數位貨幣的電子錢包安全性探討-以比特幣為例

陳伯韋*[a]、王家輝[a]、何建明[b]
銘傳大學資訊工程學系[a]
中央研究院資訊科學所[b]

摘要—數位貨幣的資產儲存方式不同於傳統貨幣，傳統貨幣是將資產存入銀行，形成了貨幣的中央處理機構。以知名比特幣為例的數位貨幣並無類似中央的機構來存放資產，使用者可創建錢包私鑰以存放自己的比特幣，錢包地址的生成會依循特定的演算法隨機生成，而比特幣地址生成的過程中必須先生成一把 256 位元的私鑰，經過雜湊函數運算後，實際有效比特幣的地址為 160 位元，藉此保障了比特幣地址的基本安全性。比特幣錢包在使用 AES 加密演算法進行加密的過程中，需要手動輸入密碼及亂數生成主密鑰，進而生成主密鑰密文與私鑰密文。然而，攻擊者的目標可以縮小到破解用戶所設置之密碼，在進行暴力破解密文便可動用所有的電子錢包資金。於是，本論文便致力於探討如何改善電子錢包私鑰的安全性問題，進而改良與設計新的加密系統架構，並採用安全性較高的 RC6 加密演算法，以有效提升數位貨幣在網際網路上分散式儲存資產的安全保障。[1]

關鍵字：數位貨幣、比特幣、電子錢包、雜湊函數、AES 加密演算法, RC6 加密演算法。

一、 前言

比特幣（Bitcoin）[1]的發展顛覆了傳統的貨幣思維，人類從以物易物演進到以貝殼為貨幣的遠古年代，稱之為實物貨幣，之後更以黃金白銀為有價值的貨幣象徵，至今我們所見的法幣，如美金、日幣與台幣等，雖然該貨幣看似張紙，但卻有其價值，因為在背後，有著各國的央行以黃金為後盾以支持著該法幣的價值。基於科技的日新月異，資訊技術的革新，傳統貨幣也逐漸的電子化，而形成了新一代的電子貨幣（electronic money）[2]，傳統的電子貨幣為中央管制的形式，我們必須透過各大銀行的合作，才得以進行電子金流，在進行跨國匯款時，甚至會遇到匯率兌換以及稅賦問題，對每筆交易收取一定比例的手續費（handling fee）。

電子貨幣大致分為兩種，一為需要經過中央機構，才得以完成交易，而電子貨幣的第二種即為數位貨幣（Digital Currency）[3]，數位貨幣是以密碼學及點對點網路（peer-to-peer， P2P）[4]架構為基礎的電子貨幣系統，不同於傳統貨幣的是數位貨幣的模式沒有中央機構控管，故在進行交易時只需要低額的手續費，也因為它交易的不可逆性，可以避免交易紛爭。數位貨幣的種類很多，最為著名的包括比特幣（Bitcoin）、萊特幣（Litecoin）[5]、狗幣（Dogecoin）[6]及瑞波幣（Ripple coin）[7]，而比特幣只是數位貨幣的其中一種。

網際網路全球化帶來的不僅是便利，也隨即伴隨著網路攻擊事件層出不窮，使得比特幣錢包的安全性，更受到使用者的重視。在台灣也有比特幣交易所，因不堪駭客使用木馬程式滲入伺服器主機的攻擊，竊取該公司的比特幣錢包檔案，使得該公司破產。而官方的比特幣錢包又未執行錢包檔案的分離式儲存處裡，故使得破解錢包密鑰變得相當容易，此次事件不僅是公司的損失，也波及到大眾對比特幣的觀感，但實質上的比特幣是有其相當安全的防護機制。

比特幣的財產儲存方式不同於傳統貨幣，傳統貨幣是將資產存入銀行，形成了貨幣的中央機構，比特幣並無類似中央的機構來存放財產，使用者可以個別生成自己的錢包地址來創建自己的錢包，而錢包地址的生成會依循特定的演算法隨機生成，在比特幣地址生成的過程中必須先成一把私鑰，其私鑰的長度為 256 位元，所能表示的私鑰為 2^{256} 個私鑰，經過雜湊運算[8]後，實際有效的地址為 160 位元，也就是 2^{160}（相當於 2.15 x10^{38}）個雜湊值，雜湊值經由地址生成程序編碼成一個有效的比特幣地址。假設有 1E/每秒的運算能力（E=2^{60} 相當於當前比特幣網際網路分散式運算能力的 10 倍）進行破解。每秒碰撞該地址成功的機率：（1/2^{160}）x（2^{60}）/2=1/（2^{101}），可知即使有龐大的運算能力，比起進行窮舉法暴力破解地址會比進行挖礦來的沒有價值，也因此保障了比特幣地址在本質上的安全。

官方的比特幣錢包稱之為 Bitcoin Core[9]，因比特幣系統相當完整，便形成了 Bitcoin 協定[10]，比特幣網路、金額傳送以及餘額查詢皆已完整包含在 Bitcoin core 中，第一次使用比特幣錢包時會自動隨機生成私鑰，生成的私鑰會以 AES 加密演算法[11]加密，儲存在名為 wallet.dat 的檔案中，AES 加密演算法在進行加密的過程中，需要手動輸入密碼及亂數生成主密鑰，完成運算程序後會生成主密鑰密文與私鑰密文，此兩筆輸出結果將會同時儲存於 wallet.dat 檔案中，攻擊者的目標會縮小到使用者所設置之密碼，進行暴力破解便可動用所有的資金。本篇論文便致力於改善錢包私鑰的安全性問題，故改良與重新設計新的加密系統架構，以提升使用者的數位貨幣的資產安全性保障。

本論文將在第二節詳細介紹比特幣牽涉到的相關技術：比特幣基礎、比特幣的挖礦、電子錢包生成、電子錢包交易、區塊鏈、AES 與 RC6 加密技術；第三節將會提出比特幣電子錢包以 RC6 加密方法為基礎的改良方法的系統架構與流程，最後是結論與未來工作。

[1] 本研究由科技部計畫補助，計畫編號 MOST 104-2221-E-130-002 。

二、 相關技術

2.1 比特幣基礎

自 2009 年比特幣的誕生，開啟了數位貨幣的時代，數位貨幣不只是比特幣，也包括萊特幣、狗幣與暗黑幣（Darkcoin）[12]等，其差異在於發行總量、新幣發行週期以及發行日期。大部分的數位貨幣都有固定的發行量，也有少數的數位貨幣以無限量的方式發行，雖以無限量的形式生產貨幣，但隨著貨幣發行難易度（i.e. Difficulty）[13]的提升，其生產量也會逐年減半，故不會有通貨膨脹之問題。

對比特幣而言，總量只會發行 2100 萬個比特幣，發行的速度也依循既定的規則，開始發行貨幣後的第一個四年約每 10 分鐘會發行 50 個比特幣進入市場，而在下一個四年，變成約每 10 分鐘發行 25 個比特幣進入比特幣網路，以此類推，其發行量會越來越趨近於零，而總量也越接近 2100 萬個比特幣，於 2013 年也順利地完成了第一次的生產減半，這表示比特幣發行機制正日漸完備。

比特幣系統中想要獲得比特幣分為兩種，其一是以上所提的發行貨幣，第二種則是做交易的驗證（verify），也就是在每一筆的比特幣交易中，都會打包出一個交易封包，我們稱之為區塊（block）[14]，區塊中包含的訊息很多，包括時間、發送端比特幣地址、接收端比特幣地址與比特幣交易數量等資訊，把這些資訊打包起來形成區塊。而為了保證每一筆交易不被變更，引用了雜湊函數演算法 SHA-256 來確認每一個區塊不被竄改，也就是交易驗證。

新的區塊會被支付比特幣的人廣播（broadcast）至周圍的礦機（miner）[15]，進入由雜湊運算建構成的 P2P 比特幣網路進行驗證，其結果再次廣播至比特幣 P2P 網路進行核對，便可得到手續費；而新的區塊會直接疊在舊的區塊上，一直串連下去形成區塊鏈（blockchain）[16]，也因為在區塊中加入了時間戳記（timestamp）[17]的概念，區塊鏈便形成了一個絕對的時間軸，許多開發者正極力開發這不可逆的絕對時間軸所衍生的相關應用。

2.2 比特幣的挖礦

在生產比特幣的機制中，導入了工作量證明（proof of work）的機制，工作量證明為證明要求「請求資源者」進行特殊的數學運算，所以設計一道數學式要求請求資源者進行計算，若順利解出題目，伺服器端也可以快速驗證其答案之正確性，才可取得資源，此機制是為了避免攻擊者進行無謂的請求資源。比特幣的產出與區塊的生成是一起的，全世界的礦工會運行工作量證明的數學式以競賽的模式，比較誰會先算出標準答案，如答案正確則可取得記帳權，取得記帳權即代表著可創建新的區塊，並得到當次的比特幣產出，而這個行為則被稱之為挖礦。礦工在生區塊的同時也會將所有交易的手續費一併納入本次的收入。

為了得到這些新產出的比特幣，所使用的硬體裝置也逐漸的提升，從一開始的 CPU 挖礦到 GPU 挖礦，至今已經開發出各式各樣的特殊挖晶片，使之能有更快的運算速度，能夠快速的解決工作量證明的題目。

在挖礦運算能力爆發性成長的年代，能夠解出一題工作量證明的問題已經成為機率問題，也是有可能發生沒有收入的可能性，為使能有更穩定的收入，並發展出了新的挖礦型態，傳統的挖礦為個體挖礦，顧名思義為用個體的運算能力與全網的運算能力競爭，收入極為不穩定，因此發展出礦池的型態，在網路上創立一個平台，募集大家一起加入挖礦，將個別的運算能力合併，即可提升解決工作量證明的機會。礦池(pool)的所得，再以每位使用者的運算力為參考依據，進行分紅給每一個礦工。

2.3 電子錢包的生成

在探討比特幣官方錢包所生成的錢包檔案 wallet.dat 前，必須先清楚比特幣地址的生成過程，以下將逐一介紹比特幣地址的生成中，所使用到的加密演算法，包括橢圓曲線加密演算法[18]、SHA-256[19]、RIPEMD 160[20]以及 BASE 58 編碼[21]：

1). 橢圓曲線加密演算法

橢圓曲線密碼學（Elliptic curve cryptography，ECC）是以橢圓曲線數學作為公鑰密碼（public-key cryptography）的方法。橢圓曲線在密碼學中的使用是在 1985 年由 Neal Koblitz 和 Victor Miller 分別獨立提出的。

橢圓曲線加密演算法的優點在於比起相同功能的加密演算法，能以更短的密鑰達到相同等級的安全性，如以相同的長度與 RSA 加密演算法相比會得到更強的強度。但其缺點為在進行加密與解密時，所耗費之時間較長，而其公式如式（1）所示。

$$y^2 = x^3 + ax + b \qquad (1)$$

在比特幣地址生成的過程中，橢圓曲線加密演算法扮演著很重要的角色，為保障加密的強度，會以一個巨大的質數作為加密過程中的參數，也因為需求的不同，橢圓曲線加密演算法也訂定出不同的巨大質數作為加密的參數，在比特幣使用到的 secp256k1 所使用到的質數為 p = FFFFFFFF FFFFFFFF-FFFFFFFF-FFFFFFFF-FFFFFFFF-FFFFFFFF-FFFFFFFE-FFFFFC2F。起初地址的生成需亂數生成 256 位元 的私鑰，之後並以橢圓曲線加密演算法進行私鑰轉公鑰的動作，生成之公鑰會在比特幣交易中加以進行運用。

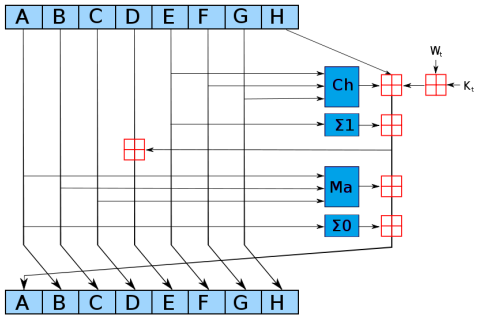
2). SHA-256

安全雜湊演算法（Secure Hash Algorithm，SHA）是一種能計算出一個數位訊息所對應到固定長度的字串的演算法。其特性為：

1. 從訊息雜湊結果反推其原輸入，在計算理論上為相當的困難。
2. 當輸入之訊息如有些許不同，便會得到與原輸出差異很大之結果
3. 若想得到以不同的輸入，取得相同的雜湊值，也是

很困難的。任何對輸入訊息的變動，都有很高的機率導致其產生的訊息摘要迥異。

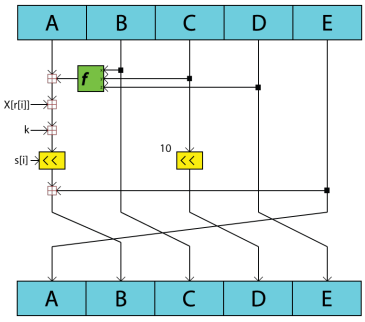
圖一為 SHA-256 雜湊函數的結構圖，在比特幣系統上的運用，SHA-256 不僅僅是使用在地址生成上，也應用在區塊鏈的鏈結上。在比特幣地址的生成過程中，會以 SHA-256 計算出雜湊值。比特幣地址在進行第一次發款之前，並不會透露出由橢圓曲線加密演算法計算出的地址公鑰，基於 SHA-256 不易被倒推的特性，得以提升比特幣地址在尚未進行第一次發款前的安全性。



圖一 SHA-256 架構圖[22]

3). RIPEMD160

RIPEMD（RACE Integrity Primitives Evaluation Message Digest，RACE），是 Hans Dobbertin 等 3 人在 MD4 與 MD5 的基礎上，於 1996 年提出。算法共有 4 個標準 128、160、256 和 320，其對應輸出長度分別為 16 位元組、20 位元組、32 位元組和 40 位元組。圖二為 RIPEMD 的架構圖：



圖二 RIPEMD 的架構圖[23]

數值	字元	數值	字元	數值	字元	數值	字元
0	1	15	G	30	X	45	n
1	2	16	H	31	Y	46	o
2	3	17	J	32	Z	47	p
3	4	18	K	33	a	48	q
4	5	19	L	34	b	49	r
5	6	20	M	35	c	50	s
6	7	21	N	36	d	51	t
7	8	22	P	37	e	52	u
8	9	23	Q	38	f	53	v
9	A	24	R	39	g	54	w
10	B	25	S	40	h	55	x
11	C	26	T	41	i	56	y
12	D	27	U	42	j	57	z
13	E	28	V	43	k		
14	F	29	W	44	m		

圖三 BASE58 編碼符號表

4). BASE58

BASE58 是專門為比特幣而設計的專屬編碼表，在比特幣地址生成的過程中為最後一個程序。BASE58 是從 BASE64 修訂而來，BASE58 的編碼表中不包括數字「0」，字母大寫「O」，字母大寫「I」，和字母小寫「l」以及「+」和「-」符號。

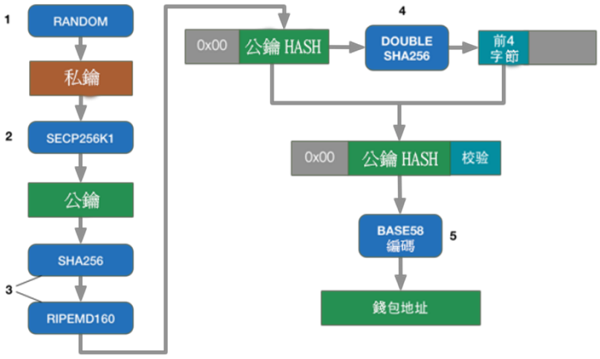
設計 BASE58（如圖三）主要的目的是：

1. 為了避免在閱讀抄寫比特幣地址時誤判。因為數字「0」和字母大寫「O」及字母大寫「I」和字母小寫「l」會非常相似。
2. 刪除「+」和「/」的符號進入 BASE58 中，是因為作為一個比特幣地址，若當中有非數字也非英文的元素在對使用者來說是很難以被接受的。
3. 大部分所使用的電腦程式，都支援以雙擊的方式全選比特幣地址。

錢包地址的建立是依照橢圓曲線演算法而生成，在理論上超過 2^{160} 個，所以不會有重複創建相同地址的問題，圖四為比特幣錢包地址生成流程圖。

比特幣錢包地址生成流程圖之說明：

1. 首先使用亂數產生器生成一個 256 位元私鑰。
2. 私鑰經過 SECP256K1 演算法處理生成了公鑰。
3. 公鑰經過 SHA-256 演算法處理生成了公鑰雜湊值，再將公鑰雜湊值經過 RIPEMD160 演算法處理生成了公鑰雜湊值。
4. 將一個位元組的位址版本號連接到公鑰雜湊值前端（在比特幣網路的公鑰位址，這一位元組為「0」），然後對其進行兩次 SHA-256 運算，將結果的前 4 位元組作為「公鑰雜湊值」之校驗值，連接在其尾端。
5. 將上一步結果使用 BASE58 進行編碼，就得到了「比特幣錢包地址」。



圖四 比特幣錢包地址生成流程圖

2.4 電子錢包交易

在比特幣網路中進行交易，使用者會利用自己的私鑰簽屬一筆交易，當簽署完成之後會將此交易廣播至比特幣網路，此時此比特幣礦池（pool）的記憶空間會緩存所有來自於世界各地的交易請求，這些交易請求的排序法會依照每一筆交易所付出的手續費作為參考權重重新排序，排序完之後便等待礦機進行打包的動作，將每一

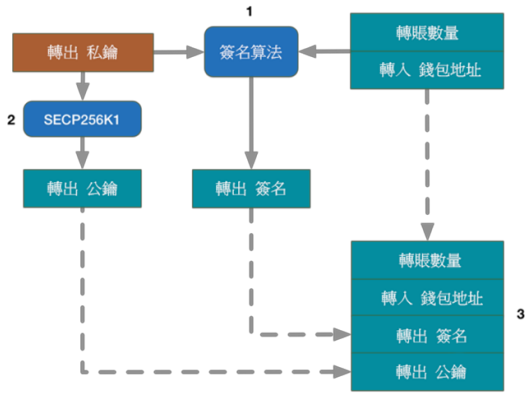
筆交易訊息集合成一個區塊，並與上一個區塊進行鏈結即生成了區塊鏈，交易紀錄都儲存在區塊鏈中。

比特幣的每一筆交易皆記載於區塊鏈帳單上，交易請求的訊息輸出及格式被訂定為比特幣核心錢包內的基本程序，付款方需要對交易進行數位簽章的簽署，得以證明其認同該項交易的許可。發送出去的比特幣會被記錄在接收方的比特幣地址上，交易的成立不需要接收方的參與，接收方即便沒有連上網路，也可以收取到發送端所發送的款項。在交易訊息中資金的來向稱之為「輸入」，資金去向則為「輸出」。如有輸入，輸入必須大於等於輸出值，輸入大於輸出的部分即為交易手續費。

但每筆交易有一輸入及一輸出的規則不是用於礦工開採比特幣的交易款項，礦工在比特幣網路中扮演著記帳的腳色，當礦工取得記帳權時，會得到一筆獎金作為獎勵，此獎勵在比特幣系統中可比擬為現實世界中的中央銀行以進行印製貨幣的行為，而礦工獲得的每筆發行資金是沒有輸入的，只有輸出。

交易的區塊被廣播至比特幣網路後，礦工會將交易打包記載至區塊中，完成打包至區塊鏈後，此時稱之為「1 個確認」。全世界的礦工平均約每 10 分鐘會生成一個區塊，每一個新區塊的產出，會使原交易的確認數加 1。當確認數達到 6 時，通常這筆交易會被認定為較安全且難以逆轉的交易。

比特幣錢包間的轉帳是通過交易實現的。交易數據是由轉出錢包「私鑰」的擁有者生成，如圖五所示：



圖五 私鑰對交易進行簽名流程圖

比特幣轉出錢包，以「私鑰」進行簽名流程圖的三大步驟說明：

1. 交易的原始數據包括「輸出金額」和「輸入的錢包地址」，但是僅有這些是不夠的，因為無法證明交易的生成者對「輸出的錢包地址」餘額有動用的權利。所以需要用「私鑰」對原始數據進行簽名。
2. 生成「轉出錢包公鑰」，這一過程與生成「錢包地址」的第 2 步是一樣的，見圖四。

將「轉出簽名」和「轉出公鑰」添加到原始交易數據中，生成了正式的交易數據，如此一來該數據就可以被廣播到比特幣網路進行轉帳了。

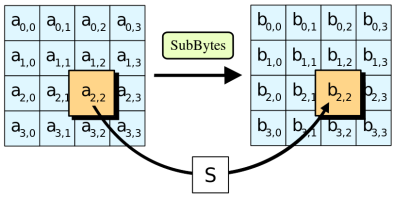
2.5 重要的區塊鏈

區塊鏈在比特幣網路中是一個重要概念，該概念在一名為中本聰在「一種點對點的電子現金系統」論文中提出，區塊鏈是一串使用 SHA-256 將在區塊與區塊進行雜湊運算，使區塊與區塊間有密不可分且不被竄改的特性。最新的區塊能夠依序鏈結到下一個區塊，也就是能夠鏈結到整個區塊鏈的底部。比特幣網路中會將所有的交易紀錄都儲存在「區塊鏈」中，因此可以將區塊鏈視為比特幣網路的帳本。區塊鏈的儲存是分散在 Bitcoin Core 的節點上，由所有節點組成分散式的資料庫，並以點對點的方式進行資料同步。

區塊鏈會記載著全世界從 2009 年開始運行至今的所有交易紀錄，而區塊鏈的成長是永無止境的，在 Bitcoin Core 中，會載入全部的區塊鏈，區塊累積至今已經超過了 50GB。為了降低區塊鏈同步造成使用者的負擔，便採用引入雜湊函數的機制。這樣用戶端將能夠刪除與自己不相關的交易紀錄區塊，以減少區塊鏈對使用者所佔用的記憶體空間。在中本聰創造了比特幣系統並開始運行所創造的第一個區塊，即稱為「創世區塊」，並在其區塊中附註一句「The Times 03/Jan/2009 Chancellor on brink of second bailout for banks」。

2.2 AES 加密

進階加密標準（Advanced Encryption Standard，AES），是一種高級加密標準，也叫 Rijndael 算法，2002 年制定為標準。它是用來取代原來的 DES 加密方法，並已經被多方分析且廣為全世界所使用。



圖六 AES 加密演算法架構[24]

AES 加密過程是在一個 4×4 的位元組矩陣上運作，這個矩陣又稱為「體（state）」，其初值就是一個明文區塊（矩陣中一個元素大小就是明文區塊中的一個位元組）。（Rijndael 加密法因支援更大的區塊，其矩陣行數可視情況增加）加密時，每回的 AES 加密迴圈（除最後一輪外）均包含 4 個步驟：

1. AddRoundKey—矩陣中的每一個位元組都與該次回合金鑰（round key）做 XOR 運算，且每個子金鑰由金鑰生成方案產生。
2. Substitute 位元組—透過一個非線性的替換函式，用尋找表的方式將每個位元組替換成對應的位元組。
3. ShiftRows—將矩陣中的每個橫列進行循環式移位。
4. MixColumns—為充分混合矩陣中各個直行的操作。這個步驟使用線性轉換來混合每行內的四個位元組。最後一個加密迴圈中省略 MixColumns 步驟，而以另一個 AddRoundKey 取代。

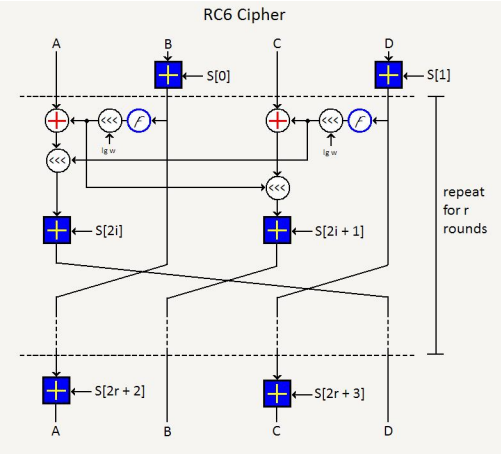
2.3 RC6 加密

根據 AES 的加密標準，一個分組加密必須處理 128 位輸入及輸出數據。儘管 RC6 的前身 RC5[25]是一個非常快的分組加密方法，但它處理 128 位元分組加密區塊時用了 2 個 64 位元工作暫存器；而 AES 目前在講究效率和簡潔前提下不支持 64 位元的操作，RC6 便修正 RC5 這個 64 位元處理的缺點，使用 4 個 32 位元暫存器而非 2 個 64 位元暫存器，以更周全地實現長位元資料的加解密。

RC6 延續了 RC5 簡單的設計架構、大量地使用數據相關的循環移位思想，同時增強了抵抗攻擊的能力，改進了 RC5 中循環移位的位數，不依賴於暫存器中可能有的位元數的不足。RC6 新的特色是輸入的明文由原先 2 個區塊擴展為 4 個，另外在運算方面則是使用了整數乘法，而整數乘法的使用在每一個運算回合中增加了擴散（diffusion）的行為，並且能即使少量的回合數也有高度安全性。

同時，RC6 中所用的操作可以在大部分處理器上高效率地實現，提高了加密速度，RC6 是一種安全、架構完整而且簡單的區塊加密法。它提供了較好的測試結果和參數方面有相當高的彈性。RC6 可以抵抗所有已知的攻擊，且能提供 AES 所要求的安全性，可以說是近幾年來相當優秀的一種加密法。

RC6 不再使用 2 個 64 位工作暫存器，而是用 4 個 32 位寄存器。這就使得在每次循環中要進行 2 次循環移位操作，讓更多的數據位元來決定循環次數。如圖七所示：RC6 把明文分別存在 4 個區塊 A、B、C、D，剛開始分別包含明文的初始值，加密運算後則為 4 個密文的輸出值。



圖七 RC6 加密演算法架構[26]

RC6 是參數變量的分組演算法，實際上是由有確定值的 3 個參數組成的加密演算法。一個特定的 RC6 可以表示為 RC6-*w*/*r*/*b*，3 個參數 *w*、*r* 和 *b* 分別代表字長、循環次數和密鑰長度，而 AES 中，*w*=32，*r*=20。RC6 的設計中，密鑰長度 *b* 為 128 位元（16 位元組）。RC6 用 4 個 *w* 位元的暫存器 *A*、*B*、*C*、*D* 來存放輸入的明文和輸出的密文。明文和密文的第一個位元組存放在 *A* 的最低位元組，經過加解密後，所得的明文和密文的最後一個位元組則存放在 *D* 的最高位元組。

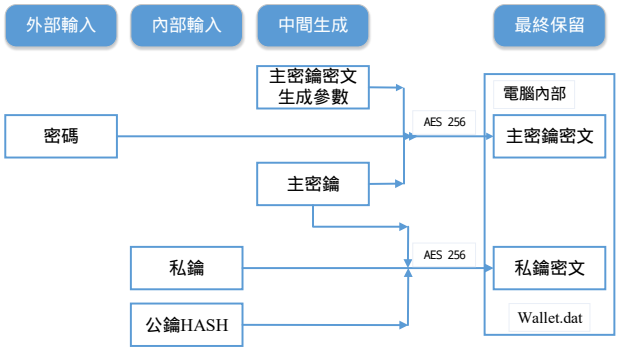
三、 比特幣電子錢包的安全性改良與系統架構

在既有 Bitcoin Core 比特幣錢包已有導入 AES 加密演算法將比特幣私鑰加密的程序，在其加密的過程中會有五個參數分別為密鑰、主密鑰、主密鑰密文、主密鑰密文生成參數、私鑰與私鑰密文參與。以下將逐一介紹各參數之用途：

A. 密碼：來源為外部輸入，即手動輸入密碼，用來加解密錢包私鑰的字串。
B. 主密鑰：為一個 256 位元的亂數，用於比特幣錢包的私鑰加密，加密完後立即刪除。
C. 主密鑰密文：為外部輸入「密碼」對「主密鑰」進行 AES-256-CBC 加密的結果，該加密為對稱式加密。
D. 主密鑰密文生成參數：主要是保存有「主密鑰」得到之「主密鑰密文」中參與運算的一些參數。由該參數配合「密碼」可以反推得到「主密鑰」。
E. 私鑰：橢圓曲線算法私有密鑰，即錢包中的核心，擁有私鑰就擁有私鑰對應的比特幣使用權，而私鑰對應的公鑰只是關聯比特幣，沒有比特幣的使用權限。
F. 私鑰密文：「主密鑰」對「私鑰」進行 AES-256-CBC 加密解密的結果，該加密為對稱加密。

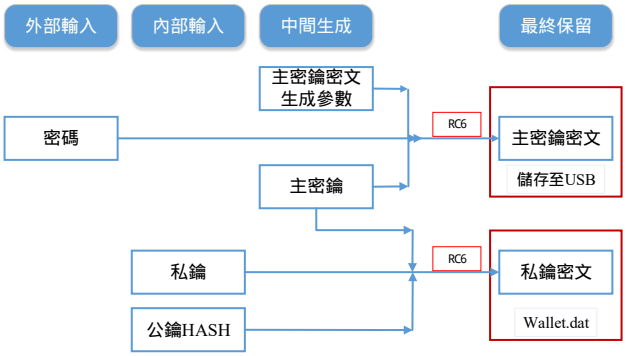
3.1. 官方錢包加密與解密

擁有比特幣地址的私鑰便代表擁有者可隨意簽署動用此私鑰所對應到的比特幣錢包地址中持有的所有資產，下圖為比特幣錢包 Bitcoin Core 使用 AES 加密演算法對比特幣地址的私鑰進行加密的過程：



圖八 官方錢包加密流程圖

1). 以亂數的方式生成 32 個位元的主密鑰，根據外部輸入的密碼結合生成主密鑰密文生成參數一起對主密鑰進行 AES-256-CBC 加密演算法，加密結果為主密鑰密文。
2). 將主密鑰對錢包內的私鑰透過 AES-256-CBC 加密演算法得到私鑰密文，加密完成後，並刪除原有的私鑰，保留私鑰密文；同時刪除主密鑰，保留主密鑰密文和主密鑰密文生成參數。



圖九 以 RC6 重新設計加密錢包流程圖

3.2. 以 RC6 重新設計的加密錢包

由於 3.1 中可知比特幣錢包的 Bitcoin Core 密鑰儲存方式是將主密鑰密文與私鑰密文共同儲存於 wallet.dat 的檔案中，若攻擊者竊取此檔案，並加以進行爆破攻擊，攻擊範圍僅侷限於使用者所設置之密碼範圍。因此本論文對既有的的加密過程提出修改，將主密鑰密文以及私鑰密文進行分離的方式儲存，才得以發揮 AES 加密演算法起初所設計密碼作用。為提升錢包加密的安全性並將 AES 加密演算法更改為 RC6 加密演算法。

延續官方錢包所使用的加密流程，因 RC6 可以抵抗所有已知的攻擊，並能夠提供 AES 所要求的安全性，並將 AES-256-CBC 加密演算法修改為 RC6，在完成第一步驟與第二步驟之後，並將所得到的主密鑰密文與私鑰密文，分開儲存於不同裝置中，以實踐 RC6 之安全強度。

根據 Milind Mathu 於 2013 年的論文[27] 可知要窮舉法的方式以 50 billion key/s 速度暴力破解所有可能出現的密鑰，AES 128 位元的密鑰 需要 10^{21} 年，而 RC6 192 位元的密鑰，則需要高達 10^{40} 年。

在替換加密演算法由 AES 加密演算法至 RC6 加密演算法後，不只使攻擊者進行暴力破解的困難度提升了許多，對使用者而言，RC6 之加密速度也比 AES 加密演算法快上 41%，因此在對使用者比特幣錢包加密解密的效率提升，使得錢包的使用上更加的有效與便利，如下圖十所示：

Input size in (Kbytes)	49	59	100	247	321	694	899	963	5345.28	7310.34	Average Time(ms)	Throughput (Megabytes /sec)
AES	56	38	90	112	164	210	258	208	1237	1366	374	4.174
RC6	41	24	60	77	109	123	162	125	695	756	217	7.19

圖十 AES, RC6 之加密速度比較表

結論與未來工作

在本論文中對比特幣錢包 Bitcoin Core 的修改，分為兩部分，一是將密鑰儲存分離，二是將加密演算法進行修改。在 Bitcoin Core 預設所適用的儲存方式進行修改，將主密鑰密文以及私鑰密文分開儲存於不同位置，甚至可以是不同裝置，以預防駭客對電腦上相關資料進行攻擊，竊取檔案後，可以直接對錢包檔案 wallet.dat 進行暴力破解，以至於失去當初使用 AES 加密演算法的初始意涵。而在分離之後，攻擊範圍不僅限於人類手動輸入之密碼，因為會有 AES 加密演算法中的主密鑰密文生成參數參與私鑰的加密，使密碼所生成的範圍會足足擴展至 2^{128} 個可能性，使攻擊者不只是要猜測手動輸日之密碼，也需要猜測密鑰密文生成參數曾能解得比特幣錢包之私鑰。

本論文不僅提升了比特幣錢包加密法 AES 加密演算法的實質加密成效，基於在眾多的加密演算法中，AES 加密演算法雖被美國聯邦政府採用，但 RC6 加密演算法所表現出的安全層級是比 AES 加密演算法還要高的，因此將比特幣錢包 Bitcoin Core 的加密法，由 AES 加密演算法修改為 RC6 加密演算法，密鑰生成的可能性提升至 2^{192} 個，在這密鑰生成的個數範圍，而比特幣有效地址的生成範圍為 2^{160} 個，RC6 加密演算法的密鑰生成範圍已經超越了比特幣有效地址的生成範圍的 2^{32} 倍，對攻擊者而言，進行經過 RC6 加密演算法處理過的錢包破解，所花費的時間，不如以高速電腦運算所有的比特幣私鑰來的有價值的多。

未來，除了將上述方法實作在開放原始碼的比特幣軟體模組上，更可以整合[28]有關重要資料檔案的防護機制，減少比特幣錢包檔案被竊取與進一步被破解的潛在危機。

參考文獻

[1] Nakamoto, Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org,2008
[2] Al-Laham, Al-Tarawneh, Abdallat,"Development of Electronic Money and Its Impact on the Central Bank Role and Monetary Policy",2009
[3] R Grinberg, Bitcoin: an innovative alternative digital currency,2012
[4] Biryukov, Alex; Khovratovich, Dmitry; Pustogarov, Ivan , "Deanonymisation of clients in Bitcoin P2P network",2014
[5] Litecoin, https://litecoin.org
[6] Dogecoin, http://dogecoin.com
[7] Ripple credits, https://ripple.com
[8] Hash function, https://en.wikipedia.org/wiki/Hash_function
[9] Bitcoin Core, https://github.com/bitcoin/bitcoin
[10] Bitcoin Protocol, https://en.bitcoin.it/wiki/Protocol_documentation
[11] Daemen, Joan/ Rijmen, Vincent, The Design of Rijndael: Aes-The Advanced Encryption Standard, 2002-04-01
[12] Darkcoin, https://www.dashpay.io
[13] Difficulty, https://en.bitcoin.it/wiki/Difficulty
[14] Block, https://en.bitcoin.it/wiki/Block
[15] Bitcoin Miner, https://en.bitcoin.it/wiki/Bitcoin_Miner
[16] Melanie Swan, Blockchain Blueprint for a New Economy ,2015
[17] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no 2, pages 99-111, 1991.
[18] Hazewinkel, Michiel, ed. (2001), "Elliptic curve", Encyclopedia of Mathematics, Springer,ISBN 978-1-55608-010-4
[19] Dmitry Khovratovich, Christian Rechberger and Alexandra Savelieva, "Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 family" ,2011
[20] Xuejia Lai; Hongbo Yu , "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD",2004
[21] The Base16, Base32, and Base64 Data Encodings. IETF. October 2006. RFC 4648. Retrieved March 18, 2010
[22] SHA-2, https://en.wikipedia.org/wiki/Secure_Hash_Algorithm

[23] RIPEMD, https://en.wikipedia.org/wiki/RIPEMD
[24] AES, https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
[25] RC5, https://en.wikipedia.org/wiki/RC5
[26] RC6, https://en.wikipedia.org/wiki/RC6
[27] Mathur, Milind, and Ayush Kesarwani. "Comparison between DES, 3DES, RC2, RC6, Blowfish And AES," Proceedings of National Conference on New Horizons in IT-NCNHIT. Vol. 3. 2013.
[28] 周立祥、廖唯志、王家輝、魏早達, "全方位電腦資料存取之安全控管系統的設計與實作," 2015資訊科技與實務研討會, 銘傳大學, 2015/3/13。