文件檢視工具

Turnitin 原創性報告

已處理到: 2024年07月15日 14:36 CST
代碼: 2388838480
字數: 13165
已提交：2

應用_FIDO_於支付服務商間交易的嚴格顧客驗證.pdf
經由 典翰 蔡

相似度指標	依來源標示相似度	
3%	Internet Sources:	3%
	出版物:	0%
	學生文稿:	1%

包含引用 | 包含參考書目 | 不含小型相符結果 | 模式: 快速觀看(典型的)報告 | 列印 | 下載

1% match (從 2023年06月22日 來的網絡)
https://money.udn.com/money/story/5613/7232380?from=edn_hotestlist_storybottom

<1% match (從 2023年09月04日 來的網絡)
https://www.fsc.gov.tw/uploaddowndoc?file=news%2F202308151434570.pdf&filedisplay=%E9%99%84%E4%BB%B61-%E9%87%91%E8%9E%8D%E7%A7%91%E6%8A%80%E7%99%BC%E5%B1%95%E8%B7%AF%E5%BE%91%E5%9C%96%282.0%29.

<1% match (從 2023年05月22日 來的網絡)
https://waseda.repo.nii.ac.jp/?action=repository_action_common_download&attribute_id=20&file_no=1&item_id=38840&item_no=1

<1% match (從 2024年05月31日 來的網絡)
https://etheses.lib.ntust.edu.tw/thesis/detail/f5306db3c7cecd18fffe8795b5005b0b/?seq=1

<1% match (從 2024年05月29日 來的網絡)
https://etheses.lib.ntust.edu.tw/thesis/detail/e2dacfcbbba35a92d4f99129b2c13356/?seq=1

<1% match (從 2024年03月25日 來的網絡)
https://ah.lib.nccu.edu.tw/bitstream/140.119/146899/1/602301.pdf

<1% match (從 2024年03月27日 來的網絡)
https://opendata.ey.gov.tw/NewOpenData/XML/191

<1% match (從 2022年09月27日 來的網絡)
https://wantrich.chinatimes.com/search/%E8%AA%8D%E8%AD%89/2

<1% match (從 2020年11月04日 來的網絡)
https://titangene.github.io/article/create-a-virtual-environment-and-manage-dependencies-with-conda.html

<1% match (從 2022年12月09日 來的網絡)
https://www.pmu.edu.sa/attachments/academics/pdf/udp/coe/dept/me/fall2020_2021/design_of_experiments_study_the_performance_of

<1% match (從 2024年06月13日 來的學生文稿)
Submitted to Asia University on 2024-06-13

<1% match (從 2023年12月24日 來的網絡)
http://blog.chinaunix.net

<1% match (從 2024年05月05日 來的網絡)
https://ir.lib.nycu.edu.tw/bitstream/11536/79542/1/452101.pdf

<1% match (從 2019年05月27日 來的學生文稿)
Submitted to Hankuk University of Foreign Studies on 2019-05-27

<1% match (從 2024年07月02日 來的網絡)
https://www.businessperspectives.org/images/pdf/applications/publishing/templates/article/assets/13288/IMFI_2020_01_Binda.pdf

<1% match (出版物)
Hanoi University

<1% match (從 2024年02月25日 來的網絡)
https://assets.kpmg.com/content/dam/kpmg/tw/pdf/2023/12/tw-kpmg-taiwan-insurance-report.pdf

<1% match (從 2023年08月24日 來的網絡)
https://support.google.com/google-ads/answer/2375433?hl=zh-Hant

<1% match (從 2023年12月29日 來的網絡)
https://vdocuments.com.br/cefccoeeccceeeoe-eccecece-aoecefccoeeccceeeoea.html

<1% match (從 2020年07月05日 來的網絡)
http://www.digitalwall.com

<1% match (從 2022年08月08日 來的網絡)
https://www.chinatimes.com/search/%E6%8E%88%E6%AC%8A%E6%8A%80%E8%A1%93?page=2

國立政治大學資訊管理學系 碩士學位論文 應用FIDO於支付服務商間交易的嚴格顧客驗證Applying FIDO for Strong Customer Authentication in Transactions Between Payment Service Providers 指導教授：陳恭博士研究生：蔡典翰撰中華民國113年6月 致謝 在這段重要的學術旅程中，我充滿感激地回顧過去兩年在碩士班的學習經歷。這段時間不僅豐富了我的知識視野，也讓我在人生的道路上學會了如何應對挑戰和成長。在這裡，我想跟所有在這段期間給予我指導、陪伴、幫助的師長、同學、親友們表達我最深的謝意。 首先，我要由衷感謝我的指導教授陳恭老師。老師深厚的學識和敏銳的見解，引導我在FIDO這個富有挑戰性的資訊安全領域進行研究。每當我遇到困難時，老師總能提供寶貴的建議和解決方案。此外，我還要感謝論文提案與正式口試的口委老師們——廖峻鋒老師、郭桐惟老師、莊永裕老

師。你們的指導和建議，使我的研究更臻完善，對於你們的幫助，我深表感謝。 接著，我也要感謝實驗室的同學和政大資管所的同學們，感謝你們在學習和生活中的支持與陪伴。和你們一起度過的碩士時光，是我最珍貴的回憶之一。因為有你們，使這段求學之路充滿了溫暖和力量，讓我能不畏艱難，持續向前邁進。 最後，我要感謝我的家人和朋友們，感謝你們的無條件支持和鼓勵。你們的信任和陪伴，是我不斷前進的動力。每當我面臨困難和挑戰時，你們的支持總是能提供我最大的力量。再次感謝大家，有你們的陪伴與支持，讓我的碩士之路充滿溫暖與信心。 i 摘要 隨著數位化和網路科技的快速發展，支付服務商在推動電子商務和數位經濟的過程中扮演著至關重要的角色。這不僅提升了支付的便利性，同時也帶來了一系列資訊安全上的新挑戰。特別是在支付服務商間進行交易時，如何提升使用者身份驗證的便利性、確保交易的安全性，成為了當前需面臨的問題。目前台灣在處理此類問題的作法，主要依賴傳統的密碼系統與簡訊動態密碼驗證，這些方法雖然普及，但可能存在安全漏洞、不便利性等潛在問題。台灣金融監督管理委員會近年來積極推動金融領域採用F-FIDO標準，旨在通過更現代化的驗證方式，提升金融交易的安全性和使用者體驗。有鑑於此，本論文提出一個結合FIDO標準與嚴格顧客驗證的方案，應用公私鑰加密技術提升支付服務商間交易的安全，為支付服務商間交易的安全性提供一個新的解決思路。 在系統實作方面，本研究模擬支付服務商間的轉帳服務，根據FIDO2的Webauthn標準，建立一個中心化依賴方，提供於中心化依賴方建立第二組公私鑰，作為二次身份驗證的基礎。此外，當使用者欲進行支付服務商間的轉帳服務時，需先通過嚴格顧客驗證，使用者於中心化依賴方的安全執行環境中，運用其在中心化依賴方所註冊的私鑰進行身份驗證，並確認交易細節後，便可執行支付服務商間的轉帳服務。 關鍵字:金融FIDO、FIDO2、Webauthn、支付服務商、嚴格顧客驗證 ii Abstract The rapid development of digital and internet technologies has made pay- ment service providers essential to e-commerce and the digital economy. While enhancing payment convenience, it also brings new security challenges, especially in user authentication and transaction security. In Taiwan, tradi- tional password systems and SMS OTPs are common but have potential secu- rity vulnerabilities. The Financial Supervisory Commission promotes the F- FIDO standard to enhance security and user experience. This paper proposes a solution combining FIDO2 Webauthn and strong customer authentication, using asymmetric encryption to secure inter-provider transactions with a cen- tralized relying party for secondary authentication. This study simulates inter-provider transfer services based on the FIDO2 Webauthn standard. It establishes a centralized relying party for secondary authentication with a second set of public-private keys. Users must pass strict customer authentication, using their private key registered with the central- ized relying party in a secure environment to authenticate their identity and confirm transaction details before executing transfers. Keywords: F-FIDO, FIDO2, Webauthn, Payment Service Provider, Strong Customer Authentication iii 目次 致謝.....摘要..... Abstract .....目次.....表次.....圖次..... 1緒論..... 1.1研究背景與動機..... 1.2研究目的與方法..... 1.3研究貢獻..... 1.4論文架構..... 2技術背景與相關研究..... 2.1技術背景..... 2.1.1 FIDOUAF ..... 2.1.2 FIDO2:Webauthn標準與CTAP協定..... 2.1.3公私鑰身份識別..... 2.2相關研究..... 2.2.1金融FIDO ..... 2.2.2支付服務商..... 2.2.3嚴格顧客驗證..... 3系統設計..... 3.1系統設計理念..... 3.1.1 系統名詞定義. . . . . 3.2 系統架構概觀..... 3.2.1 系統架構設計..... 3.2.2 資料庫設計..... 3.3系統流程設計. . . . . 3.3.1系統註冊. . . . . 3.3.2系統驗證. . . . . 3.3.3二次身份註冊與驗證. . . . . 3.3.4嚴格顧客驗證與跨機構轉帳. . . . . i ii iii iv vi vii 1 1 1 2 2 3 3 3 5 7 8 8 12 14 16 16 16 16 16 18 24 24 25 27 28 iv 4 系統實作與展示..... 4.1 4.2系統開發環境與工具. . . . . .程式碼設計. . . . . 4.2.1系統註冊/驗證. . . . . 4.2.2系統基本功能. . . . . 4.2.3二次身份註冊與驗證. . . . . 4.2.4嚴格顧客驗證與跨機構轉帳. . . . . 4.3系統實作展示. . . . . 4.3.1系統註冊/驗證. . . . . 4.3.2系統基本功能. . . . . 4.3.3二次身份註冊與驗證. . . . . 4.3.4嚴格顧客驗證與跨機構轉帳. . . . . 4.4系統評估與限制. . . . . 4.4.1 系統評估..... 4.4.2 系統限制..... 5結論與未來研究方向..... 5.1 結論..... 5.2 未來研究方向..... 參考文獻.....附錄..... 32 32 33 33 36 36 39 42 42 46 46 47 48 48 49 50 50 50 51 54 v 表次 表3.1表3.2表3.3表3.4表3.5表3.6表3.7表3.8表4.1 db_general的PSP資料表欄位設計. . . . . db_general的User資料表欄位設計. . . . . db_general的Credential資料表欄位設計. . . . . db_general的Authenticator資料表欄位設計. . . . . db1的User資料表欄位設計. . . . . db1的Transfer資料表欄位設計. . . . . db1的Credential資料表欄位設計. . . . . db1的Authenticator資料表欄位設計. . . . . .系統開發環境與工具說明. . . . . 19 19 19 20 22 22 22 23 32 vi 圖次 圖2.1無密碼體驗（UAF標準） . . . . . .圖2.2 FIDO UAF高階架構. . . . . .圖2.3 FIDO 2專案. . . . . .圖2.4 FIDO參考架構. . . . . .圖2.5使用Face ID或Touch ID來存取鑰匙圈項目. . . . . .圖2.6兆豐網路ATM註冊流程—金融FIDO註冊. . . . . .圖2.7兆豐網路ATM註冊流程—晶片金融卡拔插卡確認. . . . . .圖2.8兆豐實體ATM註冊流程—金融FIDO註冊. . . . . .圖2.9兆豐實體ATM註冊流程—顯示QR Code相關提示訊息. . . . .圖2.10兆豐身份認證APP裝置綁定流程—掃描ATM產生的QR Code圖2.11註冊iPASS MONEY —金融支付工具驗證方法二擇一. . . . . .圖2.12 iPASS MONEY金融支付工具驗證—信用卡. . . . . .圖2.13 iPASS MONEY金融支付工具驗證—銀行帳戶. . . . . .圖3.1系統架構圖. . . . . .圖3.2 db_general資料表之間的實體關係圖. . . . . .圖3.3 db1資料表之間的實體關係圖. . . . . .圖3.4應用FIDO2進行系統註冊之循序圖. . . . . .圖3.5應用FIDO2進行系統驗證之循序圖. . . . . .圖3.6二次身份註冊與驗證之循序圖. . . . . .圖3.7嚴格顧客驗證之循序圖. . . . . .圖3.8運用資料庫兩階段提交，執行跨機構轉帳的功能之循序圖. . . .圖4.1系統註冊（客戶端）—請求註冊選項與挑戰碼. . . . . .圖4.2系統註冊（伺服器端）—產生註冊選項與挑戰碼. . . . . .圖4.3系統註冊（客戶端）—簽署挑戰碼. . . . . .圖4.4系統註冊（伺服器端）—驗證挑戰碼. . . . . .圖4.5系統驗證（客戶端）—請求驗證選項與挑戰碼. . . . . .圖4.6系統驗證（伺服器端）—產生驗證選項與挑戰碼. . . . . .圖4.7系統驗證（客戶端）—簽署挑戰碼. . . . . .圖4.8系統驗證（伺服器端）—驗證挑戰碼. . . . . .圖4.9支付服務商1（客戶端）—檢查二次身份註冊與驗證的狀態. . .圖4.10支付服務商1（伺服器端）—編碼使用者資訊token . . . . . .圖4.11中心化依賴方（伺服器端）—解碼使用者資訊token . . . . . vii 4 4 6 7 8 9 9 10 10 11 12 13 13 18 21 23 25 26 28 30 31 33 34 34 34 35 35 35 36 37 37 37 圖4.12中心化依賴方（伺服器端）—路由器CORS設定. . . . . 38圖4.13中心化依賴方（客戶端）—請求註冊選項與挑戰碼. . . . . 38圖4.14中心化依賴方（伺服器端）—產生註冊選項與挑戰碼. . . . . 38圖4.15中心化依賴方（客戶端）—簽署挑戰碼. . . . . 39圖4.16中心化依賴方（伺服器端）—驗證挑戰碼. . . . . 39圖4.17中心化依賴方（客戶端）—請求驗證選項與挑戰碼. . . . . 40圖4.18中心化依賴方（伺服器端）—產生驗證選項與挑戰碼. . . . . 40圖4.19中心化依賴方（客戶端）—簽署挑戰碼. . . . . 40圖4.20中心化依賴方（伺服器端）—驗證挑戰碼. . . . . 40圖4.21中心化依賴方（伺服器端）—啟動跨機構轉帳流程. . . . . 41圖4.22支付服務商1（伺服器端）—準備. . . . . 41圖4.23支付服務商1（伺服器端）—提交. . . . . 42圖4.24支付服務商1（伺服器端）—回滾. . . . . 42圖4.25支付服務商1 —系統註冊頁面的註冊選項. . . . . 43圖4.26支付服務商1 —系統註冊頁面，選擇驗證器. . . . . 43圖4.27支付服務商1 —系統註冊頁面，使用指紋辨識. . . . . 44圖4.28支付服務商1 —系統驗證頁面的驗證選項. . . . . 45圖4.29支付服務商1 —系統驗證頁面，選擇驗證器. . . . . 45圖4.30支付服務商1 —系統驗證頁面，使用指紋辨識. . . . . 46圖4.31支付服務商1 —基本功能. . . . . 46圖4.32支付服務商1 —二次身份註冊與驗證頁面. . . . . 47圖4.33中心化依賴方—確認交易細節後，嚴格顧客驗證與跨機構轉帳48 viii 第一章緒論 近年來，支付服務商（Payment Service Provider, PSP）已提升民眾的支付便利性，並推動了電子商務及數位經濟的成長。然而，隨著服務的多元化，資訊安全已成為一個關鍵的議題。本文將透過設計和實現一個基於FIDO2（Fast Identity Online Universal Authentication Framework 2）應用公私鑰加密技術來完成嚴格顧客驗證後，才能進行支付服務商之間的轉帳服務。 1.1研究背景與動機 金融服務與人們的日常生活息息相關，例如:存款、提領、轉帳等。隨著時代與金融科技相關技術的蓬勃發展，支付服務商能提供如同實體銀行的各項金融服務，因此越來越多的使用者開始使用支付服務商所提供的各項金融服務。 關於資訊安全對於使用者身份驗證的三要素:所知之事，所持之物，所具之形。所知之事為只有正確使用者才知道的事，例如:圖型、密碼、動靜態通行碼等;所持之物為正確使用者才持有之物，例如:感應卡、晶片卡等;所具之形為正確使用者才具有的形狀，例如:指紋、臉形、聲紋等。 在現行的金融服務流程，皆需對使用者的再次進行身份驗證後，才能使用後續的金融服務。例如:線上購物，欲刷信用卡消費時，需要用手機進行簡訊動態密碼驗證;提領時，需要再次輸入使用者密碼;註冊支付服務商帳號時，需要完成金融支付工具驗證流程。這些措施的目的，即是為了確保使用者的身份，以避免被盜用的風險。然而，關於使用者的二次身份驗證，存在許多可以改善的空間，例如:簡訊動態密碼驗證需要支付簡訊費用;使用者密碼有遺忘，或是被盜用的風險;金融支付工具驗證流程要求綁定信用卡或銀行帳戶，需要額外的驗證流程。 因此，本研究期望在FIDO2架構下，應用公私鑰加密技術，讓使用者能建立第二組公私鑰對，作為二次身份驗證的基礎，並提升金融機構之間的交易安全性、便利性。 1.2研究目的與方法 本研究旨在建立一套新型的二次身份驗證機制，於支付服務商之間的交易時，能遵守嚴格顧客驗證的標準，並且能夠提升交易過程中的安全性與使用者操作的便利性。研究方法將涵蓋從理論探討到技術實作，包括分析現有的身份驗證問題，設計基於FIDO2的身份驗證流程，並進行系統開發與測試，以驗證所提出解決方案之可行性。透過本研究，我們期望能為金融科技提供新的思

路和方向，並讓支付服務商在未來的發展，帶來一定的啟示作用。 1.3研究貢獻 本研究聚焦於優化支付服務商的使用者身份驗證流程，以及強化支付服務商間的交易安全性，透過基於FIDO2的二次身份驗證機制，以及運用嚴格顧客驗證與交易確認，能為台灣的金融科技發展帶來了以下幾項主要貢獻： •優化使用者的身份驗證流程：透過實施FIDO2標準，本研究目標建立了一套較便利的二次身份驗證機制。此機制不僅降低了傳統密碼系統的風險，例如:網路釣魚攻擊、重放攻擊、密碼疲勞等問題，同時也能提升使用者身份驗證流程的便利性。 •強化支付服務商間的交易安全性：透過在交易過程中實施強化後的身份驗證措施，本研究可提升交易的安全性與可信任度。這對於提升支付服務商的接受度和推動數位經濟的發展具有重要意義。 •實務應用與理論貢獻：本研究不僅在實務上為提升支付服務商的系統安全提供了可行的解決方案，同時也在理論上豐富了FIDO2在金融科技領域應用的研究文獻。支付服務商亦能參考本研究之內容，思考該如何將金融FIDO導入現行的使用者二次身份驗證流程。透過對此次實施過程的深入分析，本研究為未來在相關領域的研究提供了新的視角與研究基礎。 1.4論文架構 本研究分為五個章節，第一章緒論包含研究背景與動機、研究目的與方法、研究貢獻、論文架構。第二章技術背景與相關研究，將針對本研究會使用到的背景技術進行介紹，以及相關研究做分析。第三章系統設計，將詳細介紹FIDO2模擬支付服務商間交易的平台之設計理念、系統架構、流程設計。第四章系統實作，將介紹系統開發環境與工具、程式碼設計、執行結果展示。並於最終第五章，闡述本研究之結論與未來研究方向的建議。 第二章技術背景與相關研究 本章節旨在闡述本研究之技術背景與相關研究，為進一步探討應用FIDO於支付服務商間交易的嚴格顧客驗證提供理論基礎與技術支持。 在技術背景方面，本文首先介紹FIDO聯盟推出的一系列身份驗證標準，包括FIDO UAF、FIDO U2F以及後續發展的FIDO2標準。FIDO 2標準下的Webauthn標準與CTAP協定，代表著身份驗證技術的新進展，這些標準透過提供更安全、更便利的使用者身分驗證方法，有效提升使用者體驗、交易安全性。此外，本章節進一步闡述公私鑰加密技術，以macOS作業系統為例，詳細說明其對通行密鑰的建立與儲存機制，展現了加密技術在保障資料安全方面的重要性。 在相關研究部分，文本簡要介紹了金融FIDO的概念，以及其在台灣的發展現狀，反映了身份驗證技術在金融行業中的應用與發展趨勢。接著，對支付服務商的角色與功能進行定義與說明，凸顯了支付服務商在促進電子商務與數位支付領域中的關鍵作用。最後，針對嚴格顧客驗證進行概述，說明其定義、實施現況以及其帶來的諸多優勢，包括提升交易安全性、可信任度等，指出嚴格顧客驗證在現代金融安全領域中的重要性與應用前景。 2.1技術背景 2.1.1 FIDO UAF 在當前數位時代，網路安全與身份驗證技術的進步對於保障個人與企業的資訊安全至關重要。在此背景下，FIDO（Fast Identity Online）聯盟提出了一系列創新的身份驗證標準，其中包括FIDO Universal Authentication Framework （UAF）與FIDO Universal 2nd Factor （U2F）。FIDO聯盟的目標是透過建立一個更安全、便利、且易於使用的身份驗證標準，來減少對傳統密碼的依賴。FIDO UAF提供了一個框架，支援多種生物辨識技術和其他非密碼形式的驗證方法，讓使用者在不使用傳統密碼的情況下進行身份驗證。而FIDO U2F是一種基於第二因素驗證的機制，允許使用者透過實體裝置進行身份驗證，從而增強安全性。 FIDO UAF的主要目標是提升網路安全性與使用者的便利性。透過減少對密碼的依賴，FIDO UAF旨在降低重放攻擊等資訊安全風險。此外，FIDO UAF透過支援生物辨識技術，例如:指紋掃描、臉部辨識等非傳統身份驗證方法，提供了一個更加便利且安全的身份驗證流程。這不僅能夠提升安全性，還能夠提升使用者便利性，進而促進廣泛的應用與接受度，如圖2.1所示。 圖2.1:無密碼體驗（UAF標準） FIDO Alliance, 2020g FIDO UAF的高階架構包括:客戶端（Client）、伺服器端（Server）、FIDO UAF標準。在這一架構中，客戶端則是使用者設備上的軟體部分，負責處理使用者的驗證資料並與伺服器端進行通訊;伺服器端負責管理註冊、驗證請求，並進行相關的安全檢查以確保交易的安全性。FIDO UAF定義了客戶端與伺服器端之間交換訊息的標準格式，以確保訊息的安全傳輸與身份驗證的有效性。 圖2.2: FIDO UAF高階架構FIDO Alliance, 2020d 在FIDO UAF的應用中，有幾個專有名詞是常見且關鍵的，如圖2.2所示。以下分別介紹 • FIDO驗證器（FIDO Authenticator）：是用於使用者驗證的裝置，負責建立與儲存驗證證明（例如:私鑰），以及在驗證過程中，依據使用者的生物特徵或其他安全因素，向FIDO客戶端提供證明。驗證器的目的是提升安全性，確保驗證過程在本地裝置上完成，避免敏感訊息在網路上傳輸。 • FIDO客戶端（FIDO Client）:是在客戶端上執行的軟體，負責協調使用者與FIDO驗證器之間的交流。它接收來自伺服器端（依賴方）的驗證請求，引導使用者完成驗證流程，並將驗證結果安全地傳遞給FIDO伺服器。FIDO客戶端能為使用者提供一個統一且安全的驗證界面。 • FIDO伺服器（FIDO Server）:是在伺服器端上執行的軟體，由服務提供商（例如:網站或應用開發者）執行。它負責處理來自FIDO客戶端的註冊、驗證請求，包括驗證使用者提供的驗證資訊。FIDO伺服器還負責維護使用者驗證資訊的安全、完整性，並確保僅在使用者成功驗證後才授予對供應資源的存取權限。 •依賴方（Relying Party）:依靠FIDO驗證系統來驗證使用者身份的實體，可以是一個網站、應用程式或任何需要安全使用者驗證的服務。依賴方會透過與FIDO伺服器的交流，確保只有經過合法驗證的使用者才能存取其提供的服務或資源。依賴方的角色是在驗證過程中代表最終服務提供者，確保安全規範的遵循與實施。 2.1.2 FIDO 2: Webauthn標準與CTAP協定FIDO2是由FIDO聯盟推出的下一代開放身份驗證標準，旨在提供網際網路一套更安全、更便利的身份驗證方法。FIDO2標準包括W3C的Web Authentication （WebAuthn）規範和FIDO聯盟的Client to Authenticator Protocol （CTAP），如圖2.3所示。FIDO2標準延續之前的FIDO UAF和FIDO U2F標準，提供了一種更強大且靈活的驗證機制，支援多種驗證方法，包括生物辨識、硬體安全鑰匙等，使得無密碼登入成為現實。這代表FIDO驗證技術的一個重大進步，旨在提升網路安全性與使用者體驗。 WebAuthn是一項網路標準，由世界廣域網聯盟（W3C）和FIDO聯盟共同開發，允許網站透過支援的瀏覽器、作業系統，使用公開金鑰密碼學進行使用者登入。WebAuthn標準讓使用者可以使用個人身份驗證裝置（例如:指紋掃描器、相機、硬體安全鑰匙等）在不使用密碼的情況下，能夠安全地登入網站。這不僅提升安全性，減少網路釣魚攻擊的風險，也簡化了登入流程，優化使用者體驗。 CTAP是FIDO2標準的一部分，分為CTAP1和CTAP2兩個版本。CTAP1基於之前的FIDO U2F協定，允許運用現有裝置（例如: FIDO安全鑰匙）透過USB、 NFC、藍牙與使用者的設備交流，進行身份驗證。CTAP2則引進了更多的功能與更高的靈活性，可支援外部驗證器（例如:手機），並且與WebAuthn緊密整合，允許使用者使用各種生物辨識技術進行無密碼驗證。CTAP2的推出，進一步擴展了FIDO2標準的應用範圍，使得跨設備和跨平台的無密碼驗證成為現實。 圖2.3: FIDO 2專案FIDO Alliance, 2021b 在FIDO身份驗證概念中，主要發生於由依賴方所控制的計算機環境，與待驗證使用者所控制的環境之間，在依賴方的環境概念上包含至少一個網頁伺服器與FIDO伺服器;而使用者環境，包含FIDO使用者設備、一個或多個FIDO驗證器、一個FIDO客戶端的軟體所組成、使用者代理軟體（User Agent software），使用者代理軟體可能是瀏覽器，或者是由依賴方交付的獨立應用程式。無論哪種情況，FIDO客戶端，雖然是一個概念上獨立的實體，實際上可能在使用者代理的範疇內全部或部分實現。就網路身份驗證而言，網頁瀏覽器實現了FIDO客戶端功能的主要部分，而底層作業系統（platform）則實現了驗證器特定模組（Authenticator-specific Module, ASM）的部分(FIDO Alliance, 2020a)，如圖2.4所示。 圖2.4: FIDO參考架構FIDO Alliance, 2021a 2.1.3公私鑰身份識別 公私鑰加密是一種非對稱加密技術，每個使用者擁有一對密鑰：公鑰和私鑰。公鑰是公開的，可以安全地分享給任何人;私鑰是保密的，只有擁有者知道。這兩個密鑰在數學上是相關連的，使用公鑰加密的訊息只能用對應的私鑰解密。這種機制能確保訊息的加密傳輸與身份驗證，而不需要共享密鑰本身。 常見的非對稱加密技術包括: RSA, ECC。前者是基於大數分解的難題，並且其安全性在於選擇足夠大的鍵長，適用於加密和數位簽名;後者是基於橢圓曲線數學的加密技術，提供相對於RSA更高的安全性和效率，特別是在需要較小密鑰大小的場景中。目前Webauthn支援使用EdDSA, ES256, RS256等公鑰加密演算法。 以macOS為例，若要存取私鑰，使用者會透過安全框架（Security framework） API對LocalAuthentication framework所提供的介面，以生物辨識技術來進行身份驗證。接著，安全區域（Secure Enclave）會將傳入的生物辨識資訊和其儲存的生物辨識資訊進行比對。隨後，安全區域會傳回通過/失敗結果，以管理存取鑰匙圈內容（Key management）的權限（圖2.5）。安全區域提供了額外的保護層，尤其適用於儲存重要的私鑰，使用者或作業系統軟體都無法存取底層身份驗證資料，例如:儲存的指紋(Apple Inc., 2023)。 圖2.5:使用Face ID或Touch ID來存取鑰匙圈項目Apple Inc., 2024 2.2相關研究 2.2.1金融FIDO 金融FIDO （Financial Fast Identity Online）是指泛金融產業中能共通之行動身份識別工具，民眾如欲辦理銀行、證券、保險等金融相關業務，只要運用金融FIDO進行身份驗證，即可線上輕鬆完成申辦，帶動數位金融創新應用蓬勃發展。讓民眾可以方便地透過生物辨識等方式在線上辦理銀行、證券、保險等金融業務，促進數位金融創新。 2021年5月，在金管會的推動下成立了「金融行動身份識別聯盟（F-FIDO)」，由跨領域金融機構共同參與，並設立業務、安控、技術委員會，致力於制定應用範圍、安全標準、技術規範。金融FIDO以我國金融體系既存之共通核驗識別「信物」一晶片金融卡作為金融FIDO註冊之身份核驗工具，提高了身份驗證的安全性和便利性(金融監督管理委員會, 2021)。 現階段以兆豐銀行為例，使用者可持晶片金融卡於網路ATM（圖2.6、圖2.7）或實體ATM（圖2.8）申請金融FIDO註冊的服務，由ATM產生專屬於使用者該次認證之QR Code（圖2.9)。接著，使用者可使用行動裝置下載「兆豐身分認證」 APP，並於兆豐身分認證APP中掃描ATM產生的QR Code（圖2.9)，以完成裝置綁定及設定生物辨識(兆豐銀行, 2024a)。 圖2.6:兆豐網路ATM註冊流程—金融FIDO註冊兆豐銀行, 2024c 圖2.7:兆豐網路ATM註冊流程—晶片金融卡拔插卡確認兆豐銀行, 2024c 圖2.8:兆豐實體ATM註冊流程—金融FIDO註冊兆豐銀行, 2024b 圖2.9:兆豐實體ATM註冊流程—顯示QR Code相關提示訊息兆豐銀行, 2024b 圖2.10:兆豐身份認證APP裝置綁定流程—掃描ATM產生的QR Code兆豐銀行, 2024d 2022年1月，金管會推出「數位身份認證及授權」的主題式監理沙盒及業務試辦，旨在結合民眾需求和業者風險控管，引進安全可靠的數位驗證解決方案。此舉讓民眾可以更安心地使用數位金融服務，應用範圍包括線上開戶、業務申辦、行動銀行登入等多元場景(金融監督管理委員會, 2022)。 2023年8月，金管會在其發布的金融科技發展路徑圖2.0中，提出金融FIDO V2計畫。計畫考慮擴大服務範圍至金融以外領域，例如: Mydata平台，並允許金控作為自體系金融FIDO伺服器，支援子公司使用(金融監督管理委員會, 2023b)。 2.2.2支付服務商 支付服務商（Payment Service Provider, PSP）是專門為商家提供簡化的支付處理服務的公司，透過整合複雜的支付流程，使商家能夠更方便地接受信用卡、電子錢包、銀行轉帳等電子支付方式。此外，目前支付服務商之間已支援跨機構轉帳的服務(一卡通MONEY, 2023)。隨著網際網路的興起，許多知名的支付服務商，例如:iPassMoney、街口支付、PayPal、Stripe等應運而生，提升了線上交易的便利性。 現階段以iPassMoney為例，當使用者欲註冊新帳號時，需以信用卡或銀行帳戶（圖2.11），若選擇信用卡進行金融支付工具驗證，需要輸入詳細的信用卡資訊作驗證（圖2.12）;而若選擇銀行帳戶進行金融支付工具驗證，以國泰世華銀行為例，需另外跳轉到國泰世華的網路銀行畫面，才能進行身分認證（圖2.13)。當完成金融支付工具驗證以後，才能使用支付服務商間的轉帳服務(一卡通MONEY, 2024)。對於現行支付服務商的金融支付工具驗證流程，若能導入FIDO2，就能為商家和消費者提供一個更便利的支付環境。 圖2.11:註冊iPASS MONEY —金融支付工具驗證方法二擇一 圖2.12: iPASS MONEY金融支付工具驗證—信用卡 圖2.13: iPASS MONEY金融支付工具驗證—銀行帳戶 2.2.3嚴格顧客驗證 嚴格顧客驗證（Strong Customer Authentication, SCA）是2015年歐盟通過的支付服務指令2（Payment Services Directive 2, PSD2）中的一項要求，旨在保護進行支付的終端使用者。嚴格顧客驗證的概念會經常與雙因素驗證（2FA）聯結在一起，需要使用者提供以下三種因素中的其中兩種： •知識（例如:密碼或PIN碼） •持有物（例如:晶片金融卡或智慧型手機） •固有特徵（例如:指紋、臉型） 此外，遠距支付交易（remote payment transaction），亦即透過網際網路或裝置，遠距離發起支付交易的行為，若要符合支付服務指令2的嚴格顧客驗證要

求，除了雙因素驗證以外，這整個過程必須在安全的執行環境中進行，並且展示交易細節以確保正確的交易被授權（dynamic linking）。在支付服務指令中，亦有提到當付款人於支付服務商線上存取其帳戶、發起電子支付交易、透過遠距方式進行任何可能會發生支付詐欺或其他濫用風險的操作時，需進行嚴格顧客驗證;另一方面，為了提供更流暢的使用者體驗，當付款人進行較低金額或較低風險的支付操作時，可豁免嚴格顧客驗證的要求，例如:小額支付、定期支付(The European Parliament and the Council of the European Union, 2015)。由歐盟所制定的支付服務指令2已於2018年1月生效，而嚴格顧客驗證，也已於2019年9月生效，具有強制力。歐盟實施嚴格顧客驗證的原因包括減少詐騙和保持開放銀行（Open Banking）的市場准入，這對於當前和未來都十分重要。 開放銀行及其延伸的嚴格顧客驗證，為支付市場帶來了新的商業模式和挑戰，並且能為消費者、商家、金融服務公司，帶來以下好處 •對消費者:嚴格顧客驗證可能要求使用應用程式進行身份驗證，而非接收簡訊，這在線上購物時可能會增加一些步驟，但主要好處是能透過支付服務指令2的保護措施來降低詐騙率，提升交易安全性。 •對商家:開放銀行和嚴格顧客驗證為商家帶來了更具競爭力的支付環境，降低了收款的整體成本。透過嚴格顧客驗證作為數位身份解決方案的重要部分，能為金融以外的新商業機會創造機會，降低新使用者加入的成本。許多公司已經使用嚴格顧客驗證和開放銀行來吸引和驗證新使用者。 •對金融服務公司:嚴格顧客驗證提供了一種驗證客戶身份的方式，正確實施能有效降低詐騙風險。使用智慧型手機進行嚴格顧客驗證還可以降低成本，因為比相比於現行的簡訊動態密碼驗證，使用智慧型手機的生物辨識技術更方便，同時也能提供更好的使用者體驗(Caccavello & Okay Inc., 2022)。 第三章系統設計 3.1系統設計理念 為了模擬實現應用FIDO於支付服務商間交易的嚴格顧客驗證，本研究設計兩個支付服務商與一個中心化依賴方，前者能提供使用者註冊（registration）、驗證（authentication）、儲值、提領、轉帳、跨機構轉帳的服務。使用者需先至中心化的FIDO依賴方註冊第二組公私鑰，來完成二次身份註冊與驗證，其中第二組公鑰會儲存在中心化的FIDO伺服器所連接的資料庫，而第二組私鑰將會儲存在客戶端，例如: macOS的安全區域。之後，當使用者欲執行支付服務商間的交易，例如:跨機構轉帳時，會先跳轉到中心化的FIDO依賴方的頁面中，透過二次身份註冊與驗證時，所使用的私鑰進行嚴格顧客驗證，並確認該次交易細節後，便能執行跨機構轉帳的服務。 3.1.1系統名詞定義•中心化依賴方（rp_general）:為各支付服務商的使用者，提供二次身份註冊與驗證、嚴格顧客驗證、跨機構轉帳的服務。 •支付服務商1 （rp1）:提供使用者應用FIDO2註冊/驗證帳號，並支援儲值、提領、轉帳、跨機構轉帳、二次身份註冊與驗證的功能。此外，也會儲存每個使用者當前的二次身份註冊與驗證狀態。 •支付服務商2（rp2）:同上述支付服務商1。 3.2系統架構概觀 3.2.1系統架構設計 本研究之系統是基於圖2.2 FIDO UAF高階架構、圖2.4 FIDO參考架構而設計，旨在模擬支付服務商間的轉帳服務，主要分成兩大區塊，分別為客戶端與伺服器端，其系統架構圖詳見圖3.1。 為了於本地端建立本研究之應用FIDO於支付服務商間交易的嚴格顧客驗證的模擬系統，根據MDN官方文件對於origin的定義(MDN Web Docs, 2023)，且考量Webauthn規範中對於RP ID的要求與限制(W3C, 2023a)，以及Chrome瀏覽器對於Webauthn的支援與限制。本研究運用OpenSSL自發證書，並用子網域的方式，區分中心化依賴方、支付服務商1、支付服務商2的origin，分別如下所示。 •中心化依賴方的origin: https://rp-general.localhost:1000 •支付服務商1的origin: https://rp1.localhost:3000 •支付服務商2的origin: https://rp2.localhost:4000 在客戶端部分，當使用者透過筆記型電腦上的信賴方應用程式（Relying Party App）發起一次註冊/驗證請求時，伺服器端會將該註冊/驗證的選項與挑戰碼（challenge）回傳到瀏覽器。瀏覽器再運用Webauthn API與作業系統進行交流。當註冊/驗證請求已傳送給作業系統，它將與內建驗證器（Bound Authenticator），來啟動註冊/驗證。若是於註冊流程中，內建驗證器將建立私鑰（Private Authenti- cation Key）;若是於驗證流程中，內建驗證器將使用已經註冊的私鑰，在本地端對使用者進行身份驗證。 在伺服器端部分，信賴方應用程式伺服器（Relying Party App Server）會包含網頁伺服器與FIDO伺服器（FIDO Server）。當客戶端的註冊/驗證請求發送到伺服端時，由網路伺服器收來自客戶端的註冊/驗證的請求資料後，它將這些資料轉發給FIDO伺服器。FIDO伺服器會使用FIDO註冊/驗證API處理這些請求，並進行必要的註冊/驗證操作。接著，若是於註冊流程中，FIDO伺服器會為使用者建立一個新的帳號，並將該使用者的公鑰（Public Authentication Key）儲存在資料庫伺服器中;若是於驗證流程中，FIDO伺服器將驗證結果與資料庫伺服器中儲存的公鑰進行比對驗證。若比對成功，表示使用者被成功驗證。整個伺服器端流程能確保註冊與驗證過程的完整性與安全性，並為客戶端所發起的註冊/驗證請求提供了完善的支援與驗證機制。 當使用者欲進行二次身份註冊與驗證時，需先登入各支付服務商的頁面，選擇進行二次身份註冊與驗證後，會跳轉到中心化依賴方的頁面，使用者需在中心化依賴方註冊第二組公私鑰，來完成二次身份註冊與驗證。第二組私鑰也會儲存在客戶端的內建驗證器中，而第二組公鑰會儲存在中心化依賴方所連接的資料庫中，供未來進行嚴格顧客驗證時，比對使用者的驗證結果用。 使用者需先完成二次身份註冊與驗證後，才能使用支付服務商間交易的服務，例如:跨機構轉帳。關於嚴格顧客驗證，使用者會運用二次身份註冊與驗證所建立的私鑰，於網頁瀏覽器的頁面上確認該次跨機構轉帳的交易細節，並簽署綁定該次交易細節的挑戰碼後，對中心化依賴方發起嚴格顧客驗證請求，接著，中心化依賴方會運用使用者二次身份註冊與驗證所建立的公鑰進行驗證，若通過驗證，中心化依賴方便會執行跨機構轉帳的服務。這樣便能確保該次跨機構轉帳的操作，可以滿足嚴格顧客驗證要求，亦即金融交易必須在安全的執行環境（僅開放各支付服務商的網域才能連接）中，讓使用者能確認該次交易細節，並針對該次交易細節所發起的挑戰碼（動態連結），透過至少兩種獨立的身份驗證元素，包括持有物（例如:筆記型電腦的內建驗證器）與固有特徵（例如:指紋），以完成嚴格顧客驗證的要求。圖3.1:系統架構圖 3.2.2資料庫設計 承本研究的系統架構圖，依賴方rp_general, rp1, rp2分別連接到db_general, db1, db2資料庫，以下分別介紹這些資料庫設計，以及其分別的欄位說明與資料表之間的實體關係圖。 首先，rp_general作為中心化的依賴方，連接到db_general，主要用來記錄各支付服務商之下，其分別的使用者，所進行的嚴格顧客驗證的相關資訊，總共有四張資料表，分別是psp, user, credential, authenticator，如下表3.1、表3.2、表3.3、表3.4所示。PSP資料表用來記錄當前有參與二次身份註冊與驗證的支付服務商名稱的清單; User資料表用來記錄使用者帳號、使用者名稱與其分別所隸屬的支付服務商名稱，另外，因系統設計考量到全域唯一性的問題，因此設定psp, username, account三個欄位，共同組合為複合主鍵; Credential資料表用來記錄憑證的相關資訊，其中id欄位為主鍵，並設定psp, username, account三個欄位，共同組合為外部鍵，與User資料表相關聯; Authenticator資料表用來記錄驗證器的相關資訊，其中credentialID欄位作為外部鍵，與Credential資料表相關聯，並設定psp, username, account三個欄位，共同組合為外部鍵，與User資料表相關聯。 db_general資料庫中的各資料表之關係，以實體關係圖呈現如圖3.2所示。 欄位名稱型別定義name String支付服務商的名稱created_at Timestamp建立時間 表3.1: db_general的PSP資料表欄位設計 欄位名稱型別定義psp String支付服務商的名稱name String使用者名稱account String使用者帳號created_at Timestamp建立時間 表3.2: db_general的User資料表欄位設計註: db_general的User資料表中，psp, name, account三個欄位，共同組合作為複合主鍵。 欄位名稱型別定義id String憑證的id，此欄位為Credential資料表的主鍵psp String支付服務商的名稱username String使用者名稱account String使用者帳號created_at Timestamp建立時間updated_at Timestamp更新時間 表3.3: db_general的Credential資料表欄位設計 註:在db_general的Credential資料表中，psp, username, account三個欄位，共同組合作為複合外部鍵，與User資料表相關聯。 欄位名稱型別定義credentialID String憑證的id psp String支付服務商的名稱username String使用者名稱account String使用者帳號credentialPublicKey String公開的驗證密鑰counter bigInteger驗證次數的計數器credentialDeviceType String建立/使用憑證的設備類型credentialBackedUp Boolean憑證是否已備份，以便在原始設備丟失、損壞或更換時能夠恢復存取權限transports String支援憑證通訊的不同技術或方法，定義了憑證如何在使用者的設備（例如:安全密鑰、智慧型手機、電腦）和要求驗證的服務之間傳輸。 表3.4: db_general的Authenticator資料表欄位設計 註:在db_general的Authenticator資料表中，psp, username, account三個欄位，共同組合作為複合外部鍵，與User資料表相關聯。 圖3.2: db_general資料表之間的實體關係圖 接著，rp與rp2皆為相同的資料庫設計，但是分別連接到資料庫db1, db2，主要用來記錄各支付服務商中，關於使用者、轉帳、註冊/驗證的相關資訊，總共有四張資料表，分別是user, transfer, credential, authenticator，以db1為例，如下表3.5、表3.6、表3.7、表3.8所示。User資料表用來記錄使用者的相關資訊，以及其當前的二次身份註冊與驗證狀態; Transfer資料表以自動增加的id編號作為主鍵，該資料表主要用來記錄使用者之間的轉帳與跨機構轉帳的紀錄; Credential資料表用來記錄憑證的相關資訊，其中id欄位為主鍵，account欄位與User資料表相關聯; Authenticator資料表用來記錄驗證器的相關資訊，其中credentialID欄位作為外部鍵，與Credential資料表相關聯。 psp1資料庫中的各資料表之關係，以實體關係圖呈現如圖3.3所示 欄位名稱型別定義name String使用者名稱account String使用者帳號balance Integer帳號餘額isVerified Boolean是否已完成二次份驗證created_at Timestamp建立時間updated_at Timestamp更新時間 表3.5: db1的User資料表欄位設計 欄位名稱型別定義id Number轉帳紀錄的id，此欄位為Transfer資料表的主鍵fromPSP String從哪個支付服務商轉帳from String從哪個使用者帳號轉出toPSP String要轉帳到哪個支付服務商to String要轉入哪個使用者帳號amount Number交易金額note String交易備註trxId String記錄資料庫交易的id，供查詢log使用status Boolean記錄資料庫交易的執行狀態，供查詢log使用created_at Timestamp建立時間updated_at Timestamp更新時間 表3.6: db1的Transfer資料表欄位設計 欄位名稱型別定義id String憑證的id，此欄位為Credential資料表的主鍵account String使用者帳號，此欄位為外部鍵，與User資料表相關聯created_at Timestamp建立時間updated_at Timestamp更新時間 表3.7: db1的Credential資料表欄位設計 欄位名稱型別定義credentialID String憑證的id，此欄位為外部鍵，與Credential資料表相關聯account String使用者帳號credentialPublicKey String公開的驗證密鑰counter bigInteger驗證次數的計數器credentialDeviceType String建立/使用憑證的設備類型credentialBackedUp Boolean憑證是否已備份，以便在原始設備丟失、損壞或更換時能夠恢復存取權限transports String支援憑證通訊的不同技術或方法，定義了憑證如何在使用者的設備（例如:安全密鑰、智慧型手機或電腦）和要求驗證的服務之間傳輸。 表3.8: db1的Authenticator資料表欄位設計 圖3.3: db1資料表之間的實體關係圖 3.3系統流程設計本研究之系統流程設計，根據FIDO UAF高階架構圖的註冊操作(FIDO Alliance, 2020b)與驗證操作(FIDO Alliance, 2020c)與FIDO UAF的註冊操作循序圖(FIDO Alliance, 2020e)與驗證操作循序圖(FIDO Alliance, 2020f)，以宏觀的角度用循序圖來呈現，如圖所示，主要可分為系統註冊（registration）、系統驗證（authentication）、二次身份註冊與驗證、嚴格顧客驗證（strong customer authentication）與跨機構轉帳，以下分別介紹。 3.3.1系統註冊 關於使用者於系統註冊新帳號的流程，可參考圖3.4所示。使用者於瀏覽器的頁面上輸入使用者名稱後，透過瀏覽器向網路伺服器發送系統註冊請求後，再到資料庫查詢該使用者目前擁有的驗證器相關資料。接著，FIDO伺服器基於所得之資訊產生註冊選項與挑戰碼（challenge），挑戰碼經作業系統傳遞給內建驗證器，要求使用者同意後，將產生一組新的公私鑰對，由使用者運用私鑰簽署挑戰碼，並將私鑰儲存在客戶端的內建驗證器中。之後，瀏覽器會向網路伺服器發送註冊結果與公鑰，由FIDO伺服器驗證註冊結果與挑戰碼，若通過驗證，便會將使用者的註冊資料與公鑰儲存在伺服器端的資料庫。最後會通知使用者，即完成系統註冊流程。 圖3.4:應用FIDO2進行系統註冊之循序圖 3.3.2系統驗證 關於使用者於系統驗證的流程，可參考圖3.5所示。使用者於瀏覽器的頁面上輸入使用者名稱後，透過瀏覽器向網路伺服器發送系統驗證請求後，再到資料庫查詢該使用者目前擁有的驗

證器相關資料。接著，FIDO伺服器基於所得之資訊產生驗證選項與挑戰碼（challenge），挑戰碼經作 業系統傳遞給內建驗證器進行使用者驗證。當使用者運用私鑰簽署挑戰碼後，會由瀏覽器對網路伺服器發送驗證結果，由FIDO伺服器運用使用者的公鑰，驗證驗證結果與挑戰碼，若通過驗證，便會通知使用者，即完成系統驗證流程。 圖3.5:應用FIDO2進行系統驗證之循序圖 3.3.3二次身份註冊與驗證 關於二次身份註冊與驗證的流程，可參考圖3.6所示。使用者於瀏覽器頁面中登入支付服務商1的首頁，選擇二次身份註冊與驗證的功能後，會跳轉到中心化依賴方的首頁，進行二次身份註冊與驗證流程，也就是在中心化依賴方建立第二組公私鑰對，其流程如同系統註冊流程，其中第二組私鑰仍會儲存在客戶端的內建驗證器中，而使用者的註冊資料與第二組公鑰會儲存在中心化依賴方的資料庫。最後會通知使用者二次身份註冊與驗證結果，即完成二次身份註冊與驗證流程。 圖3.6:二次身份註冊與驗證之循序圖3.3.4嚴格顧客驗證與跨機構轉帳當使用者欲執行支付服務商間的交易（例如:跨機構轉帳）時，系統會先檢查使用者是否已完成二次身份註冊與驗證。若已完成二次身份註冊與驗證，便可進行嚴格顧客驗證的流程。 使用者可於瀏覽器頁面上，確認使用者名稱、該次跨機構轉帳的交易細節，例如:交易對象的支付服務商、交易對象的帳號、交易金額等。當確認交易細節完畢後，使用者可透過瀏覽器對中心化依賴方進行嚴格顧客驗證，由中心化依賴方的FIDO伺服器，基於所得之資訊產生驗證選項與綁定該次交易細節的挑戰碼（challenge），挑戰碼經作業系統傳遞給內建驗證器進行使用者驗證。當使用者運用二次身份註冊與驗證所註冊的私鑰簽署挑戰碼後，會由瀏覽器對網路伺服器發送驗證結果，由FIDO伺服器運用二次身份註冊與驗證所註冊的公鑰，驗證驗證結果與挑戰碼，若通過驗證，便會通知使用者，即完成嚴格顧客驗證流程，如圖3.8所示。 當使用者通過嚴格顧客驗證後，便能執行跨機構轉帳的服務。由中心化依賴方擔任資料庫兩階段提交（two-phase commit）的協調者，支付服務商1與支付服務商2擔任參與者，中心化依賴方會依據兩個參與者於第一階段的準備結果，決定第二階段的操作，若雙方皆正常準備完成，即進行正式提交（commit）;而若有任一方於準備階段失敗時，則雙方皆需進行回滾（rollback）。最後，會通知使用者跨機構轉帳的交易結果，如圖3.8所示。 圖3.7:嚴格顧客驗證之循序圖 圖3.8:運用資料庫兩階段提交，執行跨機構轉帳的功能之循序圖 第四章系統實作與展示 本章節將介紹系統實作與展示，內容涵蓋系統開發環境與工具、程式碼設計、系統實作展示、系統評估與限制。首先，我們將介紹這套模擬支付服務商間交易系統所使用的開發環境與工具。在程式碼設計和系統實作展示部分，將展示該模擬系統的主要功能，包括系統基本功能、系統註冊/驗證、二次身份註冊與驗證、嚴格顧客驗證，以更全面地呈現本研究的功能與成果。最後，本研究將對該模擬系統進行評估，以驗證其在實際應用中的效益和潛在限制。 4.1系統開發環境與工具 本系統實作所使用的開發環境與工具如下表4.1所示。在前端部分，使用EJS套件作為伺服器端渲染畫面的模板引擎，搭配@simplewebauthn/browser函式庫，實作Webauthn客戶端的底層邏輯。在後端部分，使用Node.js，搭配@simplewebauthn/server函式庫，實作Webauthn伺服器端的底層邏輯(Matthew Miller, 2024)。在作業系統與驗證器部分，使用macOS作業系統，以及其內建的Touch ID作為驗證器，提供使用者進行FIDO, Webauthn指紋辨識用。 項目名稱說明作業系統macOS Sonoma 14.2.1實作這套模擬系統的作業系統版本驗證器裝置MacBook內建Touch ID用於進行FIDO, Webauthn指紋辨識整合開發環境Visual Studio Code微軟提供的整合開發環境，用於開發本系統的前端網頁畫面、後端伺服器前端EJS模板引擎用於伺服器端渲染畫面後端Node.js v20.12.2用於後端伺服器開發Express用於Node.js後端伺服器開發的框架重要函式庫@simplewebauthn/browser在SimpleWebauthn專案中所包含的函式庫，便於我們實作Webauthn客戶端的底層邏輯@simplewebauthn/server在SimpleWebauthn專案中所包含的函式庫，便於我們實作Webauthn伺服器端的底層邏輯@mysql2用於支援MySQL v8.0.0以上的版本的強化密碼驗證功能redis用於作為Redis鍵值資料庫的客戶端連線程式資料庫MySQL v8.2.0用於支援Mysql v8.0.0以上的版本的強化密碼驗證功能Redis v7.2.4用於儲存FIDO伺服器所發出的挑戰碼 表4.1:系統開發環境與工具說明 4.2程式碼設計 本研究之模擬系統支援FIDO2系統註冊/驗證帳號，改善現行多數支付服務商註冊/驗證帳號時，使用者可能會面臨密碼疲勞等問題。本模擬系統以現行主流的支付服務商所提供之功能作為參考依據，例如:儲值、提領、轉帳。此外，將現行需透過信用卡或銀行帳戶，進行二次身份註冊與驗證的流程，改為應用FIDO2建立第二組公私鑰的方式來完成，優化身分驗證流程的使用者體驗。最後，也導入嚴格顧客驗證的流程於跨機構轉帳的服務中，強化支付服務商之間的交易安全性。首先，會先介紹程式碼設計的部分，再展示系統實作的畫面。 4.2.1系統註冊/驗證關於系統註冊/驗證的功能，根據FIDO2所設計，各可再分為客戶端、伺服器端，以下分別說明程式碼設計。 •系統註冊•客戶端一請求註冊選項與挑戰碼:使用者於瀏覽器的頁面上，輸入使用者名稱，並選擇相關的註冊選項設定參數後，對伺服器端啟動註冊流程，請求註冊選項與挑戰碼（圖4.1）。 •伺服器端一產生註冊選項與挑戰碼:由伺服器端的FIDO伺服器，根據客戶端傳入的註冊選項設定參數，透過generateRegistrationOptions()函式，產生註冊選項與挑戰碼，將挑戰碼儲存在伺服器端的Redis資料庫後，將註冊選項與挑戰碼回傳給客戶端（圖4.2）。 •客戶端一簽署挑戰碼:當瀏覽器接收到伺服器端回傳的註冊選項與挑戰碼時，會透過startRegistration()函式，產生一組新的公私鑰對，使用私鑰簽署挑戰碼，並將私鑰儲存在客戶端的內建驗證器後，發送系統註冊結果與公鑰給伺服器端（圖4.3）。 •伺服器端一驗證挑戰碼:當伺服器端收到系統註冊結果與公鑰後，由FIDO伺服器運用之前儲存在Redis資料庫的挑戰碼，透過verifyRegistrationResponse()函式做驗證，若通過驗證，就將該使用者、公鑰等註冊相關資料儲存在伺服器端資料庫後，回傳系統註冊的驗證結果給客戶端（圖4.4）。 圖4.1:系統註冊（客戶端）—請求註冊選項與挑戰碼 圖4.2:系統註冊（伺服器端）—產生註冊選項與挑戰碼 圖4.3:系統註冊（客戶端）—簽署挑戰碼 圖4.4:系統註冊（伺服器端）—驗證挑戰碼•系統驗證•客戶端一請求驗證選項與挑戰碼:使用者於瀏覽器的頁面上，輸入使用者名稱，並選擇相關的驗證選項設定參數後，對伺服器端啟動驗證流程，請求驗證選項與挑戰碼（圖4.5）。 •伺服器端一產生驗證選項與挑戰碼:由伺服器端的FIDO伺服器，根據客戶端傳入的驗證選項設定參數，透過generateAuthenticationOptions()函式，產生驗證選項與挑戰碼，將挑戰碼儲存在伺服器端的Redis資料庫後，將驗證選項與挑戰碼回傳給客戶端（圖4.6）。 •客戶端一簽署挑戰碼:當瀏覽器接收到伺服器端回傳的驗證選項與挑戰碼時，會透過startAuthentication()函式，使用私鑰簽署挑戰碼後，發送系統驗證結果給伺服器端（圖4.7）。 •伺服器端一驗證挑戰碼:當伺服器端收到系統驗證結果後，由FIDO伺服器運用之前儲存在Redis資料庫的挑戰碼，透過verifyAuthenticationResponse()函式與公鑰做驗證，若通過驗證，就回傳系統驗證的驗證結果給客戶端（圖4.8）。 圖4.5:系統驗證（客戶端）—請求驗證選項與挑戰碼 圖4.6:系統驗證（伺服器端）—產生驗證選項與挑戰碼 圖4.7:系統驗證（客戶端）—簽署挑戰碼 圖4.8:系統驗證（伺服器端）—驗證挑戰碼4.2.2系統基本功能本研究之模擬支付服務商的系統中，提供的基本功能包括儲值、提領、轉帳，分別說明如下。 •儲值:檢查儲值操作的正確性後，增加使用者餘額。 •提領:檢查提領操作的正確性後，減少使用者餘額。 •轉帳:檢查轉帳操作的正確性後，僅限轉帳給同一支付服務商內的其他使用者。 4.2.3二次身份註冊與驗證關於二次身份註冊與驗證的功能，主要是由使用者於中心化依賴方註冊第二組公私鑰，再通過驗證後，即完成二次身份註冊與驗證流程。如同前述系統註冊的功能，根據FIDO2所設計，可再分為客戶端、伺服器端，以下分別說明程式碼設計。 •二次身份註冊與驗證•支付服務商1（客戶端）—檢查二次身份註冊與驗證的狀態:使用者透過瀏覽器登入支付服務商1的首頁，選擇開始進行二次身份註冊與驗證時，會先檢查當前使用者的二次身份註冊與驗證的狀態（圖4.9），未經過二次身份註冊與驗證的使用者才需要繼續進行。接著，再經由支付服務商1的伺服器端取得包含使用者資訊編碼的JWT token（圖4.10）。之後，再跳轉到中心化依賴方的二次身分註冊與驗證的頁面，由中心化依賴方的伺服器端解碼使用者資訊的JWT token，取得使用者資訊（圖4.11）。此外，中心化依賴方的伺服器端可透過CORS設定白名單清單，確保只有被允許的支付服務商能呼叫中心化依賴方的API（圖4.12）。 •中心化依賴方（客戶端）—請求註冊選項與挑戰碼:使用者於瀏覽器的頁面上，輸入使用者名稱，並選擇相關的註冊選項設定參數後，對中心化依賴方的伺服器端啟動註冊流程（圖4.13）。 •中心化依賴方（伺服器端）—產生註冊選項與挑戰碼:由中心化依賴方的伺服器端的FIDO伺服器，根據客戶端傳入的註冊選項設定參數，透過generateRegistrationOptions()函式，產生註冊選項與挑戰碼，將挑戰碼儲存在伺服器端的Redis資料庫後，將註冊選項與挑戰碼回傳給客戶端（圖4.14）。 •中心化依賴方（客戶端）—簽署挑戰碼:當瀏覽器接收到伺服器端回傳的註冊選項與挑戰碼時，會透過startRegistration()函式，產生一組新的公私鑰對，使用私鑰簽署挑戰碼， 並將私鑰儲存在客戶端的內建驗證器後，發送二次身份註冊與驗證結果與公鑰給伺服器端（圖4.15）。 •中心化依賴方（伺服器端）—驗證挑戰碼:當伺服器端收到二次身份註冊與驗證結果與公鑰後，由FIDO伺服器運用之前儲存在Redis資料庫的挑戰碼，透過verifyRegistra- tionResponse()函式做驗證，若通過驗證，就將該使用者、公鑰等註冊相關資料儲存在伺服器端資料庫後，回傳二次身份註冊與驗證的驗證結果給客戶端（圖4.16）。 圖4.9:支付服務商1（客戶端）—檢查二次身份註冊與驗證的狀態 圖4.10:支付服務商1（伺服器端）—編碼使用者資訊token 圖4.11:中心化依賴方（伺服器端）—解碼使用者資訊token 圖4.12:中心化依賴方（伺服器端）—路由器CORS設定 圖4.13:中心化依賴方（客戶端）—請求註冊選項與挑戰碼 圖4.14:中心化依賴方（伺服器端）—產生註冊選項與挑戰碼 圖4.15:中心化依賴方（客戶端）—簽署挑戰碼 圖4.16:中心化依賴方（伺服器端）—驗證挑戰碼 4.2.4嚴格顧客驗證與跨機構轉帳 關於嚴格顧客驗證與跨機構轉帳的功能，主要是由使用者於支付服務商發起，需先至中心化依賴方完成嚴格顧客驗證後，才能進行跨機構轉帳功能。接著，會由中心化依賴方擔任協調者，支付服務商1與支付服務商2擔任參與者，進行資料庫的兩階端提交，完成跨機構轉帳的功能，以下分別說明程式碼設計。 首先，使用者登入支付服務商的頁面，點選跨機構轉帳服務時，系統會先檢查使用者當前的二次身份註冊與驗證狀態，若已完成，便能進行嚴格顧客驗證;而若尚未完成，就會跳轉到中心化依賴方的頁面，先完成二次身份註冊與驗證後，才能繼續進行嚴格顧客驗證。接著，使用者透過瀏覽器，將交易資訊發送給中心化依賴方，並跳轉到中心化依賴方的嚴格顧客驗證與跨機構轉帳的頁面。 關於嚴格顧客驗證的功能，主要是運用使用者於中心化依賴方，二次身份註冊與驗證時所註冊的公私鑰進行驗證。如同前述系統驗證的功能，根據FIDO2所設計，各可再分為客戶端、伺服器端，以下分別說明程式碼設計。 •嚴格顧客驗證•中心化依賴方（客戶端）—請求驗證選項與挑戰碼:使用者可於中心化依賴方的嚴格顧客驗證與跨機構轉帳的頁面上，確認該次交易細節的所有參數後，對中心化依賴方的伺服器端啟動嚴格顧客驗證流程（圖4.17）。 •中心化依賴方（伺服器端）—產生驗證選項與挑戰碼:由伺服器端的FIDO伺服器，根據客戶端傳入的驗證選項設定參數，透過generateAuthenticationOptions()函式，產生驗證39選項，並將該次交易細節綁定於挑戰碼中，將挑戰碼儲存在伺服器端的Redis資料庫後，將認證選項與挑戰碼回傳給客戶端（圖4.18）。 •中心化依賴方（客戶端）—簽署挑戰碼:當瀏覽器接收到伺服器端回傳的驗證選項與挑戰碼時，會透過startAuthentication()函式，使用私鑰簽署挑戰碼後，發送嚴格顧客驗證結果給伺服器端（圖4.19）。 •中心化依賴方（伺服器端）—驗證挑戰碼:驗證嚴格顧客驗證的驗證選項與挑戰碼:當伺服器端收到嚴格顧客驗證結果後，由FIDO伺服器運用之前儲存在Redis資料庫的挑戰碼，透過verifyAuthenticationResponse()函式與公鑰做驗證，若通過驗證，就回傳嚴格顧客驗證的驗證結果給客戶端（圖4.20）。 圖4.17:中心化依賴方（客戶端）—請求驗證選項與挑戰碼 圖4.18:中心化依賴方（伺服器端）—產生驗證選項與挑戰碼 圖4.19:中心化依賴方（客戶端）—簽署挑戰碼 圖4.20:中心化依賴方（伺服器端）—驗證挑戰碼 •跨機構轉帳•中心化依賴方（伺服器端）—啟動跨機構轉帳流程:因為支付服務商間的交易，算是分散式系統的交易行為，參考MySQL XA交易的官方文件(Oracle Corporation, 2024)，因此可運用資料庫的兩階段提交，來完成跨機構轉帳的功能（4.21）。 •支付服務商1（伺服器端）—兩階段提交:由中心化依賴方擔任協調者，支付服務商1

與支付服務商2擔任參與者，中心化依賴方會依據兩個參與者於第一階段的準備結果，決定第二階段的操作。若雙方皆正常準備完成（4.22），即進行正式提交（commit）（圖4.23）;而若有任一方於準備階段失敗時，則雙方皆需進行回滾（rollback）（圖4.24）。最後，會通知使用者跨機構轉帳的交易結果。 圖4.21:中心化依賴方（伺服器端）──啟動跨機構轉帳流程 圖4.22:支付服務商1（伺服器端）──準備 圖4.23:支付服務商1（伺服器端）──提交 圖4.24:支付服務商1（伺服器端）──回滾 4.3系統實作展示 延續本研究之程式碼設計章節，本章節將展示系統操作的具體步驟及相關畫面。主要包含系統基本功能、使用者註冊/驗證、二次身份註冊與驗證、嚴格顧客驗證與跨機構轉帳，以下分別展示並說明。 4.3.1系統註冊/驗證 系統註冊:使用者於支付服務商1的系統註冊頁面中，輸入使用者名稱，可視情況調整預設的註冊選項，點選確認按鈕後，即啟動註冊流程（圖4.25）。接著，使用者可選擇欲使用哪一種驗證器（圖4.26）。最後，根據Webauthn API的規範(W3C, 2023b)，本研究之模擬系統預設的註冊選項中的Attachment是platform，表示會優先使用內建驗證器，以本研究之系統開發環境與工具為例，就會使用macbook的Touch ID指紋辨識作為驗證方式（圖4.27），即完成系統註冊的流程。 圖4.25:支付服務商1 ──系統註冊頁面的註冊選項 圖4.26:支付服務商1 ──系統註冊頁面，選擇驗證器 圖4.27:支付服務商1 ──系統註冊頁面，使用指紋辨識 系統驗證:使用者於支付服務商1的系統驗證頁面中，輸入使用者名稱，可視情況調整預設的驗證選項，根據Webauthn API的規範(W3C, 2023c)，本研究之模擬系統預設的驗證選項中的User Verification是preferred，表示依賴方偏好在操作過程中進行使用者驗證，然而即使沒有進行使用者驗證，操作仍可以繼續進行，但這並非最理想的情況。點選確認按鈕後，即啟動驗證流程（圖4.28）。接著，使用者可選擇欲使用哪一種驗證器（圖4.29）。最後，使用macbook的Touch ID指紋辨識作為驗證方式（圖4.30），即完成系統驗證的流程。 圖4.28:支付服務商1 ──系統驗證頁面的驗證選項 圖4.29:支付服務商1 ──系統驗證頁面，選擇驗證器 圖4.30:支付服務商1 ──系統驗證頁面，使用指紋辨識 4.3.2系統基本功能 本研究之模擬支付服務商的系統中，以支付服務商1的首頁為例，可檢視當前登入的使用者名稱、帳號、餘額、二次身份註冊與驗證狀態。此外，也支援基本功能包括儲值、提領、轉帳，如圖4.31所示。 圖4.31:支付服務商1 ──基本功能4.3.3二次身份註冊與驗證二次身份註冊與驗證如同一般的系統註冊流程，使用者於中心化依賴方的首頁（圖4.32），點選註冊按鈕後，跳轉到二次身份註冊與驗證的註冊頁面，使用者可視情況調整預設的註冊選項，點選確認按鈕後，即啟動註冊流程。接著，使用者可選擇欲使用哪一種驗證器。最後，使用內建驗46 證器，以本研究之系統開發環境與工具為例，就會使用macbook的Touch ID指紋辨識作為驗證方式，即完成二次身份註冊與驗證的註冊流程。 圖4.32:支付服務商1 ──二次身份註冊與驗證頁面 4.3.4嚴格顧客驗證與跨機構轉帳使用者需先完成二次身份註冊與驗證後，才能使用跨機構轉帳的功能。之後，使用者會跳轉到中心化依賴方的頁面，可檢視該次交易的細節，例如:來源支付服務商、來源支付服務商的使用者帳號、目標支付服務商、目標支付服務商的使用者帳號、當前餘額、跨機構轉帳金額、備註，確認無誤後，即可進行嚴格顧客驗證（圖4.33)。下一步，使用者可選擇欲使用哪一種驗證器。接著，使用內建驗證器，以本研究之系統開發環境與工具為例，就會使用macbook的Touch ID指紋辨識作為驗證方式，即完成嚴格顧客驗證的流程。當完成嚴格顧客驗證後，中心化依賴方會繼續進行跨機構轉帳的功能。最後，可於支付服務商1與支付服務商2的首頁，檢視跨機構轉帳後，使用者的餘額。 圖4.33:中心化依賴方──確認交易細節後，嚴格顧客驗證與跨機構轉帳4.4系統評估與限制本章節將對這個模擬系統的功能，應遵循FIDO2、嚴格顧客驗證等各項規範，以評估本系統在實際應用中的可行性與研究貢獻，並討論目前面臨的限制與挑戰。 4.4.1系統評估 在FIDO2的實作中，本系統採用Touch ID進行使用者身份驗證，提供可靠、安全的身份驗證方式，運用每次系統註冊/驗證時，FIDO伺服器產生的隨機挑戰碼，能降低發生重放攻擊的可能性。同時，FIDO2讓使用者避免對傳統密碼系統的依賴，亦能減少發生密碼疲勞的問題。 此外，在Webauthn規範中關於RP ID與origin設定的要求與限制，RP ID必須是一個有效網域名稱，用來表示發起Webauthn註冊/驗證請求的依賴方，如此便能預防網路釣魚攻擊。此外，在驗證過程中必須使用HTTPS協定，確保傳輸過程中的資料都是經過加密傳輸的，能防止在資料傳輸過程中被竊聽或竄改。 關於嚴格顧客驗證的要求，當使用者欲使用跨機構轉的功能時，能在一個安全的執行環境中（例如：中心化依賴方透過CORS設定，控管支付服務商的白名單清單），透過持有物（例如: macbook Touch ID作為內建驗證器）、固有特徵（例如:指紋）作為驗證基礎，並在瀏覽器頁面上確認該次交易細節，並驗證綁定該次交易細節的挑戰碼（動態連結），來完成此次跨機構轉帳的流程。如此，便能滿足嚴格顧客驗證的要求，提升交易的安全性，並降低發生支付詐欺的可能性。 技術面上，本研究之模擬系統於中心化依賴方透過CORS設定白名單機制，能限制僅能透過支付服務商發起二次身份註冊與驗證、嚴格顧客驗證與跨機構轉的請求，強化支付服務商之間跨機構轉帳的交易安全;另外，本研究之模擬系統運用JWT token，二次身份註冊與驗證流程、嚴格顧客驗證與跨機構轉帳流程所傳遞的使用者資訊、交易細節未被竄改，預防中間人攻擊。 4.4.2系統限制 本研究之模擬系統是基於本地端作為開發、測試環境，因此運用子網域的方式來區分中心化依賴方、支付服務商1、支付服務商2的origin，以及OpenSSL自發證書來模擬正式環境的加密傳輸機制。然而在實際應用中，需使用有效的網域名稱，並且使用HTTPS協定來傳輸驗證資料，以滿足Webauthn協定的規範。 此外，Webauthn能支援許多生物辨識方法，因此在實際應用中應能提供更多樣的驗證方式，以支援更多客戶端設備與驗證器，例如:臉部辨識、虹膜辨識、聲紋辨識等。 承第2.2.3章節:嚴格顧客驗證所述，在實際應用中，需根據應用情境，考量使用者體驗的便利性、交易安全性。舉例來說，在小額支付、定期支付的交易時，可豁免強制導入嚴格顧客驗證的機制，以提供較流暢的使用者體驗;然而，若是遠距支付交易的情境下，並且達到一定的交易金額門檻時，需強制導入嚴格顧客驗證的機制，來確保交易安全性。因此，可根據實際應用情境，評估是否應強制導入嚴格顧客驗證機制。 第五章結論與未來研究方向 5.1結論 隨著資訊科技的日益進步，數位化時代的來臨，資安意識不斷提升促使許多企業開始重視相關技術的發展，而許多傳統服務也加速數位轉型，如何提供一個更安全的應用程式環境是個越來越重要的議題。為確保個人資料的隱私與安全，除了基本的法律規範，也需要搭配各種技術把安全層級不斷向上提升，從傳統密碼、簡訊動態密碼驗證到現在的FIDO，都是為了避免潛在資訊安全的風險。 比較現行支付服務商的二次金融支付工具驗證流程，與本研究之模擬系統的二次身份註冊與驗證流程、嚴格顧客驗證與跨機構轉帳流程，現行支付服務商的二次金融支付工具驗證流程，需手動輸入信用卡資訊，或是另外跳轉到其它網路銀行的頁面，進行身份認證;而本研究之模擬系統的二次身份註冊與驗證流程，透過FIDO2標準，讓使用者能藉由生物辨識的技術，使用私鑰進行身份驗證，而無需再手動輸入信用卡或銀行帳戶的相關資訊，能有效提升使用者體驗;此外，本研究之模擬系統的嚴格顧客驗證流程，亦能沿用該私鑰進行身份驗證，拓展私鑰作為身份驗證基礎的應用情境。本研究透過建立一個模擬支付服務商間交易的系統，旨在應用FIDO2標準來優化使用者身份驗證流程，以及強化交易安全性，為支付服務商提供了一個可行的解決方案。 5.2未來研究方向承第2.2.1章節:金融FIDO所述，截至目前（2024年）金管會已推動成立金融身份識別聯盟（F-FIDO），以我國金融體系之共通核驗識別「信物」—晶片金融卡作為金融FIDO註冊之身份核驗工具。也已透過「數位身份認證及授權」的主題式監理沙盒，提供一個受控的實驗環境，讓金融機構、科技公司及其他相關單位，能在此進行數位身份認證、授權技術的創新測試。此外，已允許金控擔任自體系金融FIDO Server提供其子公司運用，完成相關法規修正或發布。 本研究之模擬系統所設計的二次身份註冊與驗證流程，需先於中心化依賴方註冊第二組公私鑰，並可作為跨機構轉帳時，進行嚴格顧客驗證所使用。根據金管會的金融科技發展路徑圖2.0中的推動措施3-1.推出的金融FIDO V2計畫，預計於2025年6月開始運作跨體系之「金融FIDO驗證轉接中心」，串聯現行各體系之金融FIDO，在當事人同意之下，由該中心提供服務及合理運用，便於使用者在不同體系間進行身分核驗，提升身分核驗之互通性(金融監督管理委員會, 2023a)。綜上所述，未來可探討支付服務商是否也能擔任自體系金融FIDO Server，讓使用者能在原支付服務商進行FIDO身份驗證，並搭配「金融FIDO驗證轉接中心」串聯其它支付服務商，如此當使用者欲使用其它支付服務商的服務時，就不需要再進行身份驗證了，以簡化使用者身份驗證的流程，優化使用者體驗。 參考文獻 Apple Inc. (2023). Accessing keychain items with face id or touch id: Overview. https: //developer.apple.com/documentation/localauthentication/accessing_keychain_ items_with_face_id_or_touch_id Apple Inc. (2024). Accessing keychain items with face id or touch id. https://docs-assets. developer.apple.com/published/3c99bf9268/rendered2x-1654018513.png Caccavello, G., & Okay Inc. (2022). Open banking: Back to basics: What is strong cus- tomer authentication? https://www.openbankingexcellence.org/blog/back-to- basics-what-is-strong-customer-authentication FIDO Alliance. (2020a). Fido security reference: Introduction. https://fidoalliance.org/ specs/common-specs/fido-security-ref-v2.1-ps-20220523.html FIDOAlliance. (2020b).Fidouafarchitecturaloverview:Authenticatorregistration.https: //fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-overview-v1.2-ps- 20201020.html#authenticator-registration FIDOAlliance. (2020c).Fidouafarchitecturaloverview:Authentication.https://fidoalliance. org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-overview-v1.2-ps-20201020. html#authentication FIDO Alliance. (2020d). Fido uaf architecture. https://fidoalliance.org/specs/fido-uaf- v1.2-ps-20201020/img/fido-uaf-architecture.png FIDO Alliance. (2020e). Fido uaf protocol specification: Registration operation. https: //fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-protocol-v1.2-ps- 20201020.html#registration-operation FIDO Alliance. (2020f). Fido uaf protocol specification: Authentication operation. https: //fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-protocol-v1.2-ps-20201020.html#authentication-operation FIDO Alliance. (2020g).無密碼體驗(uaf標準). https://fidoalliance.org/wp-content/uploads/FIDO_UAF_Experience.png FIDO Alliance. (2021a). Fido security reference architecture. https://fidoalliance.org/specs/common-specs/img/fido-security-ref-architecture.png FIDO Alliance. (2021b). Fido2 graphic v2. https://fidoalliance.org/wp-content/uploads/ FIDO2-Graphic-v2.png Matthew Miller. (2024). Simplewebauthn. https://simplewebauthn.dev/ MDN Web Docs. (2023). Origin. https://developer.mozilla.org/en-US/docs/Glossary/ Origin Oracle Corporation. (2024). Mysql 8.4 reference manual: Xa transactions. https://dev. mysql.com/doc/refman/8.4/en/xa.html The European Parliament and the Council of the European Union. (2015). Directive of eu: Article 97. authentication. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/ ? uri=CELEX:32015L2366 W3C. (2023a). Web authentication: An api for accessing public key credentials level 3: Relying party identifier. https://www.w3.org/TR/webauthn-3/#relying-party- identifier W3C. (2023b). Web authentication: An api for accessing public key credentials level 3: Authenticator attachment modality. https://www.w3.org/TR/webauthn-3/#sctn- authenticator-attachment-modality W3C. (2023c). Web authentication: An api for accessing public key credentials level 3: User verification requirement enumeration. https://www.w3.org/TR/webauthn- 3/#enum-userVerificationRequirement—卡通MONEY. (2023).電支也能手機門號跨行轉帳囉！ipass money開通服務綁定: Ipass money綁定

手機門號轉帳操作方式. https://www.i-pass.com.tw/cht/ News/Detail/103300一卡通MONEY. (2024).金融驗證的操作步驟及常見問題. https://help2.line.me/ linepay_tw/android/categoryId/50003418/3/pc?lang=zh-Hant&country=TW&contentId=50010723兆豐銀行. (2024a). 「金融行動身分識別」（金融FIDO）.https://www.megabank. com.tw/digital-finance/fido/fido兆豐銀行. (2024b).兆豐實體ATM註冊流程. https://www.megabank.com.tw/digital- finance/cloud-page/image-with-title-item/atm-process兆豐銀行. (2024c).兆豐網路ATM註冊流程. https://www.megabank.com.tw/digital-finance/cloud-page/image-with-title-item/process兆豐銀行. (2024d).兆豐身份認證APP裝置綁定流程及使用教學. https://www.megabank.com.tw/digital-finance/cloud-page/image-with-title-item/instructions金融監督管理委員會. (2021). 「金融行動身分識別聯盟」正式成立，加速提升數位金融服務的安全與便利. https://www.fsc.gov.tw/ch/home.jsp?id=96&parentpath=0&mcustomize=news_view.jsp&dataserno=202106150002&dtable= News金融監督管理委員會. (2022).金管會-數位身分認證及授權:主題式監理沙盒及業務試辦之辦理近況. https://www.fsc.gov.tw/ch/home.jsp?id=96&parentpath=0,2&mcustomize=news_view.jsp&dataserno=202207220001&dtable=News金融監督管理委員會. (2023a).金融科技發展路徑圖2.0. https://www.fsc.gov.tw/ websitedowndoc?file=chfsc/202308161025340.pdf&filedisplay=é□□%E8□□ç§□æ□□ç□¼å±□è% B7¯å¾□å□□(2.0).pdf金融監督管理委員會. (2023b).金融科技發展路徑圖2.0-具體推動事項列表. https://www.fsc.gov.tw/websitedowndoc?file=chfsc/202308161026170.pdf& filedisplay=2.0å□·é«□æ□¨å□□ä%BA□é □å□□èi¨(2.0).pdf 附錄程式碼連結本研究之模擬系統的完整程式碼連結: https://github.com/Hans-Tsai/inter-psp-transfer程式碼架構說明• controller資料夾:伺服器端程式邏輯• database資料夾: MySQL資料庫的migration紀錄、Redis資料庫的連線設定• model資料夾:運用knex套件，實作ORM的內容• modules資料夾:實作一些會用到的工具函式• routers資料夾:路由規則• views資料夾:前端畫面設計• app.js:主應用程式入口• config.js:環境變數設定• knexfile.js:設定MySQL資料庫連線• openssl.conf:用來產生OpenSSL CSR文件的設定檔• package.json:記錄Node.js會用到的套件 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 40 41 42 43 44 45 47 48 49 50 51 52 53 54