_n)$，其中$timestamp_n$是前一次驗證時所記錄下來的時間戳記。

步驟 5：病患端蒐集完數據後，取得當下時間$timestamp_{n+1}$，將數據集加密，數據會有好幾筆，以第一筆為例，透過 $session\ key$ 加密，密文 $ID_pC_1 = E_{session\ key}(D_{1ID_{p1}}|| \dots ||D_{1ID_{pi}}||timestamp_{n+1})$，製作$M_1 = HMAC_{sk_d}(ID_pC_1|| \dots ||ID_pC_k)$，最後將$ID_pC_1 \dots ID_pC_k$ 、M_1傳送至醫生端。

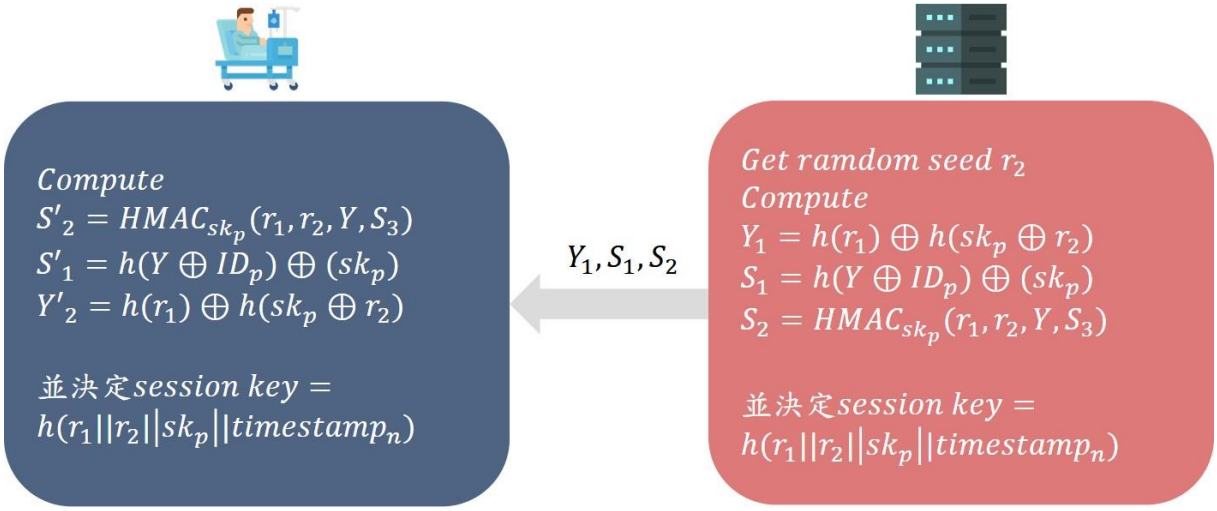

圖 12：醫院 Server 驗證階段

步驟 6：醫院 Server 收到 $ID_pC_1 \dots ID_pC_k, M_1$，首先計算 $M'_1 = HMAC_{sk_d}(ID_pC_1|| \dots ||ID_pC_k)$，比對$M'_1 = M_1$，示在傳輸參數時遭到竄改或偽造，拒絕此次連線。如果相等，即可將C_{ID_p}使用$session\ key$解密後儲存，提供被授權人 (醫生、護士) 申請查閱。

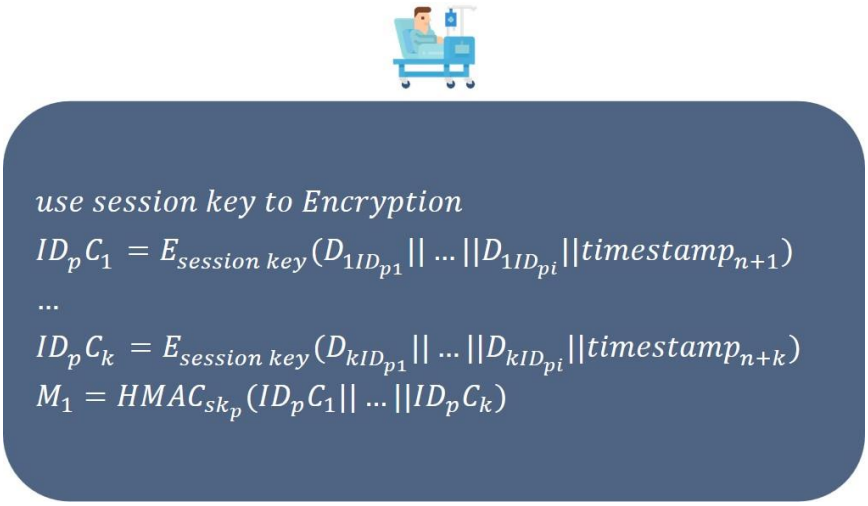

圖 13：$session\ key$加密傳輸資料

3.4.2 被授權者與醫院驗證階段

步驟 1：由醫生發起驗證要求，透過 Random number 產生r_3參數，其中r_3參數於 3.3.3 小節中介紹如何製作。取得完整驗證中儲存的sk_d，計算$X_2 = h(sk_d \oplus r_3)$，$D_1 = h(r_3 \oplus h(ID_d)) \oplus h(sk_d)$，$D_2 = HMAC_{sk_d}(ID_d, X_1, D_1)$，接著取得病患$ID_p$，用來告知醫院 Server 要申請查閱的數據，最後將$ID_d, ID_p, X_2, D_1, D_2$ 傳送至醫院 Server。

步驟 2：當醫院 Server 收到$ID_d, ID_p, X_2, D_1, D_2$五個參數時，首先根據$ID_d$找到對應的$sk_d$，接著透過一次性金鑰雜湊值計算$D'_2 = HMAC_{sk_d}(ID_d, X_1, D_1)$，比對$D'_2 = D_2$，如果不相等，表示在傳輸參數時遭到竄改或偽造，拒絕此次驗證要求。如果相等，則繼續計算$X'_2 = h(sk_d \oplus r_3)$，$D'_1 = h(r_3 \oplus h(ID_d)) \oplus h(sk_d)$，如果$X'_2 = X_2$且$D'_1 = D_1$，表示以確認醫生端身分真實性。接下來將由醫院 Server 製作參數傳回至病患端做驗證。

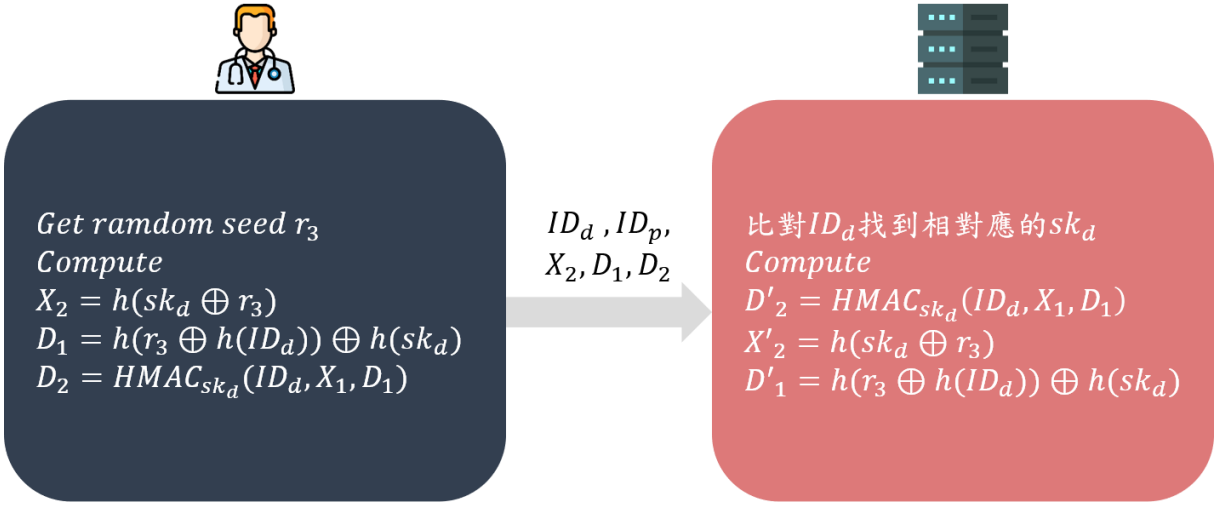

圖 14：醫生驗證階段

步驟 3：醫院 Server 透過 Random number 產生r_3參數，其中r_4參數於 3.3.3 小節中介紹如何製作。計算$Y_2 = h(r_3) \oplus h(sk_p \oplus r_4)$，$S_3 = h(Y_2 \oplus ID_d) \oplus (sk_p)$，$S_4 = HMAC_{sk_p}(r_3, r_4, Y_2, S_3)$，並決定會議金鑰$session\ key = h(r_3||r_4||sk_d||timestamp_n)$，其中$timestamp_n$是前一次驗證時所記錄下來的時間戳記，最後將$Y_2, S_3, S_4$傳送至病患端。

步驟 4：當醫生端收到Y_2, S_3, S_4三個參時，透過一次性金鑰雜湊值計算$S'_4 = HMAC_{sk_p}(r_3, r_4, Y_2, S_3)$，比對$S'_4 = S_4$，如果不相等，表示在傳輸參數時遭到竄改或偽造，拒絕此次驗證要求。如果相等，則繼續計算$S'_3 = h(Y_2 \oplus ID_d) \oplus (sk_p)$，$Y'_2 = h(r_3) \oplus h(sk_p \oplus r_4)$，如果$S'_3 = S_3$且$Y'_2 = Y_2$，表示以確認醫院 Server 端身分真實性。並決定會議金鑰$session\ key = h(r_3||r_4||sk_d||timestamp_n)$，其中$timestamp_n$是前一次驗證時所記錄下來的時間戳記。

步驟 5：首先醫院 Server 取得當下時間$timestamp_{n+1}$，根據步驟 1 由醫生端所提供的ID_p找到對應的數據集，數據會有好幾筆，以第一筆為例，透過$session\ key$加密，密文$ID_pC_1 = E_{session\ key}(D_{1ID_{p1}}|| \dots ||D_{1ID_{pi}}||timestamp_{n+1})$，製作$M_2 = HMAC_{sk_d}(ID_pC_1|| \dots ||ID_pC_k)$，最

後將$ID_pC_1, \ldots ID_pC_k$、M_2傳送至醫生端。

步驟 6：醫生端收到$ID_pC_1, \ldots ID_pC_k, M_2$先計算$M'_2 = HMAC_{sk_d}(ID_pC_1 || \ldots || ID_pC_k)$，比對$M'_2 = M_2$，如果不相等，表示在傳輸參數時遭到竄改或偽造，拒絕此次連線。如果相等，即可將C_{ID_p}使用$session\ key$解密後即可查閱原始資料。

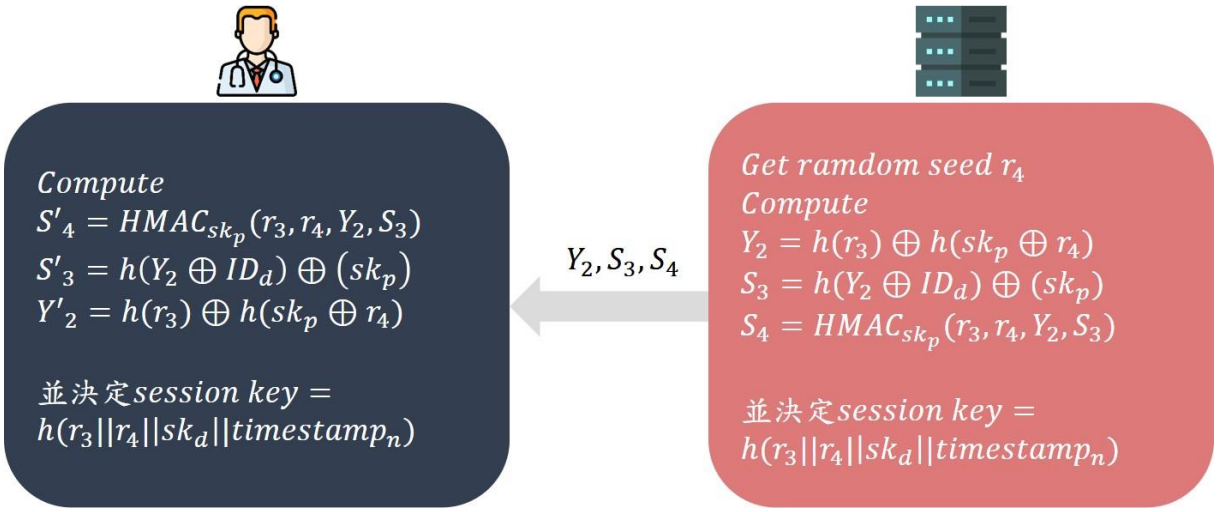

圖 15：醫院 Server 驗證階段

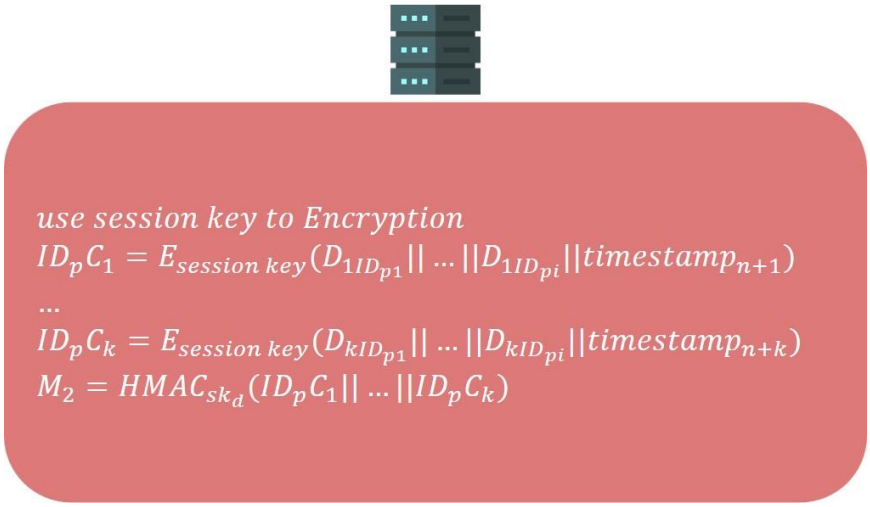

圖 16：$session\ key$加密傳輸資料

3.4.3 Random number 製作

當相同的值 (seed) 代入 Pseudo Random number generator (PRNG) 時，會得到一樣的隨機數。因此，在我們所提出驗證架構中，使用 PRNG，使得兩端在驗證時不用傳遞隨機數，對方也能透過演算法算出一樣的 Random number。以下將以r_1、r_2、r_3、r_4的產生方式進行解說。

這四個隨機數的製作方式皆相同，其中r_1由病患端製作，r_3醫生端製作，r_2、r_4接由醫院 Server 端製作，四個隨機數由前一次驗證時，根據傳輸數據的筆數，而有不同的產生方式，其中，我們使用移位運算子將資料移位，第 n 筆資料就向右移 n 個位元。在筆

$$seed_1 = (D_{1ID_{p1}} || \dots || D_{1ID_{pi}} || timestamp_{n+1})$$
$$seed_2 = \gg 1(D_{2ID_{p1}} || \dots || D_{2ID_{pi}} || timestamp_{n+2})$$
$$\dots$$
$$seed_n = \gg n-1(D_{kID_{p1}} || \dots || D_{kID_{pi}} || timestamp_{n+k})$$

圖 17：seed 製作圖

數為奇數筆時，選擇前一次傳輸數據的首筆、末筆與中間那筆當作執行 xor 運算後當作 seed；在比數為偶數筆時，選擇前一次傳輸數據的首筆與末筆執行 xor 運算後來當作 seed。如圖 17 所示，最後，經過 PRNG 產生r_1、r_2、r_3、r_4。

肆、研究結果與討論

4.1 安全性分析

在本小節中，針對本研究所提出輕量化驗證，所達到的安全性包含了抵禦重送攻擊 (Resistance to Replay Attacks) 、抵禦中間人攻擊 (Resistance to Man-in-the-middle Attacks) 、前向保密 (Forward Secrecy) 以及雙向認證 (Mutual Authentication) 。

4.1.1 抵禦重送攻擊

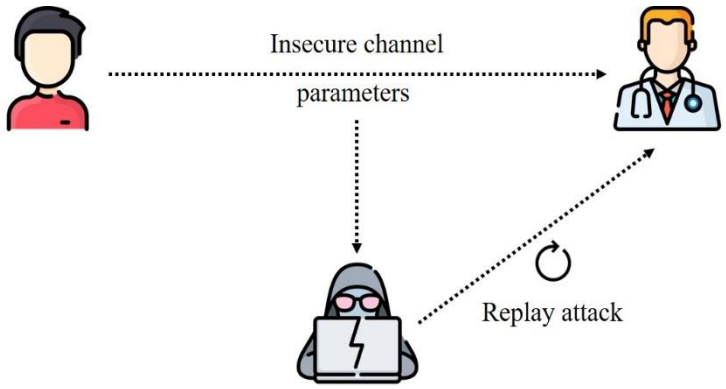

圖 18：重送攻擊

假定有一個惡意攻擊者於病患與醫院驗證期間竊取到了P_1, S_3, X_1, Y_1等參數，由於這些參數皆透過動態隨機數產生，參數僅在該次驗證時有效，下次驗證時r_1, r_2等隨機數已改變，原本竊取到的參數已無效。

4.1.2 抵禦中間人攻擊

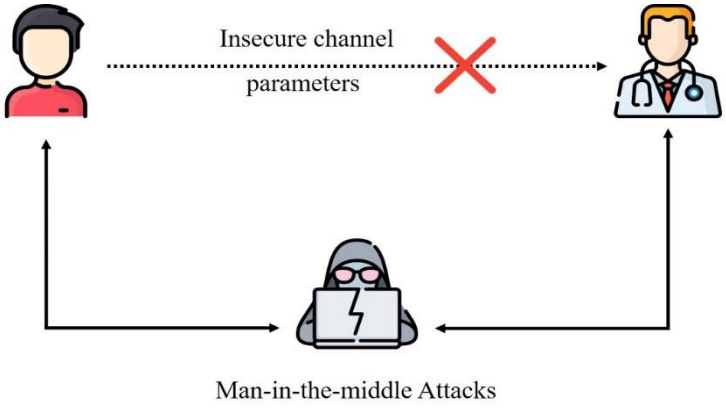

圖 19：中間人攻擊

假定有一個惡意攻擊者於病人與醫院驗證期間，想要偽造病患身分來與醫院進行驗證，因為無法得知最初的sk_p參數，以及由 PRNG 產生的r_1, r_2參數，使得無法製作X_1, Y_1, P_1, S_3等參數，因此，可以抵抗中間人攻擊。

4.1.3 前向保密

前向保密的目的是要確保在每一次輕量化驗證時，使用$session\ key$加密的資料並不會因為下一次驗證時所使用的$session\ key$，推斷出前一次的$session\ key$達到破解密文資料，本研究的$session\ key = h(r_1||r_2||sk_p||timestamp_n)$，其中$r_1, r_2, timestamp_n$等參數會隨著時間而改變，即便$session\ key$被非法取得也無法解開過去的加密資料。因此本

研究所提出的輕量化驗證滿足前向保密原則。

4.1.4 雙向驗證

在病人與醫院驗證期間，病人先傳送$X_1 = h(sk_p \oplus r_1), P_1 = h(r_1 \oplus h(ID_p)) \oplus h(sk_p)$兩個參數，在 Server 收到後，假設$sk_p$參數安全不會外流的情況下，$sk_p$參數僅有雙方知道，也就是說 Server 可以驗證X_1, P_1是否為ID_p病人所做出，驗證完成後，即完成單向驗證。接著由 Server 計算$Y_1 = h(r_1) \oplus h(sk_p \oplus r_2), S_1 = h(Y \oplus ID_p) \oplus (sk_p)$參數回傳給病人，同理$Y_1, S_1$驗證完成後，即完成雙向驗證。

4.2 效能評估

4.2.1 完整驗證效能評估

針對完整驗證中的三個階段 (金鑰產生階段、加密階段、解密階段) 。對於每個演算法，我們從屬性數量 1 個遞增到 50 個，計算各階段中所執行的時間，一共進行了 100 次取其時間平均值。在我們的觀察中發現，一位病患的屬性大約在 20~30 個之間，再加上要授權給醫生或醫院 Server 的屬性約 50 個。因此，我們在三個階段中的屬性數量最大值設定為 60。

由圖 20~22 中，可以發現其執行的時間與屬性數量呈線性成長，其中最耗時的是在加密階段中，屬性數量為 50 個花費了 2.2 秒，由此可知，倘若在 IOT 等計算能力不強的生理感測裝置相對來說會花更多時間，因此，我們提出了輕量化的驗證，使得這些裝置可以在不失安全的條件下節省更多的運算資源。

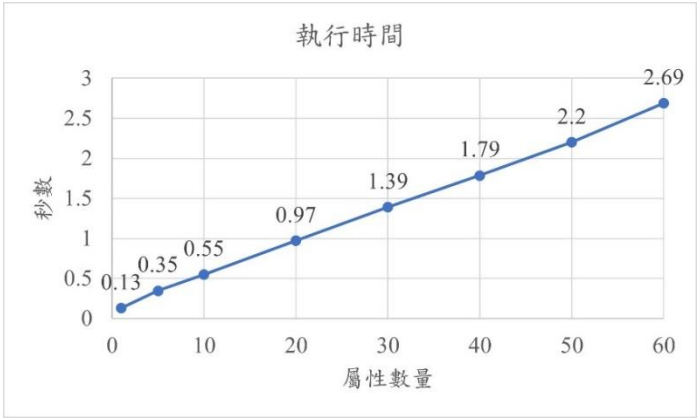

圖 20：金鑰產生階段

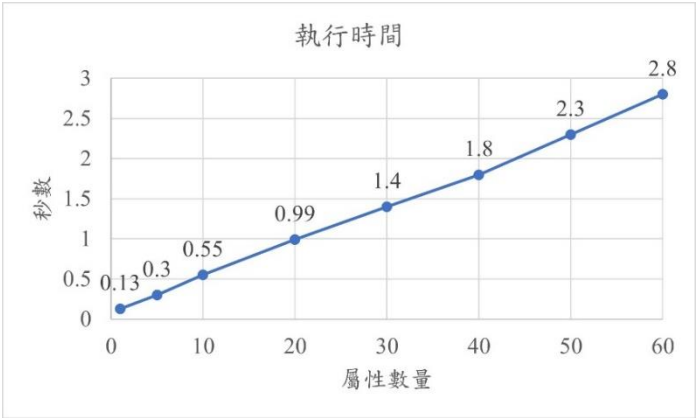

圖 21：加密階段

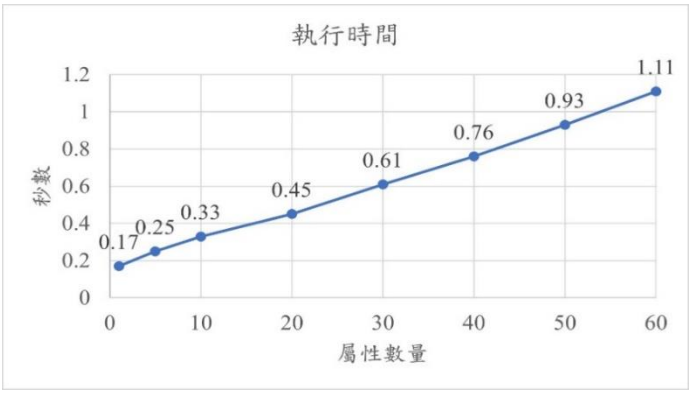

圖 22：解密階段

當我們將屬性數量固定為 30 個時，將屬性大小放至縱坐標，資料大小放至橫坐標，在我們的研究中發現，每一分鐘所蒐集到的數據及 $M = (D_{1ID_{p1}} || \dots || D_{1ID_{pi}} || timestamp_n)$，大小約為 250B，1KB 大小中的數據大約可以蒐集 3~5 分鐘，因此我們將資料大小從 1KB 遞增至 65KB。由圖 23、24 中，我們可以發現，當屬性數量固定時，其加密、解密的執行時間不易受到資料大小的影響。

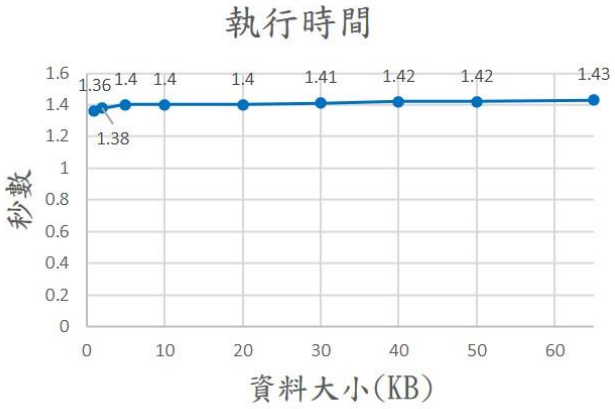

圖 23：屬性數量為 30 個加密時間

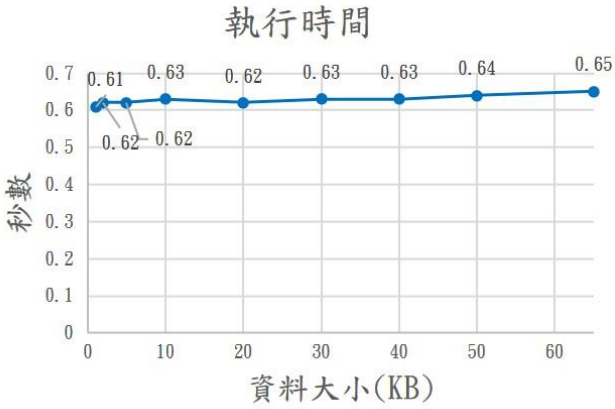

圖 24：屬性數量為 30 個解密時間

4.2.2 輕量化驗證效能評估

針對輕量化驗證中的 (病人與 Server 驗證、Server 與醫生驗證) ，這裡先忽略在傳遞參數時等待的時間，僅針對執行T_{hash}、T_{HMAC}、T_{AES}、T_{Random}的時間作為說明，一共進行了 100 次取其時間平均值。

1. T_{hash}：執行雜湊所花費的時間約為 1.5 毫秒
2. T_{HMAC}：執行一次性金鑰雜湊值所花費的時間約為 3.6 毫秒
3. T_{XOR}：執行XOR所花費的時間約為 0.9 毫秒
4. $T_{Encrption}$：使用$session\ \ key$加密所花費的時間約為 2.2 毫秒

5. T_{Random}：產生隨機數(r_1、r_2、r_3、r_4)所花費的時間約為 0.8 毫秒

在病人與 Server 驗證中、：

步驟 1~4 期間每一次皆執行了$3T_{hash} + 1T_{HMAC} + 3T_{XOR} + 1T_{Random}$，其中步驟 2 多執行了查找$sk_p$的時間，這裡先忽略不計算。故總時間約為 11.6 毫秒，與完整驗證相比，在計算參數時間花費更少，使用的參數也較少，因此在這樣的情提之下，可以改善 IOT 設備在計算能力不強劣勢。

在 Server 與醫生驗證中：

同理，與病人與 Server 驗證相同，每一次皆執行了$3T_{hash} + 1T_{HMAC} + 3T_{XOR} + 1T_{Random}$，可參考其說明。

伍、結論與未來展望

5.1 結論

本研究主要提出了一種基於屬性加密的框架下，在強調身分驗證 (Identity authentication) 與資料保密性 (Data confidentiality) 兩個目標中，我們提出了一個輕量化的雙向驗證。在完整驗證階段中，我們使用屬性加密來保障數據的隱密性、防止被授權者之間的同謀攻擊。並透過輕量化驗證讓一些運算能力較低的生理感測裝置也能應付我們所提出的參數計算。並且透過$session\ key$進行加、解密，我們的特色是使用病患的生理數據與時間戳記作為 seed，帶入 PRNG 產生隨機數，在這樣的情況下，僅有三方 (病患、醫生、醫院 Server) 知道計算隨機數，使得有心人士就算竊取到了參數，也因為不知道隨機數該如何製作而無法完成驗證。綜觀以上，我們所提出的驗證，達到了輕量化、省時、減少算力的優勢。

5.2 未來展望

本研究中所使用的屬性加密，無法支援屬性撤銷 (revoke) 的功能，也就是說，假使病患出院或是想要撤銷對於某位醫生的屬性時，只能依靠重新向屬性授權中心重新產生病患的一對金鑰 (公開金鑰、私密金鑰) 。我們提出的系統框架亦尚未支援密文搜索，目前皆仰賴醫院 Server 將收到的加密數據，解密後再儲存，使得在醫院 Server 大多都在進行加、解密，未來支援密文搜索，讓醫院 Server 工作簡單化。因此，我們可以針對屬性撤銷與密文搜索再作改進，使我們的系統框架更為完善。

參考文獻

[1] *Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, 17(4), 2347-2376.*

[2] *Mahmoud, R., Yousuf, T., Aloul, F., & Zualkernan, I. (2015, December). Internet of things (IoT) security: Current status, challenges and prospective measures. In 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST) (pp. 336-341). IEEE.*

[3] *Khemissa, H., & Tandjaoui, D. (2015, September). A Lightweight Authentication Scheme for E-health applications in the context of Internet of Things. In 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies (pp. 90-95). IEEE.*

[4] *Porambage, P., Schmitt, C., Kumar, P., Gurtov, A., & Ylianttila, M. (2014, April). Two-phase authentication protocol for wireless sensor networks in distributed IoT applications. In 2014 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 2728-2733). IEEE.*

[5] *Shamir, A. (1984, August). Identity-based cryptosystems and signature schemes. In Workshop on the theory and application of cryptographic techniques (pp. 47-53). Springer, Berlin, Heidelberg.*

[6] *Bethencourt, J., Sahai, A., & Waters, B. (2007, May). Ciphertext-policy attribute-based encryption. In 2007 IEEE symposium on security and privacy (SP'07) (pp. 321-334). IEEE.*

[7] *Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006, October). Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security (pp. 89-98).*

[8] *Sahai, A., & Waters, B. (2005, May). Fuzzy identity-based encryption. In Annual international conference on the theory and applications of cryptographic techniques (pp. 457-473). Springer, Berlin, Heidelberg.*

[9] *Guo, L., Zhang, C., Sun, J., & Fang, Y. (2013). A privacy-preserving attribute-based authentication system for mobile health networks. IEEE Transactions on Mobile Computing, 13(9), 1927-1941.*

[10] *Guo, F., Mu, Y., Susilo, W., Wong, D. S., & Varadharajan, V. (2014). CP-ABE with constant-size keys for lightweight devices. IEEE transactions on information forensics and security, 9(5), 763-771.*

[11] *Ding, S., Li, C., & Li, H. (2018). A novel efficient pairing-free CP-ABE based on elliptic*

curve cryptography for IoT. IEEE Access, 6, 27336-27345.

[12] *Hwang, Y. W., & Lee, I. Y. (2020). A Study on CP-ABE-Based Medical Data Sharing System with Key Abuse Prevention and Verifiable Outsourcing in the IoMT Environment. Sensors, 20(17), 4934.*

[13] *Zhu, H., Wang, L., Ahmad, H., & Niu, X. (2017). Key-policy attribute-based encryption with equality test in cloud computing. IEEE Access, 5, 20428-20439.*

[作者簡介] Biography

- 陳以德，2005 年取得國立交通大學資訊工程系博士，目前任教於高雄醫學大學醫務管理暨醫療資訊學系，研究興趣為 Cryptography, Watermark and Medical Informatics.
- 陳胤彤，高雄醫學大學醫務管理暨醫療資訊學系碩士生
- 蔡哲民，1998 年取得國立交通大學電子研究所博士，目前任教於崑山科技大學資訊傳播系，研究興趣為 Web Based system, Network security, and Image processing，為此篇通訊作者。