

國立臺灣大學電資工程學院資訊工程所

碩士論文

Department of Computer Science and Information Technology

College of Engineering

National Taiwan University

Master Thesis

基於 OurChain 的自主身分系統設計與實作

Design and Implementation of Autonomous Identity
System Based on OurChain

林俊佑

Jun You, Lin

指導教授：薛智文 博士

Advisor: Chih-Wen (Steven) Hsueh Ph.D.

中華民國 113 年 7 月

July, 2024

國立臺灣大學碩士學位論文

口試委員會審定書



基於 OurChain 的自主身分系統設計與實作

Design and Implementation of Autonomous Identity
System Based on OurChain

本論文係林俊佑君（R11922114）在國立臺灣大學資訊工程所完成之碩士學位論文，於民國 113 年 7 月 25 日承下列考試委員審查通過及口試及格，特此證明

口試委員：_____

（指導教授）

_____	_____
_____	_____
_____	_____
_____	_____

所 長：_____



[illegible]





摘要

隨著網路科技的快速發展，數位身分的管理與認證變得愈加重要。在傳統的身分管理系統中，中心化機構負責儲存與管理用戶的身分資料，但這樣的做法存在諸多安全風險，如資料洩漏和身分盜用。問題的根源在於中心化身分管理系統最終依賴於各個機構的自律性。當機構辜負用戶的信任，濫用資料謀取私利，或因資訊安全疏忽導致資料洩漏時，用戶往往無法即時得知，只能被動地接受損失。此時，所謂信任顯得如此蒼白無力！

我們針對身分管理系統中存在的安全風險和信任問題進行深入研究，並提出了一種全新的解決方案——自主身分識別。傳統的身分管理系統雖然已經足夠實用，但其對中心化中介者的依賴造就了更高的安全風險與更低的可信任性。本研究的核心概念是將身分識別的控制權交還給用戶，使其能夠主動地管理自己的身分資訊，從而提升系統的安全性與可信任性。

然而單純的透過分散式系統來完成身份認證無法解決多節點的信任性問題，所以我們基於 OurChain 實作了基於智能合約的解決方案，讓 OurChain「一個去中心化的區塊鏈」得以提供一個可信任的執行環境作為「自主身分識別」系統的運作基礎。

關鍵字：自主身分、管理、認證、隱私、區塊鏈





Abstract

Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Ab-
stract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract Abstract
Abstract Abstract Abstract Abstract

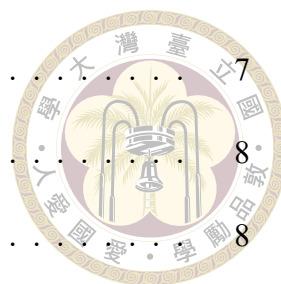
Keywords: Autonomous Identity, Management, Authentication, Privacy, Blockchain



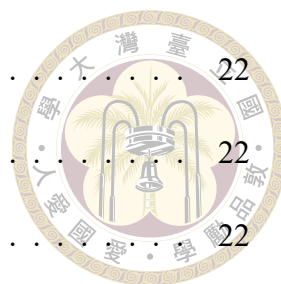


目錄

	Page
口試委員審定書	i
致謝	iii
摘要	v
Abstract	vii
目錄	ix
圖目錄	xiii
表目錄	xv
符號列表	xvii
第一章 緒論	1
1.1 研究背景	1
1.2 研究目的	2
1.3 研究意義	2
1.4 論文架構	2
第二章 文獻探討	5
2.1 身份系統的想像	5
2.2 身份系統的迭代	7
2.2.1 中心化身份	7

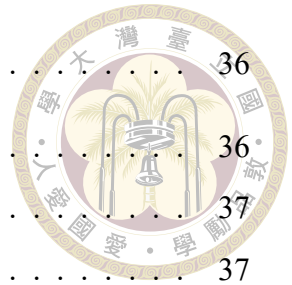


2.2.2	聯合身份	7
2.2.3	使用者中心的身份	8
2.2.4	去中心化身份	8
2.2.5	未來展望	9
2.3	自主身份的設計原則	9
2.3.1	用戶體驗	9
2.3.2	用戶認知	11
2.3.3	隱私保護	12
2.3.4	平等信任	13
2.4	自主身份系統的評估	14
2.5	總結	15
第三章 系統設計		17
3.1	系統的重要進步	17
3.1.1	別名優先機制	17
3.1.2	基於用戶時空的分析方法	18
3.1.3	基於危險程度的驗證機制	18
3.1.4	極致多因素認證	19
3.1.5	半匿名制	20
3.1.6	基於區塊鏈的共識機制	21
3.2	系統架構	22
3.2.1	整體說明	22
3.2.2	AID Server	22
3.2.3	Wallet	22



3.2.4	Consensus Core	22
3.3	資料結構	22
3.3.1	整體說明	22
3.3.2	AID Server	22
3.3.3	Wallet	22
3.3.4	Consensus Core	22
3.4	AID 的威脅模型	22
3.5	本章小結	22
第四章	系統實作	23
4.1	各模塊說明	23
4.1.1	Wallet	24
4.1.2	AID Server	26
4.1.3	Consensus Core	27
4.2	重要流程分析	29
4.2.1	AID 生成	29
4.2.1.1	用戶自主生成 AID	29
4.2.1.2	服務提供商生成 AID	29
4.2.1.3	AID 生成的通用原則	30
4.2.2	AID 中基於 RBA 的 MFA 驗證	30
4.2.3	別名在服務器上註冊	31
4.2.4	別名在服務器上登入	33
4.2.5	AID 的批次創建與綁定	34
4.2.6	AID 遷移	35

4.3	與現有方法之比較	36
4.3.1	AID 系統對 GDPR 合規性的創新解決方案	36
4.3.1.1	數據最小化和存儲限制	37
4.3.1.2	被遺忘權的實現	37
4.3.2	AID 系統的身份冒用防護與救援機制	38
4.3.2.1	身份冒用防護機制	38
4.3.2.2	創新的身份救援機制	39
4.3.2.3	安全性與可用性的平衡	40
第五章	結果與未來展望	43
5.1	結論	43
5.2	未來展望	43
	參考文獻	45
	附錄 A — 模型參數表	51
A.1	模型一	51
A.2	模型二	51
	附錄 B — BraTS 2021 分割結果圖	53
B.1	編號 0001 0050	53
B.2	編號 0051 0100	53





圖目錄





表目錄

4.1 系統各模塊支持的核心操作	23
------------------------	----





符號列表

\mathfrak{a}	\mathfrak{a} 符號解釋
\int	\int 符號解釋
v	v 符號解釋





第一章 緒論

本研究探討了將自主概念應用於數位身份管理的創新方法，提出「自主身份」(Autonomous Identity, AID) 系統，通過為用戶提供對其數位身份的全面控制權，以解決當前身份管理模式的固有限制。

1.1 研究背景

自主 (Autonomy) 的概念源於 18 世紀啟蒙時代，當時人類開始質疑對王權與神權的依賴，轉而追求通過科學與理性達成思想上的獨立與自由。啟蒙思想家伊曼努爾·康德 (Immanuel Kant) 對自主提出了開創性的定義，即「按照自己的法則來行事」。這一定義至今仍廣泛應用於政治、法律、教育等諸多領域。

康德的自主概念強調個體應基於自身的道德觀制定規則，並主動依循這些規則與他人互動。本研究旨在將這一深具哲學意涵的概念引入當代數位身份系統的設計中，以解決現有系統中的諸多困境。

我們提出「自主身份」(Autonomous Identity, AID) 的概念，作為實踐康德自主理念的一種方法。自主身份系統旨在確保每位用戶能夠「成為自己的主人」，即使在虛擬的網絡環境中，也不會因數位身份的特性而喪失自主權。本文將探討自主身份系統的理論基礎、實施方法及其對現有數位身份管理體系的潛在影響。



1.2 研究目的

本研究旨在探討自主身份 (Autonomous Identity, AID) 作為解決當前身份管理問題的潛在方案。具體而言，本文將：

1. 分析當前數位身份系統的主要挑戰和限制。
2. 評估自主身份系統對比傳統身份管理系統的優勢。
3. 提出自主身份系統的核心概念和技術基礎。
4. 討論自主身份系統在實際應用中可能面臨的挑戰和解決方案。

1.3 研究意義

自主身份系統代表了身份系統領域的一個重要典範轉移。過去的身份管理系統主要由機構控制，每個用戶被由上而下的方式賦予一個身份。而自主身份系統則轉向由用戶自己控制自己的身份，由下而上的組織成群，形成一個又一個的身份互動節點，最終透過去中心化的方式使每個節點針對想要共享的資訊取得共識。

通過賦予用戶更多對自身數位身份的自主權，自主身份系統有潛力改變個人、組織和社會之間的互動方式。本研究的結果將為政策制定者、技術開發者和最終用戶提供寶貴的見解，有助於推動更進步的身份管理創新。

1.4 論文架構

為了要基於自主的理念設計出一個完整的身分驗證系統，我們需要先了解現有的身分驗證技術，並且對於這些技術進行分析，找出其缺點 (chapter2)。接著提

出我們的系統設計，並且說明其架構、資料結構與威脅模型 (chapter3)。最後，我們會透過實作來驗證我們的系統設計 (chapter4)。







第二章 文獻探討

身份系統的發展經歷了從中心化到去中心化的演變，現代身份系統設計面臨著用戶體驗、認知負擔、隱私保護和平等信任等多方面的挑戰，需要同時滿足技術創新、法規遵循和行業最佳實踐，以創造出既能滿足多樣化需求又具備足夠靈活性的解決方案，本章將探討相關文獻以深入理解這些挑戰及可能的解決途徑。

2.1 身份系統的想像

Internet 發源於 20 世紀 70 年代，逐漸發展成為現代網際網路的基礎架構。最初，Internet 的設計源自美國國防部的 ARPANET 計畫，旨在滿足軍事需求下的網路可用性和穩定性。因此，其初期設計基於以下假設 [27]：

- 最終使用者至少在最低程度上相互信任。
- 網路由於潛在的物理攻擊而本質上不可靠。

這些假設在當時的網路環境中是合理的。然而，隨著網路的普及和應用範圍的擴大，這些假設已不再適用 [9]。Internet 從一個研究驅動的项目演變為主流社會的重要組成部分，新的需求不斷湧現，不僅挑戰了原有的設計原則，還促使我們重新審視一些既有的原則。

在現代網路環境中，使用者之間的信任關係變得日益複雜。人們需要可靠的



方式來識別自己和他人，以便在網路上進行交流和交易。因此，各種身份系統應運而生，以滿足網路環境中的身份識別需求。儘管人們希望找到一種統一而理想的身份管理系統，但正如 Cameron[7] 所指出的，這樣的系統實際上並不存在。身份系統涉及的範疇廣泛而複雜，個人與組織之間存在多樣化且往往相互衝突的需求，試圖通過單一標準來限制或規範這些需求是不切實際的。

例如，終端用戶可能希望自由地訪問和分享信息，而內容提供商和知識產權持有者則希望保護其知識產權。政府可能希望監管某些網絡活動以維護社會秩序，而用戶和隱私倡導者則強調個人隱私的重要性。這種複雜的利益衝突導致了諸如網絡中立性、數據隱私、內容審查等一系列熱點問題的出現 [38]。

面對這種複雜的網絡環境，Blumenthal[5] 認為確保設計上的一般性、彈性與開放性至關重要。他設想的未來充滿衝突：企業的管理者和被管理者爭論分紅，垃圾郵件的發送者和接收者爭論各自的困難。在這個網路世界中，不同身份的參與者沒有絕對的贏家，也沒有天生的失敗者。理想的系統應該通過所有用戶不斷的爭論和互動逐漸形成。

這種新的設計思路不僅需要考慮技術因素，還需要權衡經濟、社會、法律等多方面的因素。正如 Lessig[22] 所言，”程式碼就是法則”（”Code is Law”）。軟體設計本身就在表達某種價值觀，人們需要縝密的思考，結合對未來世界的想像，才能創造出足以改變網路環境既有困境的身份系統。

總的來說，現代網路環境的變遷對身份系統提出了新的挑戰，這些挑戰不僅來自技術層面，還涉及社會、文化和法律等多個方面。為了應對這些挑戰，我們需要重新思考身份系統的設計理念，尋找一種既能適應當前多樣化需求，又能夠為未來發展預留足夠靈活性的解決方案。



2.2 身份系統的迭代

身份系統的設計經歷了多個階段的演變，每個階段都試圖解決特定的問題，同時也帶來了新的挑戰。本節將介紹不同世代的身分系統設計，以說明彼此衝突的需求和技術限制，並為後續討論提供背景。

2.2.1 中心化身份

中心化身份系統是最早期的身份管理方案，在企業和政府機構中廣泛應用。典型例子包括 Windows Active Directory 和 LDAP（輕量級目錄訪問協議）[25, 33]。這類系統的主要優勢在於其集中管理的特性，便於系統管理員進行用戶管理和權限控制，同時確保組織內部身份信息的一致性和及時更新。然而，中心化身份系統也面臨著諸多挑戰，如單點故障風險、隱私保護問題，以及跨組織可遷移性差等。用戶通常需要為每個服務創建單獨帳戶，這不僅增加了認知負擔[19]，還導致用戶身份被服務提供商完全控制，缺乏自主權。

2.2.2 聯合身份

主要為了解決同一個用戶擁有太多身份的認知負擔，聯合身份系統允許不同組織間共享身份信息，代表性技術包括 SAML（安全斷言標記語言）和 WS-Federation [14, 28]。這種模式的出現大大改善了用戶體驗，實現了單點登錄（SSO），減少了密碼疲勞問題。聯合身份系統促進了組織間的協作和資源共享，同時也降低了重複身份管理的運營成本。然而，這種模式也帶來了隱私方面的挑戰[2]，如用戶信息在多個服務提供商間共享可能違反 GDPR 等隱私法規[11]。此外，實施和維護聯合身份系統的技術複雜度較高，參與組織之間需要建立並維護

信任關係。



2.2.3 使用者中心的身份

為了解決隱私方面的問題，使用者中心的身份系統逐漸興起。OpenID 和 OAuth 等協議的出現 [17, 31]，標誌著身份管理向用戶賦權的重要轉變。這種模式增強了用戶對個人身份信息的控制權，提供了更大的靈活性，允許用戶選擇不同的身份提供者。使用者中心的身份系統實現了以服務供應商為單位的資訊範圍控制，在一定程度上改善了隱私保護。然而，這種方式也面臨著身份碎片化的問題，多個身份提供者的存在可能導致用戶體驗的不一致。安全風險如釣魚攻擊和身份提供者數據洩露等仍然存在 [36]。此外，儘管用戶獲得了更多控制權，但他們仍然在某種程度上依賴中心化的身份提供者，因此用戶的自主性還擁有很大的提升空間 [4]。

2.2.4 去中心化身份

去中心化身份是身份系統概念的最新發展 [29]，其代表性例子包括基於區塊鏈的 uPort 和微軟的 ION（去中心化身份覆蓋網絡）[23, 24]。這種方式賦予用戶對自身身份和個人數據的完全控制權，同時提高了身份的可攜性和一致性。通過利用區塊鏈技術，去中心化身份系統不僅增強了用戶與服務提供商之間的平等性，還提供了抗審查的特性。然而，作為一種新興技術，去中心化身份系統也面臨著諸多挑戰 [32]。首先，它與現有法律框架可能存在潛在衝突，例如與 GDPR 中的被遺忘權不相容 [12]。其次，去中心化身份系統的技術複雜性可能影響普通用戶的使用體驗，降低其易用性 [20]。此外，還有諸如隱私保護、系統互操作性等問題需要解決。



2.2.5 未來展望

綜上所述，身份系統的發展經歷了多個階段，每個階段都試圖解決特定的問題。從中心化到去中心化，身份系統的設計逐漸向用戶賦權，提高了用戶對自身身份的控制權。然而，即使到了去中心化階段，我們依舊不認為找到了理想的解決方案。如同 Schardong 等人 [32, 35] 所說，當今的去中心化身份系統仍面臨著許多挑戰，包括安全性、隱私保護、易用性、信任建立等問題。因此，我們認為身份系統的設計仍有很大的改進空間，需要更多的研究和實踐來不斷優化。

2.3 自主身份的設計原則

為了設計出一個理想的身份系統，在本節中，我們會從不同維度的多個方向來探討身份系統的設計原則。這些原則旨在幫助我們理解身份系統的核心特徵，並為未來的設計提供參考。

2.3.1 用戶體驗

用戶體驗在身份系統設計中扮演著關鍵角色，直接影響系統的可用性和採納率。然而，Hamme 等人 [16] 的研究闡明了用戶體驗、安全性和隱私保護之間的複雜關係。該研究指出了一個普遍存在的現象：用戶傾向於選擇最簡單的方式來設置和使用身份系統，這種傾向可能導致系統安全性和隱私保護程度的降低。

這種情況產生了一個兩難困境，為了提高安全性而強制用戶採用複雜的身份驗證方式可能會適得其反。例如 Zhang 等人 [39] 的研究表明，要求用戶定期更改密碼往往導致用戶僅修改特定字元，反而造成更大的安全隱患。同樣地，為了增強隱私保護而要求用戶完成詳細的隱私設置也可能降低用戶體驗。Acquisti 等人



[1] 的研究發現，複雜的隱私設置過程往往使用戶感到困惑和沮喪，甚至導致他們放棄設置而選擇默認選項，從而降低了隱私保護水平。

為解決這一困境，研究者提出了「無摩擦驗證」(Zero Friction Authentication) 的概念，旨在最小化用戶在設置和使用過程中遇到的困難，同時維持適當的安全性和隱私保護水平。Hamme 等人 [16] 強調，無摩擦驗證的核心目標是在保護用戶安全和隱私的同時，顯著降低用戶的操作負擔。這種平衡對於現代身份系統的設計至關重要，因為它直接影響系統的使用率和效能。

為實現無摩擦驗證，近年來安全領域出現了多種新技術。如 Ghorbani 等人 [13] 研究了無密碼登入（如 FIDO2）的可用性，發現這種方法能透過硬體密鑰在手機上跨裝置完成高安全性的驗證，且使用者普遍認為方便並願意持續使用。Wiefling 等人 [37] 探討了基於風險的驗證（RBA），該方法透過追蹤身份與系統互動的歷史數據，在每次服務請求時動態判斷危險性，並在危險時採用更安全的多因素驗證（Multi-Factor Authentication, MFA）[6]。Alaca 等人 [3] 關於裝置指紋的研究顯示，透過在每次用戶對服務發出請求時記錄並比對裝置指紋，可以有效辨識部分惡意行為，且不會增加用戶的操作負擔。

另外，為解決複雜隱私設置帶來的問題，Acquisti 等人 [1] 提出了「隱私設計」(Privacy by Design) 的概念。這種方法將複雜的設置過程分解為多個簡單步驟，並在用戶使用系統的不同階段逐步引導用戶完成設置。研究表明，這種方法不僅能提高用戶的隱私保護水平，還能顯著改善用戶體驗。

這些新技術的應用表明，在不影響用戶體驗的前提下提供更高的安全性和更好的隱私保護是可能的。然而，如何在自主身份系統中實現真正的無摩擦驗證，以及如何有效地平衡用戶體驗、安全性和隱私保護的需求，仍然是一個值得深入研究的課題。



2.3.2 用戶認知

用戶認知在數位身份管理中扮演著至關重要的角色，直接影響到系統的安全性和有效性。LastPass[21] 的研究揭示了用戶認知與實際情況之間存在顯著差距：用戶平均估計自己擁有 20 個線上帳號，而實際上平均擁有 37 個以上的帳號。這種認知偏差背後反映了用戶使用上的不便，例如經常因為忘記密碼而無法登入帳號，或者因為各個帳號資料不互通而需要耗費大量時間來管理。

Dhamija 等人 [10] 的研究進一步指出，用戶對身份管理系統的認知和理解程度直接影響其安全行為。隨著需要管理的用戶名和密碼數量增加，用戶往往感到困惑，進而採取不安全的行為，如使用弱密碼或在多個平台使用相同密碼等。此外，用戶對身份管理系統的認知不足也會導致他們無法有效應對釣魚攻擊、社交工程等安全威脅。

為了解決這個問題，未來的身份管理解決方案應該朝多個方向發展。首要任務是簡化多層次、多維度的用戶身份管理，允許單一的身份管理多樣的別名，以適用於不同的場景。例如，用戶可以用唯一的帳戶創建三個別名，分別對應自己的三種社會身份：在家中是家長，在工作中是員工，在社交場合是朋友。甚至針對單一的服務，用戶也可以擁有多種別名，如在論壇中既可以以專家身份發表權威言論，也可以作為普通用戶表達個人觀點。這樣的設計可以幫助用戶在盡可能不增加認知負擔的情況下有效管理自己的身份。

然而，真正簡化用戶認知並非易事。即使是宣稱已解決這個問題的使用者中心身份系統，實際上也未能完全做到。以 Google 的組織管理文件 [15] 為例，為了確保不同組織擁有不同的安全限制與規定，系統仍然要求用戶在不同組織間創建不同的身份。這表明，同時簡化用戶認知與滿足組織需求，仍然是一個有待解決的挑戰。



2.3.3 隱私保護

在數位時代，隱私保護已成為身份系統設計的核心考量之一。歐盟制定的《通用數據保護條例》(GDPR) [11] 代表了目前全球最嚴格的隱私保護標準。本研究認為，一個理想的身份系統應當能夠全面符合 GDPR 的要求，從而確保用戶隱私得到最大程度的保護。然而，近年來的 GDPR 違規案例表明，即便是大型企業也面臨著遵守某些 GDPR 規定的挑戰。

基於 Schardong 等人的研究 [32]，我們發現當前身份系統中存在兩個尤為突出的關鍵問題。首先是用戶積極授權的實現困難。Saemann[30] 的研究強調，在當前的身份系統框架下，企業難以實現用戶對數據使用的明確授權。具體而言，企業難以證明其對數據或權限的使用行為已獲得用戶授權，而用戶也缺乏有效途徑證明自己的數據或權限被不當使用。這種情況不僅增加了企業的法律風險，也削弱了用戶對系統的信任。

第二個問題是被遺忘權的實現困難。Smirnova[34] 指出，滿足用戶的被遺忘權在當前身份系統中存在著一定的挑戰。用戶數據在系統中往往呈分散狀態，即使刪除核心用戶的資料，仍可能保留用戶的系統日誌或與其他用戶的互動數據。這種情況使得完全實現被遺忘權變得複雜而困難，可能導致用戶隱私無法得到全面保護。

基於以上分析，我們提出符合現代隱私保護要求的身份系統應該具備兩個關鍵特點。首先，系統應提供合理的機制，使用戶或企業能夠證明其數據使用行為是否符合授權，這將有助於提高系統的透明度和可信度。其次，系統應提供有效的方法讓用戶行使被遺忘權，確保用戶不會被難以刪除的數據綁架。這意味著系統需要設計更精細的數據管理和刪除機制。在後續研究中，我們將詳細探討如何在自主身份系統中實現這些特性，並提出相應的技術解決方案。



2.3.4 平等信任

身份系統中的平等信任問題是一個複雜的多方利益平衡問題，涉及系統的公平性和可信度。研究 [29] 強調了身份系統中各方利益的衝突，主要表現在用戶之間的權益差異、不同系統間的互操作性問題，以及用戶與系統供應者之間的利益衝突。例如，身份系統供應者可能希望獲取更多用戶個人資料以獲取利益，而用戶則希望保護自己的隱私。這種利益衝突如果處理不當，可能導致系統環境惡化、用戶權益受到侵犯，以及市場壟斷和不公平競爭。Zuboff[39] 的研究進一步指出，這種數據收集和利用的不平等可能導致所謂的「監視資本主義」，對個人自由和社會公平造成深遠影響。

在當前的身份系統中，建立健全的信任模型仍然是一個重大挑戰。傳統的二元邏輯驗證模式（即完全信任或完全不信任）已不能滿足現代身份系統的需求。研究 [32] 指出，現實世界的信任往往是模糊而不確定的，人們很難用簡單的真假邏輯分辨用戶驗證的成功與否。例如，可能存在多組憑證同時存在，部分驗證成功，部分驗證失敗的情況，或者不同憑證的可信度和重要性各不相同。Josang 等人 [18] 提出的主觀邏輯數學框架為處理這類多組不確定性憑證的問題提供了一個系統性的解決方案。該框架將憑證的可信度表示為一個區間，從而更好地模擬了現實世界的身份驗證過程。

身份驗證的雙向性問題，尤其在去中心化系統中表現突出，正如研究 [10] 所強調的。用戶需要向系統證明自己的身份，同時系統也需要向用戶證明自己的合法性和可信度。其中，雙向的合法性可以簡單透過如 SSL 之類的驗證協議完成。但雙向的信任建立過程則可能需要建立一個多維度、連續性的信任評估機制，以滿足不同場景下的信任需求。Chohan 等人 [8] 的研究提供了 DAO（Decentralized Autonomous Organization）治理的核心概念，可以為實現這些特性提供參考。

在後續的研究中，我們將探討如何在自主身份系統中實現技術上的互相驗證與制度上的互相信任，最終構建一個能夠平衡各方利益、促進公平競爭的自主身份生態系統。這種系統將能夠在保護用戶權益的同時，也為系統供應者提供合理的發展空間，從而實現真正的平等信任。

2.4 自主身份系統的評估

鑒於當前身份系統在各個方面面臨的特定挑戰，我們提出以下評估標準，用以衡量自主身份系統的設計是否符合下一代身份系統的要求：

- 創新突破：自主身份系統的設計應符合上節所述原則，才能實現真正的技術創新。以下陳列：
 - 用戶體驗：提供無摩擦驗證，簡化操作流程，提高系統易用性。
 - 用戶認知：提供簡單的身份管理機制，幫助用戶有效管理身份，減少認知負擔。
 - 隱私保護：實施有效的隱私保護機制，保護用戶個人數據，確保隱私得到充分保障。
 - 平等信任：建立互相驗證和互相信任的機制，促進所有個體間平等的信任關係。
- 法規遵循：自主身份系統應符合相關法律法規，特別是在數據保護和隱私保護方面。例如：
 - GDPR：遵守一般資料保護規範 [11] 的要求，包括用戶同意、數據可控性和被遺忘權等規定。
 - NIST：符合美國國家標準 [26]，涵蓋多因素驗證、風險評估、身份驗證和授權等要求。

- 公認原則：遵循身份識別領域的公認原則，包括但不限於：

- 身份法則：符合 Kim Cameron 提出的身份管理基本法則 [7]，包括用戶控制、最小化披露、合法性和互操作性等原則。
- 避免常見缺陷：克服 Dhamija 等人 [10] 指出的身份管理七大缺陷，如密碼管理、安全提示和密碼重置等問題。
- 去中心化身份：符合 Allen[4] 所說，去中心化身份應該遵循的 10 個原則。



這些評估標準綜合考慮了技術創新、法律合規性和業界最佳實踐，為評估和設計下一代自主身份系統提供了全面的合理的目標。

2.5 總結

本章從身份系統的迭代、設計原則和技術評估三個方面探討了現代身份系統的設計問題。我們發現，現代身份系統的設計仍面臨著諸多挑戰，包括用戶體驗、用戶認知、隱私保護、平等信任等方面。為了解決這些問題，我們提出了一系列設計原則，包括無摩擦驗證、隱私設計、用戶認知簡化、.....等。在未來的研究中，我們將進一步探討如何在自主身份系統中達到這些評估標準，並提出相應的技術解決方案。





第三章 系統設計

本章節將會介紹我們設計的系統，並且說明其架構、資料結構以及威脅模型。

3.1 系統的重要進步

為實現用戶對其身份的完全自主權，本研究提出了多項創新機制，旨在解決現有的技術困境。

3.1.1 別名優先機制

傳統的身份驗證系統要求使用者提供唯一帳號作為登入識別。然而，這種設計無形中剝奪了用戶對個人名稱的自主權。一個連名稱都無法自主的系統，顯然難以達成本研究的設計目標。因此，我們提出讓用戶在註冊身份時優先使用自己喜歡的別名，再透過 UUID 機制產生唯一識別號。在日常操作中，系統優先讓用戶使用別名作為帳號，只有當別名因重複而難以識別時，才會要求用戶使用與 UUID 關聯的機制進行識別。當然，何時該使用別名，何時該使用 UUID，這是個複雜的問題。為此，我們提出了「基於用戶時空的分析方法」，試圖系統性地進行分析。



3.1.2 基於用戶時空的分析方法

每個身份本質上可視為一個隨時間變化的動態向量空間，其維度可謂無窮。每次對用戶的身份驗證都可視為取得特定時間僅含部分維度的向量。這方法的核心是讓用戶自主決定提供哪些維度，並在每次與系統互動時攜帶這些資訊。基於此概念，系統可藉由比對當次傳入的向量和近期暫存在快取中的所有向量來推測用戶的真實身份。

舉例來說：用戶在某次登入時提供了自己的別名、性別、所在地區等資訊，而在下次登入時僅提供了性別、所在地區等資訊，系統可以通過比對這兩次向量來推測用戶的別名。這種方法使用戶在不提供完整資訊的情況下，仍能通過部分資訊完成身份驗證。

然而，這種方法也帶來了一些問題。例如，用戶提供的維度多寡會影響系統的準確性，甚至用戶提供的維度是否包含可變資訊會影響系統的安全性。因此，我們提出維度的選擇甚至各個維度的權重應由用戶自主決定，而系統僅提供推薦機制。這樣可讓用戶在不同情境下使用不同方法，以滿足其需求。

總的來說，當僅有一個用戶被比較出來時視為可識別，有零個或多個用戶被比較出來時視為不可識別。這種機制受比較方法影響，因此我們提出了「基於危險程度的驗證機制」，希望能在不同情境下選擇適合的比較方法。

3.1.3 基於危險程度的驗證機制

不同的比較方法實質上是不同的嚴謹程度。在嚴謹程度較高的情況下，用戶可能需要更多維度符合，或需要在更接近的時間點內提供的紀錄才算數。這可能導致用戶難以找到識別對象，進而要求補充更多資訊再次驗證。相反，在嚴謹程



度較低的情況下，用戶可能僅用少量維度的資訊配合較遠時間點的紀錄來完成驗證，這可能導致用戶被誤認為其他用戶，威脅系統安全。

考慮用戶行為的危險程度，我們設計出以下規則：

1. 當用戶的行為被視為危險時，系統應提高驗證的嚴謹程度。
2. 當用戶的行為被視為安全時，系統應降低驗證的嚴謹程度。

接著，我們將嚴謹程度拆分為時間和空間兩個軸，使用笛卡爾座標系統表示，以細分出更複雜的情境：

1. 當用戶行為超危險時，比較標準應為時間非常近或多個維度符合。
2. 當用戶行為危險時，比較標準應為時間相對近或維度相對符合。
3. 當用戶行為安全時，比較標準應為時間相對遠或維度相對不符合。
4. 當用戶行為超安全時，比較標準應為時間非常遠或僅幾個維度符合。

最後，用戶行為的危險程度在我們的系統中是由用戶自主決定的，系統僅提供推薦機制。我們希望藉此滿足所有用戶在各種情境下的需求。例如，對於個人銀行帳戶，可能所有行為都視為最高危險，因此需要最高的驗證嚴謹程度。而對於個人社群帳戶，讀取文章的行為可能視為不危險，發布文章則視為危險，因此需要不同的驗證嚴謹程度。我們建議用戶將危險的概念定義為對用戶數據變動的敏感程度：越敏感的數據越危險，越危險的數據越需要嚴謹的驗證機制。

3.1.4 極致多因素認證

傳統身份驗證系統中，多因素認證被廣泛使用，但僅被視為登入時的驗證方案。然而，我們認為多因素中的每個因素都可視為一個維度，而這些維度可由用戶自主選擇。因此，我們提出了極致多因素認證的概念：我們認為不存在特定的

多因素驗證方法作為用戶登入的唯一方式，而應該反過來，從服務器的角度觀察，每個因素的驗證應該是用戶向服務器自主證明自己身份的方法。



因此，在足夠危險的情況下，可能需要連續多種因素的驗證；而在足夠安全的情況下，可能只需要一種因素的驗證。甚至即便遺失了某個重要因素，也不應被視為無法登入，而應被視為需要提供更多其他因素的驗證。

3.1.5 半匿名制

傳統身份驗證系統中，用戶身份通常不是實名制就是匿名制。然而，這種設計無形中造成了用戶隱私的困境。因此，我們提出了半匿名制的概念，讓用戶可以在關聯到真人的同時又可以拒絕被關聯到真人。半匿名機制旨在滿足 GDPR 中的被遺忘權與積極授權權，即用戶的個資只有在個人積極有意願提供時才能被使用。

解決遺忘權的設計理念是讓用戶自主掌握證明自己身份的權利。因此，只要用戶刪除了自己的身份，即使系統中仍有用戶的資料，也再無人可以證明資料的歸屬，因為唯一可以證明資料歸屬的用戶已刪除了自己的身份。

解決積極授權權的設計理念是讓用戶自主管理授權的證明。值得注意的是，這裡的用戶不只是服務的使用者，也可能是服務的提供者。因此，每個授權誕生的同時，都存在兩個證明：一個是用戶授權給服務的證明，另一個是服務接受授權的證明。這種設計可以解決現有系統中的部分問題。例如，當用戶發現個資被越權濫用時，可以出示簽章證明自己的權利，進而要求服務提供者負責。又或者，當服務提供者被指控越權濫用時，可以出示簽章證明自己的權利，進而避免被誤解。

3.1.6 基於區塊鏈的共識機制



在傳統身份管理系統中，基於安全與隱私的考量，數據孤島問題一直難以確實解決。在自主身份系統中，使用者可以用個人裝置存儲自己的身份資料，並且在加入另一個自主身份系統時，可以選擇性地分享部分資料。然而，這種設計無法徹底解決數據孤島問題。試想，如果用戶想共享的資料是有價值且可能被更改的，例如資產證明，這樣的資料不能僅靠用戶的裝置來存儲。

因此，我們提出了基於區塊鏈的共識機制。每個自主身份節點可以在鏈上交針對特定用戶數據的校驗碼，進而讓其他節點可以驗證用戶特定數據的真實性。這種設計使用戶能在不同的自主身份系統中共享資料，同時保證資料取得共識。



3.2 系統架構

3.2.1 整體說明

3.2.2 AID Server

3.2.3 Wallet

3.2.4 Consensus Core

3.3 資料結構

3.3.1 整體說明

3.3.2 AID Server

3.3.3 Wallet

3.3.4 Consensus Core

3.4 AID 的威脅模型

3.5 本章小結

系統如何解決所有缺點，並且保留所有優點。



第四章 系統實作

本研究已成功實現了系統的核心功能，並在受控的測試環境中進行了全面的概念驗證（Proof of Concept, PoC）。系統架構由三個關鍵模塊組成：AID Server、Wallet 和 Consensus Core

4.1 各模塊說明

本節將詳細介紹系統的核心架構, 包括 AID Wallet、AID Server 和 Consensus Core 三個主要模塊。表4.1概述了每個模塊支持的核心操作。

Table 4.1: 系統各模塊支持的核心操作

AID Wallet	AID Server	Consensus Core
AID Generation	AID Register	Set AID Owner
MFA Login	AID Login	Set AID Manager
Set User Data	MFA	Get AID Owner
Load User Data	RBA	Get AID Manager
Wallet Migration	AID Migration	

AID Wallet 模塊是用戶與系統交互的主要界面。它支持以下核心功能：

- **AID Generation:** 生成新的去中心化身份標識符。
- **MFA Login:** 實現多因素認證登錄。
- **Set User Data:** 允許用戶設置和更新存在個人設備的數據。
- **Load User Data:** 從個人設備中加載用戶數據。

- **Wallet Migration:** 支持錢包遷移功能, 確保用戶數據的可攜性。

AID Server 作為系統的中央服務器, 負責處理身份驗證和數據管理。其主要功能包括：

- **AID Register:** 註冊新的去中心化身份。
- **AID Login:** 處理用戶的登錄請求。
- **MFA:** 實現多因素認證機制。
- **RBA:** 執行基於風險的認證。
- **AID Migration:** 管理身份遷移過程。

Consensus Core 模塊負責維護系統的一致性和安全性。它提供以下關鍵操作：

- **Set AID Owner:** 設置或更新 AID 的所有者。
- **Set AID Manager:** 指定 AID 的管理者。
- **Get AID Owner:** 檢索 AID 的當前所有者信息。
- **Get AID Manager:** 獲取 AID 的管理者信息。

這三個模塊協同工作, 共同構成了一個安全、可靠且靈活的去中心化身份管理系統。AID Wallet 提供用戶友好的用戶端接口, AID Server 作為後端服務的一部分處理核心的身份驗證和數據管理任務, 最後 Consensus Core 則確保整個系統的一致性和可信度。

4.1.1 Wallet

本研究中的 Wallet 模塊是一個基於 Flutter 框架開發的跨平台網絡前端應用, 為用戶提供了一個直觀且功能豐富的圖形用戶界面 (GUI)。該模塊的設計和實現不僅展示了系統的可擴展性和互操作性, 還凸顯了與其他系統組件的無縫集成能



力。



Wallet 應用與 AID Server 進行了深度集成,這種集成體現了系統的模塊化設計理念,主要表現在以下幾個方面:

- **身份驗證機制:** 利用 AID Server 提供的去中心化身份驗證機制,實現了符合 W3C 標準的安全可靠的用戶登錄流程。這種方法不僅提高了系統的安全性,還增強了用戶隱私保護。
- **別名系統:** 實現了基於別名的登錄功能,這種設計不僅增強了用戶體驗的便捷性,還提供了額外的隱私保護層,使用戶能夠在不暴露真實身份的情況下進行身份驗證。
- **去中心化身份管理:** 用戶可以在應用內創建新的去中心化身份 (AID) 或註冊新的別名,這種設計體現了系統的自主性和靈活性,使用戶能夠更好地控制自己的數字身份。

Wallet 應用的核心功能模塊包括:

1. **用戶認證模塊:** 實現了基於 AID Server 的安全登錄機制,支持多因素認證和風險基礎認證 (RBA)。
2. **AI 對話模塊:** 集成了符合微軟 AI 聊天規範的對話界面,為用戶提供了基於自然語言處理 (NLP) 的智能交互體驗。
3. **去中心化身份管理模塊:** 允許用戶創建、管理和恢復多個去中心化身份 (AIDs),實現了符合 W3C DID 規範的身份管理功能。
4. **別名註冊與管理模塊:** 提供了直觀的界面,用於註冊和管理用戶別名,增強了系統的可用性和用戶隱私保護。

綜上所述,Wallet 模塊作為 AID 系統的前端範例,不僅通過與 AID Server 的緊密集成,為用戶提供了一個安全、便捷的去中心化身份驗證平台,還成功整合了最

新的 AI 技術規範。這一實現不僅證明了 AID 身份驗證系統在實際應用中替代當前中心化第三方登入系統的可行性,還為未來的去中心化身份管理和 AI 輔助服務的結合提供了寶貴的實踐經驗。




4.1.2 AID Server

本研究中的 AID Server 是一個基於 Golang 開發的高性能 Web API 服務器,其設計旨在提供自主身份認證 (Autonomous Identity, AID) 系統的核心功能。AID Server 的實現不僅體現了 AID 系統的特殊交互機制,還嵌入了第三方服務的後端功能,以驗證 AID 系統替代當前主流 OpenID 驗證系統的可行性。選擇 Golang 作為主要開發語言基於其卓越的並發處理能力、跨平台兼容性以及豐富的標準庫。特別是, Golang 的協程 (Goroutine) 機制能夠有效處理高並發請求,確保系統在高負載環境下保持穩定性能。

AID Server 實現了一系列核心功能,涵蓋自主身份 (Autonomous Identity, AID) 管理、數據存儲、人工智能集成以及用戶行為分析。在 AID 管理方面,系統支持創建、存儲和管理多重身份,每個身份可關聯多個別名,從而增強系統靈活性和用戶隱私保護。數據存儲採用了嵌入式數據庫解決方案,結合 LevelDB 和 SQLite,既簡化了部署流程,又為未來向完全去中心化應用擴展奠定了基礎。LevelDB 主要用於高性能鍵值存儲,而 SQLite 則處理需要複雜查詢的結構化數據。通過集成微軟的 OpenAI API, AID Server 提供了符合微軟 AI 聊天規範的接口,使 Wallet 模塊能夠無縫接入先進的自然語言處理功能。此外,系統還記錄和分析用戶的時空操作數據,為實施風險基礎認證 (Risk-Based Authentication, RBA) 算法提供了堅實的數據基礎。

在安全機制方面, AID Server 實現了多層次的保護策略。基於收集的用戶行



為數據，系統能夠實時評估每次操作的風險程度，實現動態的 RBA。當檢測到高風險操作時，系統會自動觸發額外的身份驗證流程。AID Server 還支持多種多因素認證（Multi-Factor Authentication，MFA）方法，包括但不限於硬體驗證和基於時間的一次性密碼（Time-based One-Time Password，TOTP）等。系統提供了靈活的 API 接口，便於未來集成新的 MFA 方法。此外，所有敏感數據在存儲前均經過加密處理，確保即使在數據泄露情況下，用戶的隱私信息也不會輕易被獲取。

在去中心化架構方面，AID Server 通過與 Consensus Core 模塊的緊密集成，實現了 AID 所有權和管理權的分布式設置與查詢。通過 Consensus Core 提供的 API 接口，AID Server 能夠建立跨節點的 AID 關聯，實現了 AID 的分布式管理。這種設計不僅增強了系統的去中心化特性，還為未來的橫向擴展提供了技術基礎。

總括而言，AID Server 作為 AID 系統的核心組件，不僅提供了高效、安全的身份管理功能，還通過集成先進的安全機制，為去中心化身份認證系統的發展奠定了堅實的理論和技術基礎。

4.1.3 Consensus Core

本研究中的 Consensus Core 模塊基於 Bitcoin 區塊鏈的硬分叉 OurChain 平台開發而成，是一套創新的智能合約系統。其設計旨在為自主身份 (Autonomous Identity, AID) 管理系統提供分布式一致性和安全性保障。OurChain 是由我們開發出來的強大的區塊鏈平台，具有高性能、低功耗和高安全性的特點。Consensus Core 模塊通過與 OurChain 平台的深度集成，為 AID 系統提供了一個安全、可靠的共識機制，確保了整個系統的一致性和可信度。

在 AID 生成過程中，Consensus Core 利用 OurChain 的共識機制確保每個 AID

的唯一性和不可篡改性。這種基於區塊鏈的共識過程具有以下優勢：



- 有效抵禦潛在的網絡攻擊
- 維護整個系統的完整性和可信度
- 為跨節點的 AID 操作提供可靠的信任基礎

Consensus Core 作為 AID 系統的信任錨點, 使用戶能夠自主控制其數字身份在多個節點間的遷移和存在。這種設計理念帶來了多重效益:

1. 增強了系統的去中心化特性
2. 賦予用戶對其數字身份的完全自主權
3. 通過區塊鏈技術實現分布式信任
4. 有效解決傳統中心化身份管理系統面臨的單點故障和信任集中化問題

Consensus Core 的設計還考慮到了系統的可擴展性和互操作性。通過標準化的智能合約接口, 該模塊為未來整合不同 AID Server 的實作預留了可能性, 進一步增強了 AID 系統的適應性和兼容性。

最後, Consensus Core 還透過智能合約實作類似內容分發網路 (Content Delivery Network, CDN) 的分散式快取伺服器 (Distributed Cache Servers) 架構。這些快取伺服器儲存了用戶網際協定 (Internet Protocol, IP) 位址與代管 AID 伺服器地址的映射關係。此設計顯著提升了跨 AID 伺服器時空分析演算法 (Cross-server Spatio-temporal Analysis Algorithm) 的運行效率。在獲取用戶的時空資訊後, 系統無需遍歷所有的 AID 伺服器來確定可能的 AID 代管者, 而是通過查詢快取伺服器即可迅速鎖定目標。這種優化措施大幅降低了系統延遲, 提高了整體性能, 同時保持了 AID 系統的分散式特性和隱私保護能力。



4.2 重要流程分析

4.2.1 AID 生成

在自主身份（Autonomous Identity，AID）系統中，AID 作為普適性的數字身份標識符，具有高度的靈活性和適應性，萬事萬物皆可以用 AID 來識別。根據實際應用需求，AID 的生成方式可以有顯著差異。本實作重點驗證了兩種主要的 AID 使用場景：

4.2.1.1 用戶自主生成 AID

在這種場景下，用戶可以在個人設備上直接生成並管理多個別名關聯的 AID。這種方法的關鍵特徵如下：

- **多因素認證 (MFA)**：用戶需要通過 MFA 來證明對 AID 的合法持有權。
- **元數據嵌入**：AID 生成過程中，除了基於 UUID (Universally Unique Identifier) 規範產生的識別號外，還需要嵌入輔助 MFA 驗證的元數據 (meta data)。
- **元數據類型**：典型的元數據包括但不限於：
 - 手機號碼
 - 電子郵件地址
 - 其他可用於身份驗證的資料...

4.2.1.2 服務提供商生成 AID

在另一種場景中，當用戶在特定服務中註冊別名時，供應商為了方便管理相同別名的用戶，會額外生成 AID 在內部取代別名用來完成用戶識別：

- **服務端管理:** AID 由服務提供商生成並管理。
- **用戶便利性:** 用戶只需記住自己的別名，無需直接管理 AID。
- **責任劃分:** 服務提供商承擔 AID 管理的責任，簡化了用戶的操作流程。



4.2.1.3 AID 生成的通用原則

無論採用哪種生成方式，AID 的創建過程都遵循以下原則：

1. **基於 UUID 規範:** AID 的核心識別號需遵循 UUID 規範，確保全局唯一性。
2. **情境相關元數據:** 根據具體使用情境，在生成過程中同步創建所需的元數據。
3. **區塊鏈記錄:** 將 AID 中需要公開的部分信息通過區塊鏈智能合約進行記錄。

這一步驟旨在：

- 確保 AID 的唯一性
- 保證 AID 信息的不可篡改性
- 提供去中心化的公開驗證機制

通過這種設計，AID 系統能夠在保證安全性和可信度的同時，為不同的應用場景提供靈活的身份管理解決方案。這種方法不僅滿足了現代數字身份管理的需求，還為未來更複雜的身份認證場景提供了可擴展的框架。

4.2.2 AID 中基於 RBA 的 MFA 驗證

在本實作中，系統對所有用戶的登入操作以及對應用程式中人工智慧（Artificial Intelligence，AI）功能的訪問進行全面監控。系統記錄包括裝置指紋（Device Fingerprint）與網際協定（IP）地址等元數據（Metadata）。這些數據將被應用於風險基礎認證（Risk-Based Authentication，RBA）演算法，以評估每次操

作的風險程度。當計算得出的風險程度超過預設閾值時，系統將自動啟動多因素認證（Multi-Factor Authentication, MFA）流程，要求用戶進行額外的身份驗證。此外，當系統檢測到用戶執行高風險行為時，會通過持續要求 MFA 驗證來提高用戶的可信度。

一個典型的風險評估範例如下：當用戶嘗試登入時，系統讀取當前用戶的裝置指紋和 IP 地址，並檢索該用戶前幾次登入的同類數據。通過比對這些數據，系統可以識別潛在的異常行為，例如 IP 不應該在不同國家或是裝置不該一直切換。若發現顯著差異，系統將要求額外的 MFA 驗證。以下提供了一個簡化的 MFA 觸發演算法示例：


Algorithm 1 MFA 觸發決策演算法範例

```
1: function NEEDMFA(uMeta, opMeta, preRec)
2:   if uMeta  $\equiv$  preRec.uMeta then
3:     if opMeta  $\approx$  preRec.opMeta then
4:       return false
5:     else
6:       return true
7:     end if
8:   end if
9:   return true
10: end function
```

其中， \equiv 表示完全等同， \approx 表示近似相等。此演算法通過比較用戶空間元數據和當前操作元數據與先前記錄的數據來決定是否需要觸發 MFA。此方法能有效地平衡安全性和用戶體驗，為系統提供了靈活且強大的身份驗證機制。

4.2.3 別名在服務器上註冊

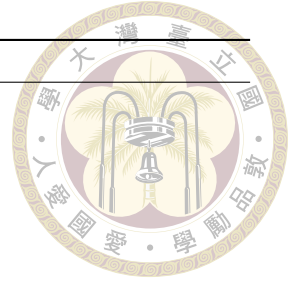
本研究提出了一種創新的匿名身份（AID）註冊和關聯機制。這種機制為使用者提供了極為便捷的服務註冊方式。他們只需要定義首選別名和簡單的數位個人識別碼（Personal Identity Number, PIN）即可完成註冊。該系統允許重複別名



和 PIN 碼，從而顯著減輕使用者的認知負擔。我們開發的時空分析演算法使 AID 伺服器能夠透過將使用者目前上傳的設備資訊與網際網路協定 (IP) 位置進行比較來分析和識別潛在的關聯 AID。在此過程中，我們實現了基於風險的身份驗證 (RBA) 的多重身份驗證 (MFA) 機制，要求用戶進行額外的驗證以增加其 AID 的可信度來完成 AID 關聯過程。這種方法的獨特之處在於它不需要使用者明確傳遞 AID。此外，由於 AID 可以綁定到另一個 AID 伺服器進行託管，基於 AID 系統的特殊機制甚至無法追蹤使用者的真實 AID，從而在便利性和隱私保護之間取得了平衡。

典型的註冊流程如下：使用者嘗試註冊使用本實現錢包中的 AI 聊天軟體，並選擇別名和 PIN 碼。AID 伺服器收到請求後，會檢查使用者提交的裝置資訊和 IP 位址，從而通過 Consensus Core 快取服務獲知該 IP 位址最近的操作是由哪些 AID 伺服器管理的，然後發布請求給這些 AID 伺服器。這些伺服器根據使用者提交的資訊執行 RBA 分析。以我們的實作為例，註冊被認為是一項重要的活動，所有請求最初都會被對應的 AID 伺服器拒絕。隨後，使用者直接連接的 AID 伺服器將要求使用者對其想要使用的 AID 進行 MFA 認證。在實作中，以一次性密碼 (OTP) 為例。使用者接收並輸入 OTP 後，提高了對應 AID 在特定時間和空間內的可信度。前端協助用戶再次發出註冊請求，此時唯一的高可信度 AID 伺服器接受用戶的註冊請求，並將用戶的別名與 AID 關聯起來。值得注意的是，在這個實作中，系統在建立別名時會建立一個 AID，並在系統內部使用 AID 來辨識別名。因此，您實際上不是將別名與 AID 關聯，而是將新的 AID 與 AID 關聯。下面提供一個簡化的處理演算法：

此演算法展示了註冊過程中的主要步驟，包括 AID 生成、伺服器查詢、MFA 驗證以及最終的註冊確認。這種設計不僅確保了用戶註冊的便利性，還通過多層次的驗證機制保障了系統的安全性和用戶的隱私。



Algorithm 2 AID 註冊與關聯流程

```
1: function REGISTERALIAS(alias, pin, ip, deviceInfo)
2:   aid ← generateAID(alias, pin)
3:   aidServerList ← remoteConsensusCore.cache(ip)
4:   for aidServer in aidServerList do
5:     if aidServer.askForBind(aid, ip, deviceInfo) then
6:       return "Registration successful"
7:     end if
8:   end for
9:   return "Registration Failed please try MFA"
10: end function
11: function MFAinWALLET(aid, ip, deviceInfo)
12:   saveRequest(ip, deviceInfo)
13:   return "MFA OK"
14: end function
15: function MAIN
16:   registerAlias(...)                                ▷ First attempt, should fail
17:   MFAinWallet(...)                                  ▷ User performs MFA
18:   registerAlias(...)                                ▷ Second attempt, should succeed
19: end function
```

4.2.4 別名在服務器上登入

本研究提出了一種創新的別名登入機制，該機制結合了時空分析演算法（Spatio-temporal Analysis Algorithm）、風險基礎認證（Risk-Based Authentication, RBA）以及多因素認證（Multi-Factor Authentication, MFA），以解決直接使用別名進行登入這一具有挑戰性的問題。

我們的實作不僅顯著提升了用戶的登入體驗，還確保了系統的安全性和可信度。典型的登入流程如下：

1. 用戶在應用程式介面中輸入別名（alias）和個人識別碼（Personal Identification Number, PIN），隨後提交登入請求。
2. 匿名身份（Anonymous Identity, AID）伺服器接收到請求後，可能會遇到多組相同別名與 PIN 碼的情況。
3. 時空分析演算法根據用戶的裝置資訊和網際協定（Internet Protocol, IP）位



址，分析並識別最可能對應的幾組別名，進而提取這些別名的 AID。

4. 系統通過共識模組（Consensus Module）中的快取查詢功能，定位這些 AID 所關聯的用戶 AID 在相應 AID 伺服器的位址。
5. 主要的 AID 伺服器向這些關聯的 AID 伺服器發送請求，要求進行 RBA 分析。

若 RBA 分析結果顯示風險程度較高，可能出現兩種情況：

- 該別名並非用戶登入的目標別名
- 該別名當前進行別名登入的風險程度過高

基於這兩種條件，系統可能面臨三種情境：

1. 沒有別名通過 RBA 分析，此時系統將要求用戶進行 MFA 驗證。
2. 僅有一個別名通過 RBA 分析，用戶可直接登入。
3. 多個別名通過 RBA 分析，表明當前難以準確識別用戶，系統將要求用戶進行額外的 MFA 驗證以增加可信度，這將有助於提高後續登入時 RBA 分析的通過機率。

本實作的創新之處在於巧妙地結合了時空分析、風險評估和多因素認證，在保障系統安全性的同時，為用戶提供了便捷的登入體驗。這種方法不僅解決了別名重複使用的問題，還通過動態風險評估和額外的身份驗證機制，確保了整個登入過程的安全性和可靠性。

4.2.5 AID 的批次創建與綁定

本實作中，我們成功實現了一個創新的身份管理流程：單一用戶可批次創建多個 AID（Autonomous Identity），並允許多個用戶將這些 AID 綁定到其個人帳戶

下。這種設計不僅展示了 AID 系統作為當前中心化第三方登入系統替代方案的可行性，還證明了其應對複雜身份管理需求的能力。一個典型的應用場景是購票系統，其中一位用戶可以為自己和家人一次性購買多張票，之後每個家庭成員可以使用自己的個人 AID 進行登入和票務管理。

本實作的簡化流程如下：用戶購票成功後，購票網站的 AID Server 創建對應張數的 AID 作為票券，並將這些 AID 關聯到各個家庭成員的 AID。因此，當家庭成員需要使用票券或登入時，只需使用自己的 AID 即可。登入過程中，系統會根據關聯的用戶 AID 執行相應的風險基礎認證（RBA）和多因素認證（MFA）。這種實現方式不僅簡化了用戶體驗，還保證了每個家庭成員的身份安全和隱私保護。

4.2.6 AID 遷移

為了實現用戶對其身份的完全自主權，本研究提出並實現了一種創新的 AID（Autonomous Identity）遷移機制。此機制允許用戶自由選擇並更換其信任的 AID 伺服器作為個人 AID 的代管者，從而增強了系統的靈活性和用戶的控制權。

本節詳細闡述了 AID 遷移機制的實現過程及其重要性：

1. **用戶自主權**: 通過允許用戶自由選擇 AID 伺服器，本機制確保了用戶對其數位身份的完全控制權。這不僅提高了系統的透明度，還增強了用戶對系統的信任度。
2. **遷移機制流程**: 本實作的 AID 遷移過程包含以下關鍵步驟：
 - 用戶通過 Wallet 應用程式發起遷移請求，指定目標 AID 伺服器地址。
 - Wallet 應用程式將遷移請求（包含目標 AID 伺服器地址和用戶的 AID）傳送至 Consensus Core 模組。
 - Consensus Core 模組利用智能合約技術，將用戶的 AID 記錄更新至區



塊鏈。

- 系統通知目標 AID 伺服器接收用戶的 AID，同時要求原 AID 伺服器刪除相關用戶資料。

3. **基於共識的操作轉移:** 遷移完成後，所有針對用戶 AID 的操作都會自動轉向新的 AID 伺服器。這是因為系統設計確保所有操作都從 Consensus Core 模組讀取最新的 AID 伺服器資訊。
4. **安全性與隱私保護:** 本機制在設計時充分考慮了安全性和隱私保護。通過使用區塊鏈技術和智能合約，確保了 AID 遷移過程的透明性和不可篡改性。同時，要求原 AID 伺服器刪除用戶資料，進一步保護了用戶隱私。
5. **系統彈性:** AID 遷移機制顯著提高了整個身份管理系統的彈性。用戶可以根據個人需求、服務質量或隱私考慮等因素，靈活地選擇或更換 AID 伺服器。

這種 AID 遷移機制不僅體現了去中心化身份管理的核心理念，還為未來更加靈活和用戶友好的身份管理系統奠定了基礎。通過賦予用戶更大的控制權，本研究在提升用戶體驗的同時，也增強了整個系統的安全性和可信度。

4.3 與現有方法之比較

本節詳細闡述了我們的實作如何解決傳統第三方登入系統的固有問題。

4.3.1 AID 系統對 GDPR 合規性的創新解決方案

隨著數據隱私保護日益受到重視，歐盟通用數據保護條例（GDPR）成為了全球範圍內最嚴格的隱私和安全法律之一。本節深入探討了 AID（Autonomous Identity）系統如何創新性地應對 GDPR 帶來的技術挑戰，並提供了超越傳統身份驗證系統的解決方案。



4.3.1.1 數據最小化和存儲限制

GDPR 規定，個人數據的收集和存儲應該在必要的最小範圍內進行。AID 系統在這方面提供了卓越的解決方案：

- **分散式數據存儲:** AID 系統不僅像傳統第三方登入系統那樣允許服務供應商存儲最小必要的數據，還進一步創新，允許用戶將核心個人信息存儲在自己的設備上。
- **按需提交機制:** 每次登入或註冊時，用戶只需提交服務所需的最小數據集給服務供應商。這種機制顯著降低了不必要的數據暴露風險。
- **動態數據管理:** AID 系統能夠根據不同服務的具體需求，動態調整所提供的個人數據範圍，確保始終符合 GDPR 的數據最小化原則。

這種創新的數據處理方式不僅符合 GDPR 的要求，還大大增強了用戶對個人數據的控制力，為數據隱私保護設立了新的標準。


4.3.1.2 被遺忘權的實現

GDPR 賦予了用戶要求刪除個人數據的權利，即所謂的”被遺忘權”。AID 系統在實現這一權利方面提出了突破性的解決方案：

1. 傳統系統的局限性:

- 主流身份驗證系統通常依賴用戶主動要求服務供應商刪除數據。
- 即使服務供應商誠實地執行刪除操作，也無法保證數據不被其他相關方保留。
- 用戶的活動日誌和互動記錄等間接數據通常難以徹底刪除。

2. AID 系統的創新方案:

- 
- **別名機制**: 用戶註冊新服務時使用別名，服務供應商只能獲知關聯的 AID 由哪個 AID Server 代管，而無法直接識別用戶的真實身份。
 - **身份解耦**: 這種設計使得即使發生數據洩露，也難以將數據準確關聯到特定真實用戶。
 - **”否認權”**: AID 系統為用戶提供了一種實質上的”否認權”，即使在極端情況下，用戶也可以合理地否認某些數據與其真實身份的關聯。

AID 系統的這種創新設計不僅技術上實現了 GDPR 的被遺忘權要求，還在概念上重新定義了數字身份的隱私保護模式。它提供了一種前所未有的隱私保護層級，即使在數據洩露的極端情況下，也能有效保護用戶的真實身份。

4.3.2 AID 系統的身份冒用防護與救援機制

在數字身份管理領域，身份冒用和身份恢復是兩個關鍵挑戰。本節詳細探討了 AID（Autonomous Identity）系統如何通過創新的”極致多因素認證”（Extreme Multi-Factor Authentication, EMFA）機制來有效應對這些挑戰，提供了超越傳統身份驗證系統的安全保障和用戶體驗。

4.3.2.1 身份冒用防護機制

傳統身份驗證系統中，用戶身份通常依賴單一或有限的識別因素（如用戶名、密碼或電子郵件地址），這種設計存在固有的安全風險：

- **單點失效風險**: 單一識別因素的洩露可能導致整個身份被盜用。
- **社會工程攻擊脆弱性**: 攻擊者可能通過欺騙或操縱獲取關鍵身份信息。
- **身份恢復機制的安全隱患**: 傳統的身份恢復過程（如密碼重置）可能被攻擊者利用。

AID 系統通過實施 EMFA 機制，有效解決了上述問題：



1. 動態多因素認證:

- 系統動態評估每次認證請求的風險級別。
- 根據風險評估結果，要求用戶提供不同數量和類型的身份因素。
- 這種方法顯著提高了身份冒用的難度，因為攻擊者需要同時掌握多種身份因素。

2. 身份因素多樣性:

- AID 系統支持廣泛的身份驗證因素，包括但不限於：
 - 知識因素（如密碼、安全問題）
 - 所有因素（如硬件令牌、智能手機）
 - 固有因素（如生物特徵）
 - 行為因素（如使用模式、地理位置）
- 這種多樣性大大增加了攻擊者成功冒用身份的難度。

3. 持續性身份驗證:

- AID 系統在整個會話過程中持續評估用戶身份的可信度。
- 如果檢測到異常行為，系統可以即時要求額外的身份驗證。

4. 去中心化身份存儲:

- 用戶的身份信息分散存儲，減少了大規模數據洩露的風險。
- 即使某些身份因素被盜，攻擊者也難以獲得足夠信息來完全冒用身份。

4.3.2.2 創新的身份救援機制

AID 系統的 EMFA 機制不僅提高了安全性，還為身份救援提供了創新解決方

案：



1. 漸進式身份恢復:

- 用戶可以通過逐步提供更多身份因素來提高自己的可信度。
- 系統根據提供的因素數量和質量，逐步授予用戶更多權限。

2. 多維度身份驗證:

- 在恢復過程中，系統考慮多種身份因素的組合，而不是依賴單一“主要”因素。
- 這種方法增加了身份恢復的靈活性，同時維持了高安全標準。

3. 風險自適應恢復流程:

- 系統根據恢復請求的風險級別動態調整所需的身份因素。
- 低風險操作可能只需要少量因素，而高風險操作則要求更嚴格的驗證。

4. 社交網絡輔助恢復:

- AID 系統引入了創新的社交恢復機制，允許用戶預先指定信任的聯繫人。
- 在緊急情況下，這些信任聯繫人可以協助驗證用戶身份，增加了恢復過程的安全性和可靠性。

4.3.2.3 安全性與可用性的平衡

AID 系統的 EMFA 機制在提供強大安全保障的同時，也注重保持良好的用戶體驗：

- **上下文感知認證:** 系統根據用戶的使用環境和行為模式動態調整認證要求，減少不必要的認證步驟。
- **漸進式安全提升:** 用戶可以逐步增加其帳戶的安全級別，而不是一次性要求所有可能的身份因素。
- **用戶友好的界面設計:** 儘管背後的機制複雜，但系統為用戶提供了直觀、易

用的界面，簡化了身份驗證和恢復過程。

總結而言，AID 系統通過其創新的 EMFA 機制，不僅有效解決了身份冒用問題，還提供了靈活、安全的身份救援方案。







第五章 結果與未來展望

5.1 結論

5.2 未來展望

商業流程....





參考文獻

- [1] A. Acquisti, I. Adjerid, R. Balebako, L. Brandimarte, L. F. Cranor, S. Komanduri, P. G. Leon, N. Sadeh, F. Schaub, M. Sleeper, Y. Wang, and S. Wilson. Nudges for privacy and security: Understanding and assisting users' choices online. ACM Comput. Surv., 50(3), aug 2017. 隱私設置的研究.
- [2] G.-J. Ahn and M. Ko. User-centric privacy management for federated identity management. In 2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007), pages 187–195, 2007.
- [3] F. Alaca and P. C. van Oorschot. Device fingerprinting for augmenting web authentication: classification and analysis of methods. In Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC '16, pages 289–301, New York, NY, USA, 2016. Association for Computing Machinery.
- [4] C. Allen. The path to self-sovereign identity, 4 2016.
- [5] M. S. Blumenthal and D. D. Clark. Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. ACM Trans. Internet Technol., 1(1), aug 2001.
- [6] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano. The quest to replace pass-

words: A framework for comparative evaluation of web authentication schemes. In 2012 IEEE Symposium on Security and Privacy, pages 553–567, 2012.

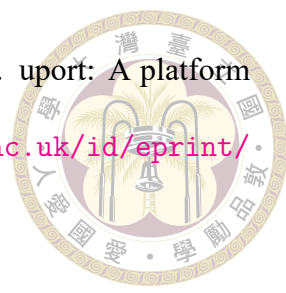


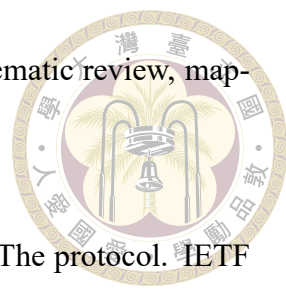
- [7] K. Cameron. The laws of identity. White paper, Microsoft Corporation, 2005.
- [8] U. W. Chohan. Decentralized autonomous organizations (daos): Their present and future. March 2024. Available at SSRN: <https://ssrn.com/abstract=3082055>.
- [9] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in cyberspace: defining tomorrow's internet. SIGCOMM Comput. Commun. Rev., 32(4), aug 2002.
- [10] R. Dhamija and L. Dusseault. The seven flaws of identity management: Usability and security challenges. IEEE Security and Privacy, 6(2):24–29, 2008.
- [11] European Parliament and Council of the European Union. General Data Protection Regulation (GDPR), 2016. Regulation (EU) 2016/679.
- [12] M. Finck. Blockchains and data protection in the european union. Eur. Data Prot. L. Rev., 4:17, 2018.
- [13] S. Ghorbani Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel. Is fido2 the kingslayer of user authentication? a comparative usability study of fido2 passwordless authentication. In 2020 IEEE Symposium on Security and Privacy (SP), pages 268–285, 2020.
- [14] M. Goodner and A. Nadalin. Web services federation language (ws-federation) version 1.2. OASIS Standard, 2009.
- [15] Google Cloud. Best practices for planning your identity architecture. <https://cloud.google.com/architecture/identity/>

[best-practices-for-planning#combine_cloud_identity_and_g_suite_in_a_single_account](#), 2024. Accessed: 2024-07-08.

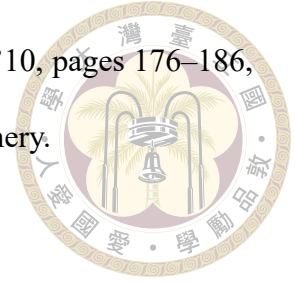


- [16] T. Hamme, V. Rimmer, D. Preuveneers, W. Joosen, M. A. Mustafa, A. Abidin, and E. Argones Rúa. Frictionless authentication systems: Emerging trends, research challenges and opportunities. 09 2017.
- [17] Hard. The oauth 2.0 authorization framework. IETF RFC 6749, 2012.
- [18] A. Jøsang, S. Marsh, and S. Pope. Exploring different types of trust propagation. In K. Stølen, W. H. Winsborough, F. Martinelli, and F. Massacci, editors, Trust Management, pages 179–192, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. 信任傳播.
- [19] A. Jøsang, M. A. Zomai, and S. Suriadi. Usability and privacy in identity management architectures. In Proceedings of the Fifth Australasian Symposium on ACSW Frontiers - Volume 68, ACSW '07, pages 143–152, AUS, 2007. Australian Computer Society, Inc. 使用這認知過度的文章.
- [20] M. Kubach, C. H. Schunck, R. Sellung, and H. Roßnagel. Self-sovereign and decentralized identity as the future of identity management? In H. Roßnagel, C. H. Schunck, S. Mödersheim, and D. Hühnlein, editors, Open Identity Summit 2020, Lecture Notes in Informatics (LNI), pages 35–47, Bonn, 2020. Gesellschaft für Informatik. DID UX.
- [21] LastPass. Psychology of passwords: The online behavior that’s putting you at risk, 2020.
- [22] L. Lessig. Code is law: On liberty in cyberspace. Harvard Magazine, 2000.

- 
- [23] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena. uport: A platform for self-sovereign identity. https://publications.aston.ac.uk/id/eprint/42147/1/uPort_SSI_DrNitinNaik.pdf, 2017.
- [24] Microsoft. Ion - we have liftoff! Microsoft Azure Blog, 2020.
- [25] Microsoft. Active directory domain services overview. <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>, 2021.
- [26] National Institute of Standards and Technology. Digital identity guidelines. Special Publication 800-63-3, National Institute of Standards and Technology, Gaithersburg, MD, June 2017.
- [27] P. Nikander, A. Gurtov, and T. R. Henderson. Host identity protocol (hip): Connectivity, mobility, multi-homing, security, and privacy over ipv4 and ipv6 networks. IEEE Communications Surveys & Tutorials, 12(2):186–204, 2010.
- [28] OASIS. Security assertion markup language (saml) v2.0 technical overview. Technical report, OASIS Committee Draft, 2005.
- [29] A. Preukschat and D. Reed. Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials. Manning Publications, May 2021. Available translations: Korean, Simplified Chinese.
- [30] M. Saemann, D. Theis, T. Urban, and M. Degeling. Investigating gdpr fines in the light of data flows. Proceedings on Privacy Enhancing Technologies, 2022.
- [31] N. Sakimura. Openid connect core 1.0. OpenID Foundation.

- 
- [32] F. Schardong and R. Custódio. Self-sovereign identity: A systematic review, mapping and taxonomy. Sensors, 22(15), 2022.
- [33] J. Sermersheim. Lightweight directory access protocol (ldap): The protocol. IETF RFC 4511, 2006. 輕量化 AD.
- [34] Y. Smirnova and V. Travieso-Morales. Understanding challenges of gdpr implementation in business enterprises: a systematic literature review. International Journal of Law and Management, 66:326–344, 01 2024.
- [35] R. Soltani, U. T. Nguyen, and A. An. A survey of self-sovereign identity ecosystem. Security and Communication Networks, 2021(1):8873429, 2021.
- [36] S.-T. Sun and K. Beznosov. The devil is in the (implementation) details: an empirical analysis of oauth sso systems. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12, pages 378–390, New York, NY, USA, 2012. Association for Computing Machinery. OAuth 2.0 實作之漏洞, 建議更改.
- [37] S. Wiefeling, M. Dürmuth, and L. Lo Iacono. What's in Score for Website Users: A Data-Driven Long-Term Study on Risk-Based Authentication Characteristics. In 25th International Conference on Financial Cryptography and Data Security, FC '21, pages 361–381. Springer, Mar. 2021.
- [38] T. Wu. Network neutrality, broadband discrimination. Journal of Telecommunications and High Technology Law, 2:141, 2003.
- [39] Y. Zhang, F. Monroe, and M. K. Reiter. The security of modern password expiration: an algorithmic framework and empirical analysis. In Proceedings of the 17th ACM

Conference on Computer and Communications Security, CCS '10, pages 176–186,
New York, NY, USA, 2010. Association for Computing Machinery.





附錄 A — 模型參數表

A.1 模型一

A.2 模型二





附錄 B — BraTS 2021 分割結果圖

B.1 編號 0001 0050

B.2 編號 0051 0100