實習進度報告

2020.08.19

蔡承翰

瑞波幣(XRP)

選擇瑞波幣作為研究對象的原因

- ✓ 專為金融機構間的跨國支付系統所設計
- ✓ 在公開帳本上被廣泛交易，交易市值僅次於比特幣和以太幣
- ✓ 官方提供技術資料完整
- ✓ 推出瑞波幣的機構 ripple 有可能參與聯準會即將推出的跨行實時支付系統 FedNow (目前只是傳言，但是 ripple 和聯準會頗有淵源，聯準會也公開稱讚過 ripple，參考以下網站和 ripple 官網)
- ✓ 日本 SBI 集團的 SBI Ripple Asia 公司致力於運用 RippleNet 提供的 DLT 技術發展亞洲的跨行支付系統

關於 ripple

- ✓ ripple 是最初推出公開交易的瑞波幣的公司，也負責公開帳本 XRP Ledger 的維護
- ✓ 在金融機構間提供實時交易解決方案 RippleNet (非公開項目)
- ✓ XRP Ledger 被設計為即使 ripple 不再干涉網路，仍舊可以由社群自行營運的去中心化支付系統

XRP Ledger 的使用

- ✓ 開源項目(GitHub)
- ✓ 由所有伺服器共同維護，官方提供的伺服器 API 為 rippled (C++)
- ✓ 官方提供由 JavaScript 撰寫的 API 來與帳本互動(網路上有人用 python 存取 API 功能並提供 python library)

關於 XRP Ledger

- ✓ 社群角色：Gateway (金融機構), Account (使用者), Server (營運者)
- ✓ Gateway 可以在鍊上發行虛擬資產，該虛擬資產被視為 Gateway 的負債，持有該資產的人可以向金融機構贖回實體資產
- ✓ Account 必須信任金融機構才能持有該金融機構發行的虛擬資產，使用者和該金融機構間必須有 trust line
- ✓ Trust line 代表一使用者對一金融機構的信任程度，使用者可設定對某一機構某一虛擬資產的持有上限(與他人交易來的資產不在此限)

- ✓ Server 共同維護健全的網路發展，透過 P2P 網路和共識機制來決定新的帳本(區塊)，通常 Gateway 具有較大的動機去維護系統的運行
- ✓ 交易費只能以作為原生貨幣的 XRP 支付，費用將不會分配給 server 而是直接銷毀
- ✓ 由 Gateway 所發行的資產可以進行互換，系統將搜尋對使用者最有利的兌換方式，也可以使用 XRP 作為過渡(bridge currency)
- ✓ XRP 的供給在創世區塊就已固定，不會再有多的 XRP 發行

XRP Ledger 支付系統

- ✓ Direct XRP payment：直接支付 XRP 到一帳戶，或是激活該地址
- ✓ Cross-currency payment：涉及非 XRP 的虛擬資產支付
  - 支付單一虛擬資產
  - 支付方付出一虛擬資產，接收方收到另一虛擬資產(可為 XRP)
  - 系統會自動進行成本最低的資產兌換(尋找 offer 或是以 XRP 作為過度貨幣)
- ✓ Check：形同真實支票，支付時並不會有資產變動，而是需要接收方在逾期前兌現
- ✓ Escrow：滿足特定條件才會執行的支付
  - 原理和之前在 NEM 看到的 HTCL 相同
  - 只支援 PREIMAGE-SHA-256
  - 允許 XRP 進行跨鍊交易
  - 必須公開形成條件雜湊值 preimage 才能執行交易
- ✓ Partial payment：發送方可設定最大的支付量，接收方所接收的金額為扣除交易費用和限制(如 trust line 或是發送方帳戶餘額不足)後的結果
- ✓ Payment channel：允許雙方建立非實時的交易管道
  - 建立起 payment channel 後，支付方可以透過 claim 支付金額
  - 關閉 payment channel 後支付金額才會同時交割
  - 為金額小但是量大的交易節省交易成本
- ✓ 交易成本
  - 最少 0.00001 XRP (10 drops)
  - 有些特定交易種類需要更多的交易費(Multi-signed transaction, EscrowFinish transaction, AccountDelete transaction)
  - Key Reset transaction 不須交易費

- Server 可以設定自己處理的交易的交易費下限(避免垃圾交易影響工作效率)

XRP Ledger 去中心化資產兌換

- ✓ Offer：對資產兌換的需求
  - 紀錄願意交換的金額(匯率)
  - 若逾期前被支付，資產兌換將被執行
- ✓ Rippling：若再不同機構發行同一資產上都有 trust lines，系統可能在兌換過程中對同一資產進行互換
  - 只發生在兌換過程中
  - 若不希望自己的帳戶被他人的兌換所進行 rippling，可在設定中取消
- ✓ 資產兌換時，系統會尋找成本最低的路徑(offer 或 rippling，也可能兩者都有)，或是使用 XRP 作為過渡貨幣

XRP Ledger 共識

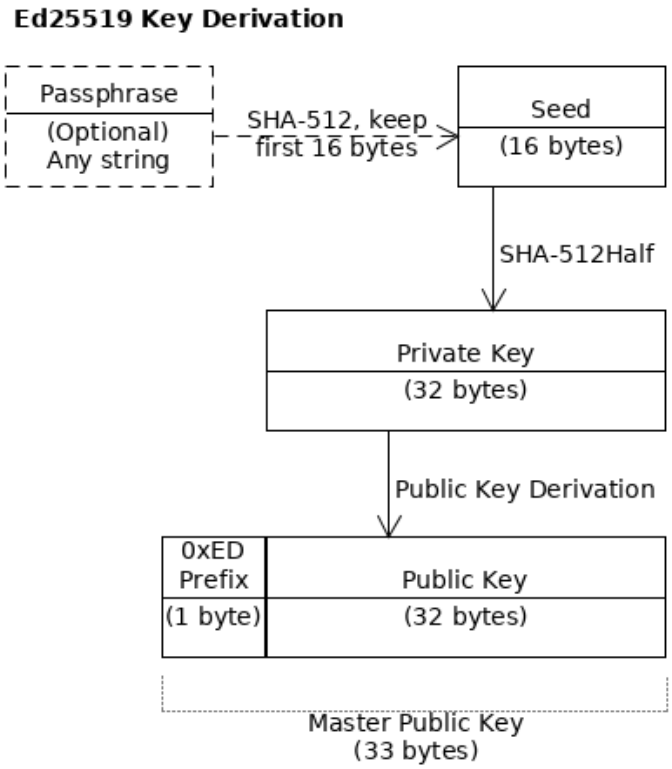
- ✓ BFT 共識機制
- ✓ 8 成以上誠實伺服器才能使系統順利運作
- ✓ 8 成以上惡意伺服器才能顛覆系統
- ✓ 伺服器必須選擇他所信任的其他伺服器(Unique Node List, UNL，官方有提供預設)
- ✓ 共識過程
  - 交易與數位簽章由客戶端送往被信任的伺服器，經過伺服器檢查後廣播至網路上
  - 每一伺服器各自先決定哪些交易被包含在新的帳本中，並和自己的 UNL 比對
  - 若 UNL 中大部分的節點都包含一筆交易，則將那筆交易包含；若大部分的節點不包含一筆交易，則將那筆交易剃除
  - 當網路對被包含的交易有共識時，每個節點將各自計算新的帳本
  - 節點將計算好的帳本雜湊值公開比對
  - 若 8 成以上的節點計算的雜湊值一樣，則共識完成，帳本內所有交易的結果確定(不論成功執行與否)，交易費被銷毀
- ✓ 未被包含的交易可能在下一輪共識中被包含
- ✓ 交易的排序並非依照時間，而是依照特別的規則
  - 支付較高的交易費將使交易更早被執行

XRP Ledger 帳戶

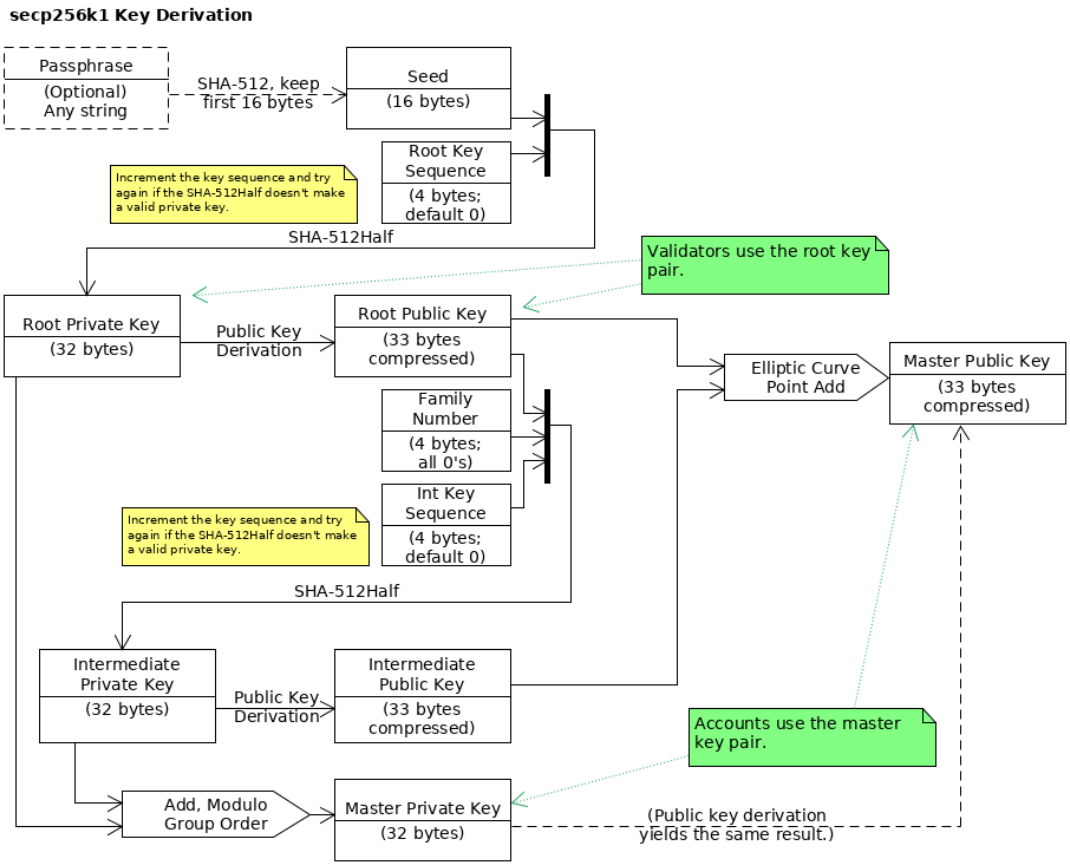
- ✓ 一個帳戶由一組 master key pair 所組成
- ✓ 生成公私鑰對後，當接收超過 20 XRP 的款項後帳戶才會被激活(被記錄在帳本裡)
- ✓ Reserve：帳戶中不能動用的金額(避免罐頭帳號)
  - 以 20 XRP 為標準，在帳本上多持有一個項目(Offer, Trust line, Escrow, Payment channel, check, Signer list) reserve 增加 5 XRP
  - 當 reserve 不夠時，帳戶將無法發出交易，直到接收金額使帳戶餘額超過 reserve 為止
- ✓ Master key 可以指定 regular key pair 作為送出交易時數位簽章的 signer
  - 當 regular key 妥協，master key 可以迅速變更 regular key 以避免更大損失
  - 官方建議 regular key 存在電腦上，master key 使用紙本保存

密碼學

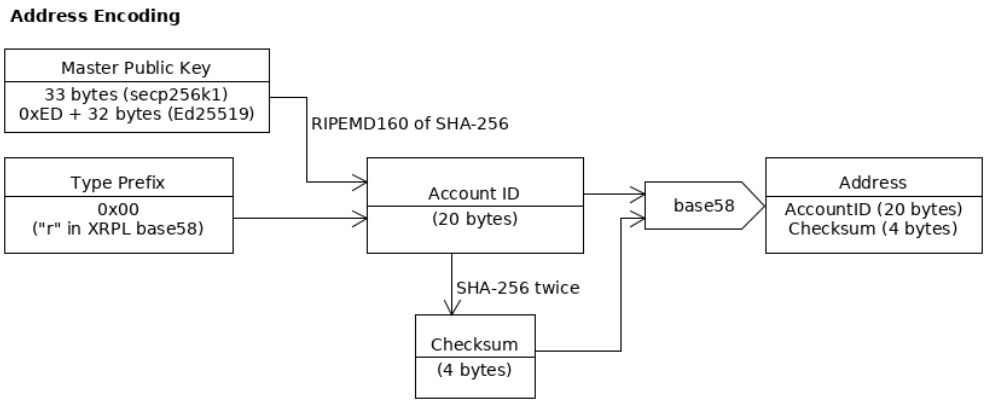
- ✓ 密鑰生成(圖片來源)
  - Ed25519 公私鑰對



- ECDSA(secp256k1)公私鑰對



✓ 地址(Address)生成(base 58)



✓ 應用到的演算法

- ECDSA secp256k1 or Ed25519 (擇一即可，預設為 ECDSA)
- SHA256 (在 Escrow 和密鑰生成時用到)
- SHA512 half (取 SHA512 的前 256 bits)
- RIPEMD160 (Address 生成)

冷錢包

關於 XRP 冷錢包的可能實作(參考 Ledger Nano S 以及其他市面上硬體錢包的功能)

- ✓ 硬體本身的功能
  - 密鑰生成
  - 保護私鑰
  - 對連線設備上的軟體傳送來的交易內容進行簽章、計算雜湊值後回傳
- ✓ 搭配連線設備(電腦、手機等裝置)上的軟體的功能
  - 形成正確的交易格式
  - 將附有簽章的交易傳送給伺服器
  - 追蹤交易的執行與帳戶餘額
  - 提供給使用者使用的 UI 介面
- ✓ 設計上的一些疑惑
  - 使用 ECDSA 或是 Ed25519?
  - 需要準備 regular key 嗎?
  - 需要支援甚麼樣的交易類型? (可能從一般 XRP 支付出發再慢慢擴展功能)
  - 交易傳送至哪一個伺服器? (可以使用官方的，但是官方並不保證他們運行的伺服器會持續運作)
  - 和連網設備的連接方式(USB, 藍芽)?
  - 密鑰的保管

其他

演算法

- ✓ SHA-2 演算法(SHA512 為主)
- ✓ AES C code (花不少時間在 MixColumn)

嵌入式系統設計

- ✓ Coursera 上由科羅拉多大學波德分校開設的兩門課程
  - Introduction to Embedded Systems and Development Environment
  - Embedded System and Hardware Architecture
  - 對嵌入式系統架構有基本認識

- 還不是很確定硬體該如何和主機上的軟體溝通

✓ 開發環境架設

- Ubuntu virtual box 和 WSL
- 相關編譯套件

未來計畫

Ripple API 使用方法研究

✓ Source code 因為過於龐大而且也不是用我熟悉的語言撰寫(也引入了很多外部 library)，不知道從何看起…

✓ 從研究 API 的使用開始

✓ 研究詳細的交易格式

✓ 學習 JSON 格式的傳輸

✓ RIPEMD160 演算法

拿到開發版後

✓ 實作 XRP 的冷錢包(確切功能尚須討論)