# 財團法人國際合作發展基金會
# 替代役役男專題報告

## 題目：數位身分認證系統隱私保護加密法

## 壹、 緣起

聖克里斯多福及尼維斯（以下稱為克國）政府已於2021年通過國家長期電子化政府發展藍圖，將數位身分認證系統之建置列為首要目標，並希望借助我國資通訊技術優勢及數位治理經驗，支援克國建立公民數位身分認證機制，使之成為未來各項線上服務之重要基礎設施，以推動克國數位國家與智慧政府的發展。

本專題希望通過分析比較文獻資料，選出一套適用於克國數位身份認證系統的加密演算法，對資料庫中的個人資料進行加密，並且讓資料庫中的資料在加密狀態下維持可操作性。這個加密演算法將能使計畫推行或決策更穩健執行，列舉二例如下：

一. 克國民眾使用此數位身分認證系統時，無需擔心儲存於資料庫中的個人資料在未經授權下遭他人取用，其中包含系統管理者，增加民眾對系統的隱私安全信心，有助於數位身分證推廣。

二. 使用具有可操作性的加密演算法，使系統管理者在處理資料庫資料時無需對資料做解密，加強系統運算速度，減少系統所需反應時間，增進使用者體驗。

因此，我們希望找出適用於克國數位身份認證系統的加密演算法，並統整其對系統的隱私安全性及計算速度等優點，以供日後相關人員參考及利用。

## 貳、 研究目標

基於《數位身分認證計畫》的資料及系統規格，根據用戶人數、系統架構、安全性要求、網路速度等條件，從文獻中選出可行性較高的加密演算法，並統整其能帶來的優勢。

對於目標應用於數位身分認證系統的加密演算法，初步構想首先需要具有可操作性，提升系統管理便利性以及運算速度。其次因數位身分認證個人資料中同時具有數字以及字串等不同資料型態，加密演算法需要其都能夠處理，未來資料庫中新增不同資料型態的欄位時也能應用，增加泛用性及可擴展性。最後，根據用戶人數及安全性需求，選擇能在其中取得平衡的加密演算法，使加密演算法能同時兼顧運算量與安全等級。

我們預期最終得出的加密演算法能達成先前試舉的二項應用，與數位身分認證系統結合能夠使本計畫更好地達成計畫目標，建立符合國際技術與資安標準之數位身分認證機制，提高便民服務效能，提升民眾個人資料安全性。最後以此加密演算法方案，提供給系統實作方作為工程參考。

## 參、 文獻回顧

在[1]Secure and Efficient Query Processing Technique for Encrypted Databases in Cloud這篇文章中，提到雲端計算的計算力以及儲存空間等優點，而同時面對著安全性及隱私性挑戰。本篇中提出bit vector-based model (BVM)，使資料資料庫中的資料在不洩漏額外資訊情況下具有可操作性。同時，本篇也介紹各種技術，包括已實作的系統架構，能達成加密態資料的可操作性。

在[2]CryptDB: Processing Queries on an Encrypted Database這篇文章中，展

示了一種名為CryptDB的具實用性的系統架構，可以執行大部分的SQL指令於加密態資料上。CryptDB不需要對DBMS伺服器內部做修改，並且能夠應用於大部分SQL標準DBMS。

在[3]Big Data Analytics over Encrypted Datasets with Seabed這篇文章中，展示另一種名為Seabed的系統架構，著重於在大量加密態資料中進行分析的效率。另外，有別於CryptDB使用非對稱加密技術，Seabed使用的是additively symmetric homomorphic encryption scheme (ASHE)。

在[4]Processing Analytical Queries over Encrypted Data這篇文章中，展示名為MONOMI的系統架構，主要將整個資料庫作加密並於加密態資料上運作。特點在於designer和planner的設計，以此針對不同類型資料的資料庫選出適合的加密法。

在[5]EnclaveDB: A Secure Database using SGX這篇文章中，則展示有別於上述幾種系統的加密系統架構，名為EnclaveDB。藉由將需保護的字料放入Enclave中，並置於可信任的硬體裝置中來達成資料保密性。同時，EnclaveDB透過一通訊協定確保資料不可修改性與即時性，並支援不同指令已達到可操作性。

## 肆、 研究方法

根據文獻回顧結果，得到各種加密系統已達成加密態資料的可操作性，接著會以不同系統間的泛用性，安全性，執行效率，所需成本列表分析。最後根據克國實際情況，得出克國推薦使用系統架構。

以克國現有數據來說，總人口數約為五萬上下。以系統中資料量在五萬筆的情況來說，系統運算效率要求並不高。相反，因身份認證牽涉廣大，對於安全性的要求更高，因此在對不同加密方法做比較時，安全性優先度最高。其次是泛用性。因克國原本各部門中已有現有資料系統，考慮到介接的穩定性及方便性，對於加密技術的泛用性為第二優先考量。最後才是加密技術所需運算成本，因資料量不大，即使計算成本較高，在能保證前兩者等級的情況下，依然是具有實際應用價值的方案。

以下會更詳細說明各種加密技術的運作細節以及其優缺點。

## 伍、 研究成果

本節會更詳細說明各種加密技術的運作細節以及其優缺點。

1. Bit vector-based model (BVM)

BVM的功能主要在於減少需解密的資料。在BVM中是由query manager(QM)負責接受資料庫指令、加解密等工作。當使用者端需要操作資料時QM會根據每筆資料的bit vector比對，從資料庫中將符合條件的資料調回並解密操作。因為有bit vector篩選，使得BVM不需要對全部資料進行解密一樣可以找到需要的資料的資料做操作，進而減少所需時間。

I. 加密流程與使用技術

為了安全性以及運算速度，在BVM中採用的是AES對稱加密法。在QM當中，主要有四種程序：

- Creation : 建立資料庫表格的程序。當QM收到指令要在資料庫中建立表格的時候，會先詢問表格欄位中哪些是敏感資料欄位。接著使用者需要將這些敏感欄位的值域做切分成數個子域，而QM會根據切分出來的子域建立bit vector。在bit vector中，每一bit對應一個欄位資料的子域。初始bit vector所有bit皆為0，而QM會根據每筆資料的欄位數值將bit vector對應的bit轉為1。舉例來說，若有一欄位為Name，使用者可以根據名字開頭將此欄位分成四個部分，分別是

A~F, G~L, M~R, S~Z，形成四個子域。這四個子域會由四個bit分別代表。若其中一筆資料中Name欄位資料是Alice，則A~F所代表的bit會是1，其他三個則為0。其他欄位也依此，最後形成的就是bit vector。完成bit vector後，QM會為其加上索引，作為之後搜尋使用。接著將敏感欄位資料以AES加密法做加密，最後將資料儲存於資料庫中，因此資料庫中的資料是加密態的。

- Search : 在進行資料搜尋的過程中，有三種情況。
  (1) 搜尋條件使用的欄位皆為非敏感欄位：資料庫中對應資料皆為明文狀態，QM直接根據指令從資料庫調回資料。
  (2) 搜尋條件使用的欄位皆為敏感欄位：QM根據指令從儲存的bit vector中找出符合條件的，並以索引從資料庫中調回資料。接著進行解密，找出真正符合條件的資料進行操作。
  (3) 搜尋條件使用的欄位有敏感欄位與非敏感欄位：若搜尋條件只有AND，則可以當作第一種情況，以非敏感欄位符合條件的將其調回，解密後再進行敏感欄位的比對。若搜尋條件中有OR的情況，則對敏感欄位與非敏感欄位以上述兩個情況處理。
- Update：以Search程序找出目標資料，若對敏感欄位進行改動則需解密，並在改動完後更新對應bit vector。
- Delete：以Search程序找出目標資料，進行移除。

II. 評估

- 功能性評估：BVM理論上能與所有資料庫語言相容，因它主要目的在於減少因操作資料庫資料時需要解密的資料量，所有操作會在QM中解密並執行，因此並不影響原本資料庫語言的運作。
- 安全性評估：BVM中為了安全性以及效率，選擇使用對稱式加密法中的AES加密法。AES加密法是現今對稱是加密法中安全等級很高的加密法。對稱式加密法中的密鑰傳送問題，因密鑰只會有QM保管，因此減少密鑰傳送時的風險。
- 效能評估：與沒有使用BVM的資料庫系統相比，BVM會造成一定延遲，主要原因在於加密與解密所需運算。以克國實際情況來說，資料量約在五萬筆，則增加的延遲可維持在0.5秒以內，在可以接受範圍內。而在測試中BVM可以將需要解密的資料漸少在總資料量的35%以內，相對於直接對資料加密而沒有使用BVM的資料庫系統，有顯著的提升。

2. CryptDB

為首個實際應用SQL指令於加密態資料的系統架構，並且不需要對內部資料庫管理系統(DBMS)作出修改。其他還有如Seabed、MONOMI，與CryptDB有類似的系統架構，這裡以CryptDB說明此種系統架構。

I. 威脅模組

CryptDB主要面對兩種威脅模組，分別為DBMS威脅，以及任意威脅。

DBMS威脅中包含DBMS軟體危險，管理者權限進入DBMS主機，以及主機暫存記憶體進入權限。在CryptDB系統架構中，透過架設proxy，使所有資料在進入DBMS主機前都會經由proxy進行加解密。因此DBMS主機中不會存有任何解密態資料，使得CryptDB在面對此威脅模組中依然能提供保密性。

任意威脅模組中則是攻擊者以proxy為攻擊目標的情況，攻擊者試圖透過攻擊proxy從中獲得加解密用的密鑰。解決方法是透過不同使用者的密碼產生個別使用者的密鑰，每組密鑰只對部分資料擁有權限。

II. 加密流程與使用技術

當使用者端發出資料庫指令，proxy會攔截指令並重寫它。proxy會根據

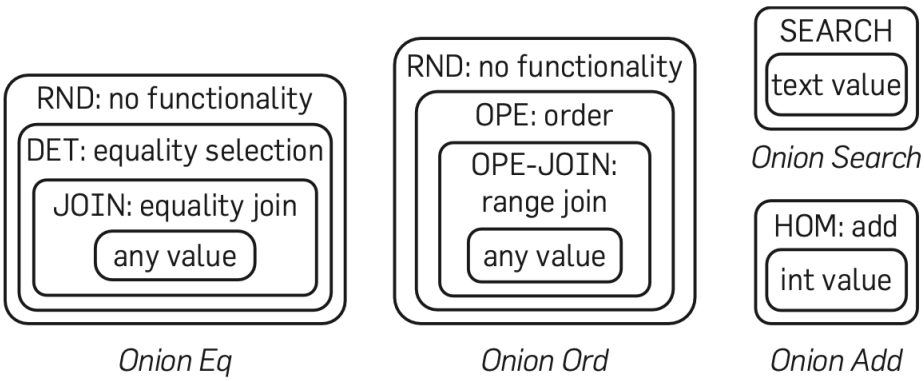
指令類型選擇適合的加密操作對其表格以及欄位名稱進行加密。接著proxy會根據指令要求，決定是否要提供相應的密鑰給DBMS，讓其對資料庫中的資料加密層級做調整。接著proxy才會將剛剛從使用者端收到的指令傳送到DBMS的伺服器上。伺服器在執行完指令後，會將加密態的執行結果傳送回proxy，由proxy解密之後將明文態資料送回使用者端。

在加密流程中使用到的技術主要有兩個，分別為SQL-aware encryption以及Adjustable query-based encryption。

SQL-aware encryption是整個加密流程中使用到的不同加密法，在資料庫中有不同類型的資料型態，因此CryptDB對不同型態的資料會採用不同的加密法，使加密態資料依然能執行SQL資料庫指令。CryptDB中使用的SQL-aware encryption有以下六種：

- Random (RND)：有最高安全性的加密，以隨機性的方式產生密文，也就是說兩個相同的資料會產生不同的密文。同時，因為隨機性，使得RND加密法加密後的資料無法進行任何操作。
- Deterministic (DET)：DET加密法對相同的資料會產生相同的密文，因此在密文狀態中可以確認兩筆資料是否相同，在SQL中支援select with, GROUPBY, COUNT等指令。
- Order-preserving Encryption (OPE)：OPE加密法加密後的密文會保留明文時的順序性，也就是說兩明文 x、y，若x<y，則OPE(x)<OPE(y)。在SQL中支援ORDER BY, MIN, MAX等指令。同時，因為暴露順序性的性質，OPE是安全性相對較弱的加密法。
- Homomorphic Encryption (HOM)：HOM是跟RND有相同安全性的加密法，而同時它具有在密文狀態下運算並解密後的結果與在明文狀態下直接運算相同的特性。而他的缺點是需要較大的運算成本及時間。在SQL中支援SUM, AVERAGE等指令。
- Join (JOIN)：JOIN加密法加密後的資料允許用不同欄位的數值進行相同比對，以達成SQL中的JOIN指令。
- Word search (SEARCH)：用來對密文狀態的資料進行搜尋，作法類似RND，但目前只能支援full word searches。

Adjustable query-based encryption的目的在於對於不同的資料在維持可操作性上使用最安全的加密法。在CryptDB是透過洋蔥式SQL-aware encryption



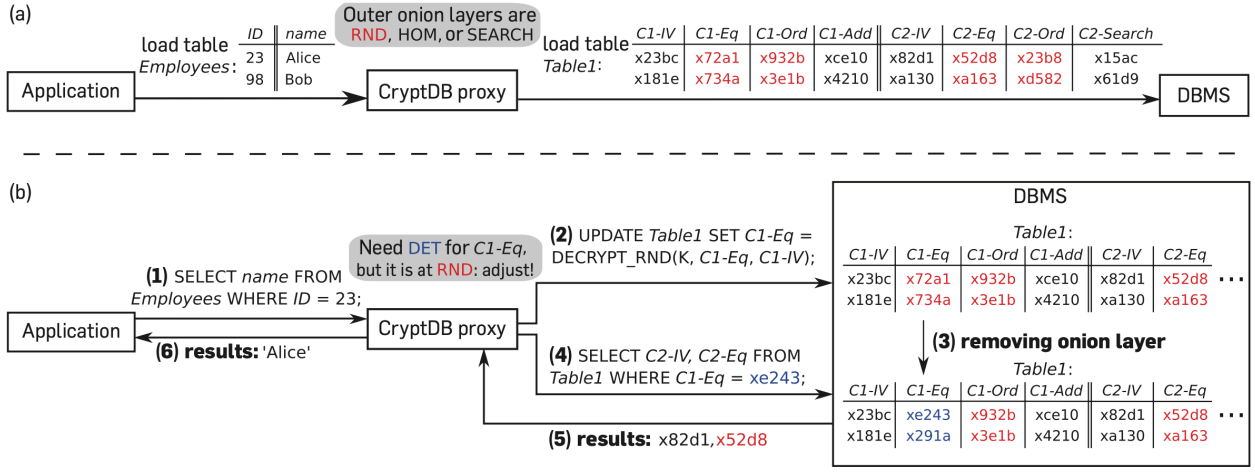

圖一、不同類型的資料加密層

來達成目的。CryptDB會對每一筆資料進行一到多次安全性越來越強的加密，每一個層級的加密法允許的操作都不相同。加密不同類型的資料會用到的加密法也都不同，例如整數型態的資料不會用SEARCH加密法進行加密。同一欄位中的資料在同一加密層級中使用的密鑰是相同的，這是因為加密態資料若要進

行操作需要在使用同一密鑰的條件下。而不同欄位之間使用的相同加密法的時候，使用的密鑰則不相同，避免過多的資訊外洩。不同類型的資料加密層如圖一所示。

每一筆資料一開始會以最高級安全性層級做加密。而當使用者端發出SQL指令時，proxy會判斷需要將資料庫中表格的資料解密到哪一層級，以進行指令操作。而需注意的是，資料庫中解密過後的加密層並不會恢復。

以CryptDB加密流程示意圖(a)為例，當使用者端將一表格傳送到資料庫管理系統前，會先經由proxy加密，而在資料庫管理系統主機中資料儲存方式就如圖(a)所示。每一欄位會經由多層加密而形成。而在CryptDB加密流程示意圖(b)中則說明當使用者端對資料庫管理系統下達SQL指令，CryptDB會進行的操作流程。在此例中，使用者端下達指令SELECT name from Employees WHERE ID = 23，而ID欄位的加密層目前處於RND，無法進行比對操作，因此proxy會將密鑰傳送給資料庫管理系統，將ID欄位的RND加密層解開，此時ID欄位的加密層為DET，可以進行比對操作，這時proxy才把SQL指令傳送到資料庫管理系統，在加密態資料上進行SQL操作，並將加密態的執行結果傳送回proxy，由proxy將其解密後回傳給使用者端。




CryptDB加密流程示意圖

III. 評估

- 功能性評估：CryptDB在與不同SQL系統進行測試後，得到能與大部分SQL指令相容的結果。與CryptDB不相容的指令主要有兩種，一種是字串處理相關指令，一種是日期處理相關指令。不相容的指令在所有測試指令中佔比0.44%，對於資料庫操作影響不大。
- 安全性評估：在所有加密層中，安全度等級最高是RND，最低是OPE，而在資料庫中，預設的加密層為RND，只有在進操作後才會調整加密層。在測試結果中，約有6.6%欄位資料最後位於OPE加密層，27.4%欄位資料位於DET加密層，65.5%位於RND加密層。因此CryptDB的資料安全性具有一定保障。
- 效能評估：在流通量上，使用CryptDB的SQL系統約比未使用CryptDB的SQL系統減少26%。主要減低流通量的因素為涉及HOM加密層操作的指令如SUM，因HOM加密法特性需要較大運算成本與時間。在延遲上，CryptDB則會產生約0.12ms的延遲。以克國實際狀況來說，因資料量不大，此流通量影響對於系統運作不會產生過多影響。

3. EnclaveDB

EnclaveDB是一資料庫系統，用來保護資料的保密性，不可修改性，即時性等。做法是將表格、索引等敏感資料儲存於enclave中，並以可信任硬體裝置保護。另外，EnclaveDB中以一特殊協議對資料紀錄進行檢查，以保證資料的不可修改性及即時性。在上述保護下，EnclaveDB得以抵抗多種威脅，如惡意資料庫管理者，對作業系統的攻擊等等。

enclave是一可信任執行環境，會在系統中獨立出一塊虛擬空間，其中的程式碼以及資料受到保護，不會被系統中其他軟體攻擊。當需要emclave中的資料或服務時，會由原本的執行緒轉換成enclave-mode的執行緒，以執行user-mode的程序。除了獨立保護，enclave同時也提供印信以及遠端驗證。印信透過密鑰對資料的保密性以及不可修改性做出保證。遠端驗證則提供與enclave建立互信，以架設安全頻道並傳送機密資料。

### I. 威脅模組

在威脅模組中我們考慮要面對的威脅是攻擊者能控制除了enclaves中程式碼以外的資料庫系統中的軟體堆疊。這就代表攻擊的攻擊目標包含作業系統、虛擬機管理程式或資料庫伺服器。可能的攻擊方式包含竄改資料庫檔案中的記錄、發起replay attack、監測或更改指令控制流程等等。

在面對以上威脅中，攻擊者是無法打開封包以及enclaves中的程式碼的。目前研究證明上述要求是有辦法達成的[6]。同時我們假定使用者端也是可信任的。在這樣的條件下，EnclaveDB的目標是保證資料保密性。

### II. 加密流程與使用技術

EnclaveDB由數個元件組成，以下一一介紹：

- 可信任核心：一般的資料庫常會使用作業系統中提供的服務，如執行緒，記憶體管理等等。然而這種方法在作業系統可能被攻擊的情況下並不安全。因此需要使用在enclave中的作業系統。可信任核心的目的就是為了取得emclave中作業系統所提供的服務，包含enclave中的記憶體及執行緒管理等等。
- 執行程序：在收到資料庫指令時，系統會創建一個執行程序，並指派一個起始timestamp給他。接著執行程序中的儲存步驟，載入相應的compiled binary，然後把控制權交給編譯碼中的函式。compiled binary會負責呼叫資料庫引擎，在資料庫中的表格執行操作，並更新執行程序狀態。整個儲存步驟都會在enclave中完成，以預防資料庫管理者在指令執行過程中對資料進行竄改。在完成儲存步驟後，接著進行commit，檢查是否發生衝突，並指派結束timestamp給執行程序，並等待執行程序完成紀錄。在上述流程中，由於timestamp以及紀錄，可以預防來自惡意管理員的攻擊手法。同時為了資料不可修改性，針對使用者端會進行一些改動。例如使用者端與資料庫系統透過enclave建立安全通道，並共享session key，session key用來加密參數，以預防重複攻擊。除了使用者端，伺服器端也需要相應調整。執行程序的驗證程式會放在eclave中保護，所有進入資料庫的指令都會經過驗證，確認與資料庫中的compiled binary中的參數都符合才會執行後續操作。
- 密鑰管理：EnclaveDB中的密鑰管理與其他系統相較之下較為簡單，由使用者管理每一加密欄位的密鑰。加密欄位會儲存在enclave中，而這些欄位中的資料會由記憶體加密引擎做加密，以保證資料完整性。此外，使用者需要管理一密鑰，用來對資料庫狀態進行加密，並將密鑰儲存於密鑰管理裝置。若發生錯誤需要回朔資料庫狀態，會使用此密鑰進行驗證。

- 記錄檔：在EnclaveDB中認為host是不可信任的，因此需要保證惡意管理者對資料紀錄不能查閱或修改。所用的方法是將資料記錄進行連載，

每筆資料記錄會包含上述執行程序中產生的起始及結束timestamp。而為了避免資料記錄無限制成長，會週期性的產生檢查點並修剪資料紀錄。為了資料記錄完整性，enclaveDB使用特別的通訊協定以檢查資料紀錄的完整性與最新性。協定中包含三個counter，分別追蹤已經寫入的資料紀錄、追蹤可見執行程序產生的資料紀錄、以及追蹤修剪的資料紀錄。於[5]中詳細證明資料紀錄配合通訊協定保證了資料的完整性、連續性及最新性。

III. 評估

- 功能性評估：EnclaveDB並未對原本資料庫語言作出改動，理論上與不同資料庫可以相容。然而在功能性上依然有可以改善之處，如動態改動權限使用者等等。
- 安全性評估：EnclaveDB的安全性很大程度取決於enclave的安全程度。因在enclave中，應用程式運行環境是完全獨立於host，使用的記憶體空間也是完全獨立出來的硬體，因而可以很大程度保證其中的密鑰保密度。而在enclave與其他元件運行時，會經由驗證程序確保資料來雅與安全性。
- 效能評估：根據測試的結果，使用EnclaveDB會造成記憶體以及硬碟頻寬效能比未使用的情況較低，原因在於加密時所產生的延遲。於不同標準下測試產生不同結果，產生的效能降低率在15%~40%，以克國資料量不多的情況下雖不至於對系統運行造成影響，但若在實際應用上可以預期成為使用者負擔。

## 陸、 討論及建議

經過研究成果的資料整理，彙整出三種資料加密法，分別是Bit vector-based model (BVM)、CryptDB、EnclaveDB。這三種加密法中，安全性CryptDB與EnclaveDB相近，而BVM較差，原因是BVM是減少運算時所需解密的資料，依然有部分資料解密是執行指令不需要的。相較之下，CryptDB會在資料庫中的加密態資料上完成操作，只回傳執行結果，而EnclaveDB則是在獨立記憶體中完成執行過程，兩者透露出的資訊比BVM要少，因此判斷BVM在安全性上表現較差。在泛用性方面，三者的差距不大，唯CryptDB在少部分資料庫指令上可能遇到困難，但如上述研究成果，無法執行的指令佔比極少，對執行過程影響可以忽略。效率上來說，CryptDB效率最高，其次是BVM，EnclaveDB則表現較前兩者差。由以上幾點分析，結合克國實際狀況，我會推薦使用CryptDB，安全性有足夠的保證，不會輕易洩露資料庫資料，無論是系統攻擊者或是惡意資料庫管理員。而在保證安全性的情形下，CryptDB依然能擁有優秀的效率表現，讓其在使用上不會造成負擔。

最後進行總結，現今環境中資料安全議題愈趨重要，除了防範系統攻擊者，資料庫管理者的行為也須受到重視。在聖克里斯多福及尼維斯數位身分認證計畫中，個人隱私資料的安全性十分重要，為保護克國民眾的資訊安全，增加民眾對系統的隱私安全信心，協助推廣計畫執行，本專題通過分析比較文獻資料，選出一套適用於克國數位身份認證系統的加密演算法，對資料庫中的個人資料進行加密，並且讓資料庫中的資料在加密狀態下維持可操作性，使資料能取得安全性與效率之間的平衡。研究成果中介紹三種不同的資料加密系統，以不同方式對資料進行保護。經分析比較優缺，結合克國實際情況，最後得出結論於克國推薦使用CryptDB，使資料於系統以加密態儲存，同時具有可操作性，付出可負擔的運算成本的同時，增加資料安全性。以上是本專題研究成果，供日後相關人員參考及利用，以期為聖克里斯多福及尼維斯數位身分認證計畫提供幫助。

# 柒、 附件

[1]Sultan Almakdi, Brajendra Panda, “Secure and Efficient Query Processing Technique for Encrypted Databases in Cloud ”, 2019 2nd International Conference on Data Intelligence and Security (ICDIS)

[2]Raluca Ada popa, Catherine M.S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan, “CryptDB: Processing Queries on an Encrypted Database”, Computer science and artificial Intelligence lab, M.I.t., Cambridge, Ma.

[3]Antonis Papadimitriou1†, Ranjita Bhagwan∗, Nishanth Chandran∗, Ramachandran Ramjee ∗, Andreas Haeberlen†, Harmeet Singh∗, Abhishek Modi∗, Saikrishna Badrinarayanan1‡, “Big Data Analytics over Encrypted Datasets with Seabed”, University of Pennsylvania, ∗Microsoft Research India, ‡UCLA

[4]Stephen Tu, M. Frans Kaashoek, Samuel Madden, Nickolai Zeldovich, “Processing Analytical Queries over Encrypted Data”, MIT CSAIL

[5]Christian Priebe, Kapil Vaswani, Manuel Costa, “EnclaveDB: A Secure Database using SGX”

[6]R. Sinha, M. Costa, A. Lal, N. P. Lopes, S. Rajamani, S. A. Seshia, and K. Vaswani, “A Design and Verification Methodology for Secure Iso- lated Regions,” in *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI ’16, 2016.

[7]https://www.anjuna.io/resources/what-is-a-secure-enclave