行政院國家科學委員會專題研究計畫　成果報告

系統晶片安全護衛

計畫類別：個別型計畫
計畫編號：NSC93-2215-E-009-030-
執行期間：93 年 01 月 01 日至 93 年 10 月 31 日
執行單位：國立交通大學資訊科學研究所

計畫主持人：何慎諾

計畫參與人員：蕭棨名，張正佳，薛仲佑，徐家駿

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 93 年 11 月 4 日

系統晶片安全護衛計畫

摘要

此計畫的目標是研究關於高安全性架構的矽智財模組。在將來的智慧型網路環境的，智慧型系統將會越來越需要有線或無線通訊。假定利用一個公共建設和開放的通訊頻道來傳輸資料，因為懷有惡意的罪犯日益的增加而對我們造成資料被竊取的威脅，所以具有相關功能的系統晶片(SoC)才會更需要安全模組來保護我們的資料。為解決此問題，已有基本的加密模組應用在系統晶片上。然而現有的商業用途模組設計的方式非常類似，以致於它們易於招受到外來的攻擊。此計畫以定義出整個系統晶片的安全架構和需求為目的，也考慮到提供比現有的架構更佳防護性之所有可行方案。這些對於使用者是否信任他們所使用裝置的安全性是很重要的考量。在文中我們說明此方法的概念，也作了此方法之架構設計，以增加本方法之可行性。

一、 背景

此計畫的目標是研究關於高安全性架構的矽智財模組。在將來的智慧型網路環境的，智慧型系統將會越來越需要有線或無線通訊。假定利用一個公共建設和開放的通訊頻道來傳輸資料，因為懷有惡意的罪犯日益的增加而對我們造成資料被竊取的威脅，所以具有相關功能的系統晶片（SoC）才會更需要安全模組來保護我們的資料。

為解決此問題，已有基本的加密模組，如數位加密標準（DES)、先進加密標準（AES）應用在系統晶片上。然而現有的商業用途模組設計的方式非常類似，且大多數人仍然使用安全性較低的密碼防護。而用戶所使用的密碼過於較簡單，可能成為資訊安全的漏洞。以致於它們易於招受到外來的攻擊。

此計畫以定義出整個系統晶片的安全架構和需求為目的，也考慮到提供比現有的架構更佳防護性之所有可行方案。這些對於使用者是否信任他們所使用裝置的安全性是很重要的考量。因此我們提出此一套安全方法的概念，也作了此方法之架構設計，以增加本方法之可行性。

二、 簡介

1. SET 協定（Secure Electronic Transaction）

由 Visa 國際組織、Mastercard 國際組織、IBM、Microsoft、Netscape、Verisign、美國運通銀行等公司所共同支持的安全加密模式稱為 SET 協定（Secure Electronic Transaction）其特性如下所示：

a. 除了保障資料在傳輸過程的安全之外，更透過多方認證的機制，同時把銀行也一起納入交易流程之中進行規範。

b. 使用 SET 協定進行網路交易時，當你在結帳時所輸入的「信用卡」資料，不但不怕被中途擷取，而且也不用擔心被盜用，因為這個資料是直接傳送到銀行的，商家根本無法得知，而且銀行除了信用卡資料之外，對其它顧客商店間的詳細營業資料也無從得知？因為這些資料的傳送是不須經過銀行而直接由顧客端傳送到商店端的，另外顧客在交易付款時身份已經過銀行確認過，除非商品發生瑕疵等的其它因素，否則顧客也不能以可能遭到盜刷為由而拒絕付款，這樣對商店的業務機密及銷售權益也就受到了多一層的保障了。

其安全性如下所示：

a. 訊息隱密性（Confidentiality）：

將資料亂碼所使用的隨機亂數金鑰(DES key)及持卡人卡號等重要資料以接收方的 RSA 1024 bits 公開金鑰亂碼保護，再使用此 DES key 以亂碼方式保護其他的交易相關資料；Non-SET 為以 Triple DES 保護用戶的重要資料，亦可以數位信封(Digital Envelope)的亂碼方式保護，訊息不易被竊知。

b. 訊息完整性（Integrity）：

於網際網路上傳輸的任何訊息皆以雜湊函數（SHA-1）產生訊息摘要，再以訊息發送者的私密金鑰（Private key）對摘要後的訊息執行數位簽章（Digital Signature）保護，故訊息不會被非法篡改。

c. 身分辨識性（Authentication）：

每一參與交易者皆需向公正可信賴的第三者取得合法的交易憑證，以確認其身分，故交易資料不會被冒名傳送。

d． 交易不可否認性（Non-Repudiation）：

私密金鑰僅有傳送方才擁有，因此以該私密金鑰對傳送訊息產生的數位簽章，只要留存每筆包含電子簽章的交易紀錄，則交易的收送雙方均不能否認已傳輸的交易。

2. 爪哇卡（Java Card）

隨著我國經濟的發展和金融現代化，IC 卡算是卡片家族中最晚出道的，但卻是目前最有前途的一種卡片，最終它將會取代目前所有的條碼卡、磁卡的地位而成為卡片的明日之星。IC 卡是利用晶片來儲存資料，相較於磁條卡，具備較大的容量、較佳的防偽及保密功能，最佳的產品代表即是行動電話用的 SIM 卡，另一個則是中華電信前一波標案的公用電話晶片卡，甚至信用卡也將全面汰換之晶片卡，皆屬 IC 卡的範疇。而智慧卡為 IC 卡中功能最大並有系統之概念。

爪哇卡（java card）為智慧卡的其中一種不但可以儲存大量資訊，更具有極強的保密性，抗干擾、無磨損、壽命長，目前正廣泛的應用於金融、電信、運輸、教育、旅遊、政府、零售、健保、電子商務等各行各業。

- 符合 ISO7816 規範的一種 IC 卡， 通常在智慧卡上面有 uP 與 O/S， 主機可以透過標準定義的程序來存取它。例如 GSM SIM Card 與衛星電視收視卡都是智慧卡的一種。

- 智慧卡定義卡中各個目錄與檔案的存取權限，限制了主機存取卡上物件的權力，某些物件需要有密碼才能存取，有些則完全不能存取。如秘鑰（private key）、GSM Ki， 則需要透過 Challenge-Response 方式 由主機發出一問題（Challenge），然後智慧卡在運算完畢之後傳回。透過這種方式可以避免通訊過程中遭受竊聽跟破解。

- 智慧卡仍然存在數種可能的破解方法， 如：針測式、軟體式、竊聽式、試誤法、電流分析 等。

3. μC/OS-II

μC/OS-II 是建構在 uCOS 基礎之上的一個即時作業系統核心，是在 1992 首次被發表出來。目前在市面上已有廣泛的應用，從攝影機、醫學儀器、樂器、引擎控制器、網路配接卡、高速公路電話到工業機器人等等，並且在世界上許多個學校也都拿來當做即時系統的主要教科書。

μC/OS-II 跟其它分時作業系統不同，不支援時間片段法（time slice)。它是一個基於優先權的即時作業系統。每一個任務的優先權必須不同，分析它的源碼會發現，μC/OS-II 把任務的優先順序當作任務在標識來使用，如果優先順序相同則任務將無法區分。而進入就緒態優先權最高的任務首先得到處理器的使用權，若要執行其它任務，只有等到它交出 CPU 的使用權。

μC/OS 具有向上相容性，意指由前一代的程式碼到下一代仍可運行，只需經過一些設定上的更改。另外它亦提供了固定大小記憶體管理（fixed-sized memory manager)、使用者任務自建（user definable callouts on task creation)、任務刪除（task deletion)、任務切換（task switching)、系統時間記號（system tick)， 任務擴充（TCB extensions)，堆疊檢查（stack checking）等功能。它擁有以下幾種特性：

a. 程式碼完全公開（Open Source）

b. 可攜性（Portable）

c. 容量適用於唯讀記憶體（ROMable）

d. 可變動性（Scalable）

e. 可搶先性（Preemptive）

f. 可決定性（Deterministic）

μC/OS-II 對共用的資源提供了一些保護的機制。可將一完整程式分成幾個任務，不同的任務執行不同的功能。對於共用資源也提供了很好的解決辦法。一般情況下使用的是信號。我們建立一個信號並對它進行初始化，當一個任務需要與其它任務共用資源時，它必須先得到這個信號。在這個同時即使有優先權更高的任務進入了就緒態，因為無法得到信號，也不能用該資源。在 μC/OS-II 中稱為優先順序反轉。簡單地說，就是高優先權任務必須等待低優先權任務的完成。在上述情況下，在兩個任務之間發生優先權反轉是無法避免的。所以在使用時，必須對所開發的系統有清楚的了解才能選擇對於某種共用資源是否使用信號。

4. JINI

Jini 是一種以爪哇技術（java technology）為基礎用來建構網路分散式系統的一套中介軟體（middleware），它利用網路配接（sockets）的分散式計算技術跟遠端函式呼叫（remote method invocation）以達到在網路上擁用隨插即用的功能（network plug and work），新的服務能夠快速地加入組織（federation）並立即地提供服務，而用戶端也能夠很快速地搜尋到此組織下有提供何種服務並使用它以滿足其需求。其特色如下：

a. 隨插即用（network plug and work）
b. 無軟硬體特性的區別（erase hardware software distinction）
c. 擁用自動化網路連結功能（enable spontaneous networking）
d. 以服務為基礎的架構（promote service-based architecture）
e. 簡單易使用（simplicity）

一個 Jini 系統或組織是由一群用戶端和服務端依照 Jini 的通訊協定連結而組成的。雖然一個 Jini 系統通常全部是由爪哇語言（pure java language）實現出來，但是用戶端或者是服端並不一定要用 java 語用實現，也可以用原生碼（native code methods），甚至是由其它的語言實作在一起，所以 Jini 是一個連結不同來源的用戶端跟服務端之中介軟體層（middleware layer）。如同圖一所示，以下為 Jini 主要的構成元件：

a. 基礎建設（infrastructure）
    i. 遠端函式呼叫之整合式分散式系統（distributed system integrated into remote method invocation）
    ii. 搜尋通訊協定（discovery protocol）跟連結通訊協定（join protocol）
    iii. 查詢公共服務（look up services）

b. 程式模式（programming model）
    i. 資源配置或釋放之租賃介面（leasing interface）
    ii. 事件通知介面（event and notification interface）
    iii. 交易執行介面（transaction interface）

c. 服務（services）
    i. 列印服務（printing service）
    ii. 爪哇空間管理服務（java-spaces service）
    iii. 交易執行監督（transaction manager）

	Infrastructure	Programming Model	Services
Base Java	Java VM	Java APIs	JNDI
	RMI	JavaBeans	Enterprise Beans
	Java Security	...	JTS
Java + Jini	Discovery/Join	Leasing	Printing
	Distributed Security	Transactions	Transaction Manager
	Lookup	Events	JavaSpaces Service

圖一、Jini Architecture Segmentation

安全性：

Jini 技術之安全模式的設計方式是建置在兩個息息相關的概念上，即委託人（principal）跟存取控制名冊（access control list)。委託人在安全上的功能是代表某些實體存取 Jini 組織的服務，舉例來說在某些時侯服務端本身也會需要存取其它的服務端，而這種情況就是依照委託人此安全概念來實現的。而存取控制名冊在安全上的功能即當有用戶端想存取資源時，必須依照存取名冊的規定來行動，不可採取違反存取名冊的行為。

5. 演算法

祕密金鑰（Secret Key）密碼系統，亦稱作對稱金鑰密碼系統，其名稱「對稱金鑰」的由來，是因為這類型密碼系統是使用相同的金鑰，來進行加密與解密的作業。雖然對稱密碼系統擁有作業速度快等等的優點，但是其主要的缺點就在於：金鑰的傳輸過程必須是絕對安全的。因此，金鑰的傳輸勢必不能使用和密文傳輸相同的管道（例如網際網路等等），否則，攻擊者只需要在傳輸過程中能夠竊取金鑰及密文，再加上本身就已經是公開的加解密演算法，整個祕密金鑰密碼系統將遭到破解，當然加密也就變成毫無作用的工作了。

為了避免傳送金鑰時必須透過一個絕對安全的通道這個需求，密碼學大師*Diffie*與*Hellman*在1976年提出了公開金鑰（Public Key）密碼學的概念。這個概念為密碼學的研究開闢了一個新的方向。此後陸續有許多學者提出了各種不同的公開金鑰密碼系統及其不同的應用。

公開金鑰密碼系統，和祕密金鑰密碼系統類似地，在加解密的過程中也需要用到外加的金鑰。兩者間不同的是，在公開金鑰密碼系統中，加密金鑰和解密金鑰兩者並不相同。在公開金鑰密碼系統中，除了擁有不相同的加密金鑰和解密金鑰這個特性之外，在加密金鑰與解密金鑰之間也存在一個複雜的數學關係，而這個關係必須複雜到無法由加密金鑰推導出解密金鑰。假設傳送方想要讓接收方能安全地傳送某些資料，運用公開金鑰密碼技術，接收方此時將其加密金鑰公開，讓傳送方取得此加密金鑰。之後，傳送方就可以將想要傳送給接收方的資料，即明文（Plaintext），使用此加密金鑰進行加密後再傳送給接收方。當接收方欲解讀密文（Ciphertext）的時候，僅需使用相對的解密金鑰將密文解密，接收方就可以將密文還原為明文，得到祕密傳輸的資訊。

倘若在訊息在網路上傳輸的過程中，有一個第三者想要竊取此一祕密資訊時，那麼我們假設這個第三者將訊息經過整理與分析後，可能取得的資訊如下：

a. 在接收方公開加密金鑰的過程中，竊視者能取得加密金鑰。

b. 而在傳送方完成加密，傳輸密文時，竊視者能取得密文。

c. 已公開的加解密的演算法。

這時即使這個第三者竊得了加密金鑰，但是他仍然不能由所得的密文解出明文來。而由於加密金鑰與解密金鑰間有著複雜的數學關係，因此該竊聽者無法（或者說以現在的科技而言是非常的困難）藉由取得傳送方的加密金鑰、密文等等資訊而推導出接收方的解密金鑰。因此，雖然該竊聽者能夠得到這些資訊，但是他卻沒有辦法根據這些資訊得到接收方的解密金鑰及明文，這就是公開金鑰密碼系統最基本的應用。

另外值得注意的一點是：接收方傳送加密金鑰的過程是「公開」的。換句話說，這個加密金鑰讓他人知道並不會對整個加密系統的安全性有任何影響。這個特性克服了祕密金鑰密碼系統中，通訊雙方所使用的金鑰必須保密的缺點。目前使用得最為廣泛的公開金鑰密碼技術是Rivest、Shamir、Adleman三位學者在1978年所發表的RSA公開金鑰密碼系統，其中 RSA 為三位學者的姓名縮寫。以下，我們將介紹公開金鑰密碼系統的基本應用。RSA公開金鑰密碼技術已經成為業界標準。

RSA公開金鑰密碼技術的加解密過程如下：

a. 隨機選擇兩個大的質數 p、q

b. 計算 $n = p \times q$

c. 隨機選擇加密金鑰 e， e必須滿足下列條件

GCD(e, ψ(n)) = 1，其中ψ為Eulerψ，ψ= (p-1)ˣ(q-1)

d. 利用Extended Euclean Algorithm計算解密金鑰

$$d = e^{-1} \bmod \phi(n)$$

e. 將數值 p、q及解密金鑰 d 保密

f. 公開數值 n 及加密金鑰 e ，讓傳送方能夠取得，並將明文 M 加密成為密文 C 。

$$C = M^{e} (\bmod n)$$

g. 接收方收得密文C，利用解密金鑰d將其解密。解密過程為：

$$C = M^{d} (\bmod n)$$

運用RSA公開金鑰密碼技術演算法，傳送方利用接收方的加密金鑰 e 將明文 M 加密成為密文 C 後傳送給接收方，而接收方可以利用解密金鑰 d 將密文解密。

在本研究中將資料亂碼所使用的隨機亂數金鑰(AES key)及持卡人卡號等重要資料以接收方的 RSA 1024 bits 公開金鑰亂碼保護，再使用此 AES key 以亂碼方式保護其他的交易相關資料。

AES演算法的優點：

a. AES演算法可在個人電腦上快速地執行

b. AES演算法可在智慧卡上有效率地執行

c. AES演算法的回合運算可平行處理，若AES演算法的明文區塊大小為192位元或256位元，則AES演算法可架構為一具有抗拒碰撞（Collision-Resistant）特性的雜湊函數（Hash Function），在這其中，AES演算法被當作一壓縮元件。

d. AES演算法的設計允許明文區塊及金鑰大小由128位元開始，以32位元為單位，變化至256位元。

e. 雖然在文件中，AES演算法的運算回合數是固定的，但也可視需求加以變化。

AES演算法的限制：

a. AES演算法的解密元件較不合適在智慧卡上實作，因為這將花費較多的程式碼與運算時間。

b. 利用軟體實作AES演算法，加密與解密需要不同的程式碼。

c. 利用硬體實作AES演算法，解密電路並非全部都能重複使用。

安全性：

2000 年 10 月，NIST(美國國家標準和技術協會)宣布通過從 15 種侯選算法中選出的一項新的密匙加密標準。新的標準將會代替密匙長度變的太短的舊的 DES 演算法。AES 與目前使用廣泛的加密演算法—DES 演算法的差別在於，如果一秒可以解 DES，則仍需要花費 1490000 億年才可破解 AES，由此可知 AES 的安全性。

RSA 認證裝置特性：

a. 一次性密碼(one-time password)：

可能是較安全的防護。由於一次性密碼只使用一次，即使被駭客得知下次也無法使用。

b. 雙因素認證(two-factor authentication)：

目前國內已有許多金融業，如銀行正在導入藉由令牌(token)與認證伺服器組成的雙因素認證(two-factor authentication)或利用簡訊將一次性密碼傳到使用者行動電話，提供消費者登入網路銀行使用。在雙因素身分認證的過程中，使用者被要求輸入使用者名稱及一個替代密碼， 此替代密碼即為由他/她所持有的 RSA 認證裝置所顯示出當時的 token code， 加上一個識別碼(PIN)．代理程式混和了使用者輸入的資訊以及只有此受保護的裝置所知道的資料然後將此資訊傳送給 RSA ACE/Server， 當此資料證實為有效時則此存取即被允許．RSA ACE/Server 則會按照其紀錄檔， 讓該名使用者依其認證權限的層級來做存取。

c. 簡單使用：

RSA 認證憑證簡單到只要輸入一個密碼，但是卻更安全。每個終端使用者被指定一個 RSA 認證憑證，是以每 60 秒產生唯一一個新的不可預測的碼。當簽入網路時，使用者鍵入此數字再加上一個 PIN 碼或片語。這樣所形成的密碼，使得 RSA ACE/Server 能夠成功地辨明他以允許進入網路。RSA 裝置認證只需要非常少的步驟並且在幾秒之內完成認證。其安全性如下所示：

a. 防止駭客：

每個認證令牌都有一個唯一的 64 位元的種子值(seed value)，用來結合強

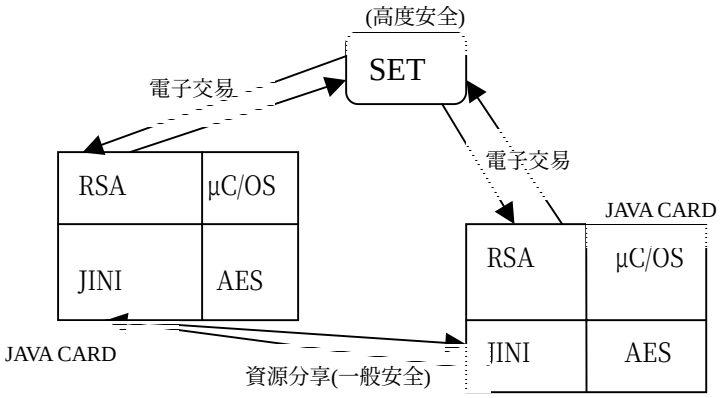
力的數學以在每 60 秒產生一個新的密碼。對認證令牌的組合來說，只有 RSA ACE/Server 知道在那一時刻的數字。因為這數字是不可預測的，對駭客來說，想要在任何時間猜對數字是不可能的。專利的時間同步技術的每個認證憑證和安全伺服器保證更高階的安全。此外，硬體和軟體兩者設計成阻力調節。

b. 安全性較高：

由於 Token 與伺服器同步，每 60 秒產生一組新的 Tokencode，所以除非使用者名稱、識別碼及 Token 卡同時被竊取，否則安全性相當高。相對的 Token 的費用較為昂貴，Token 卡加上認證伺服器，平均一個使用者的成本約三千多元。目前多應用於安全性要求較高的金融產業，寶來證券即提供部分成交量較高的網路下單客戶 Token 機制，以保障電子交易的安全性。

三、 架構

本方法的主要概念即在現有之安全交易 SET 協定（Secure Electronic Transaction）上使用 Java card 運行μC/OS-II 即時作業系統並以 Jini 為中介軟體（middleware）能夠自動建立網路連結。另外，直接將先進加密標準（AES）演算法以硬體方式包含在 Java card 內以保護資料安全。也把非對稱密碼法(RSA)用硬體的方式安裝在 Java card 中，作為使用者認證憑證以確保使用者身份。利用現有安全概念之作業系統、軟體、加密技術以及交易環境來實現一個安全的安全系統晶片架構(system on a chip architecture)成為一個安全的智慧型網路分散式環境。



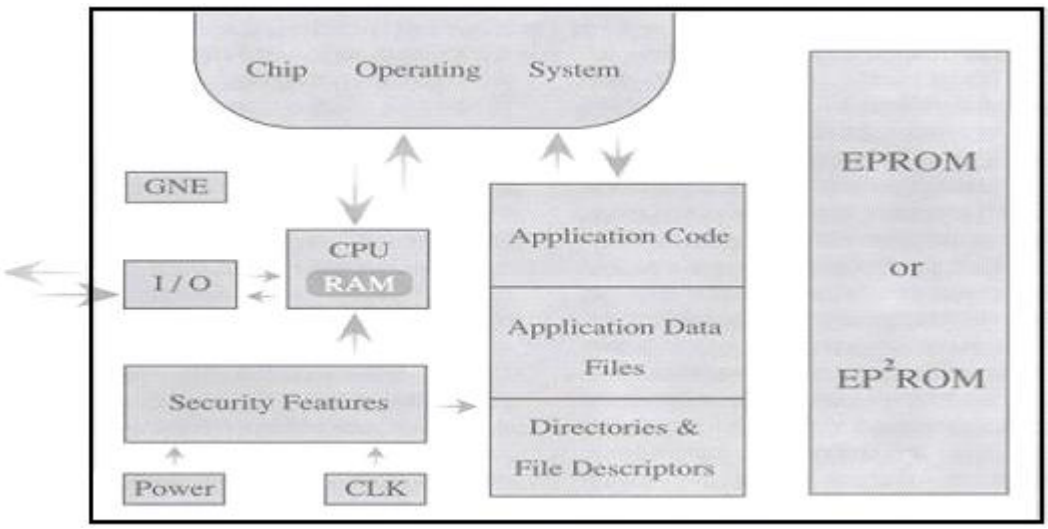
系統晶片安全護衛架構圖

圖二、架構圖

此計劃的架構設計分為三個項目：

1. 設計系統內部架構

晶片內除 CPU 外，內建 AES、RSA 硬體電路在內。安裝µC/OS-II 做為即時作業系統。以 AES 演算法加密所要傳輸的資料，RSA 則用來判別使用者的身份，使用硬體電路可以加速運算過程並且有效利用晶片。而µC/OS-II 則為一具備即時性之安全作業系統。



圖三、符合 ISO7816 規範之爪哇卡規格

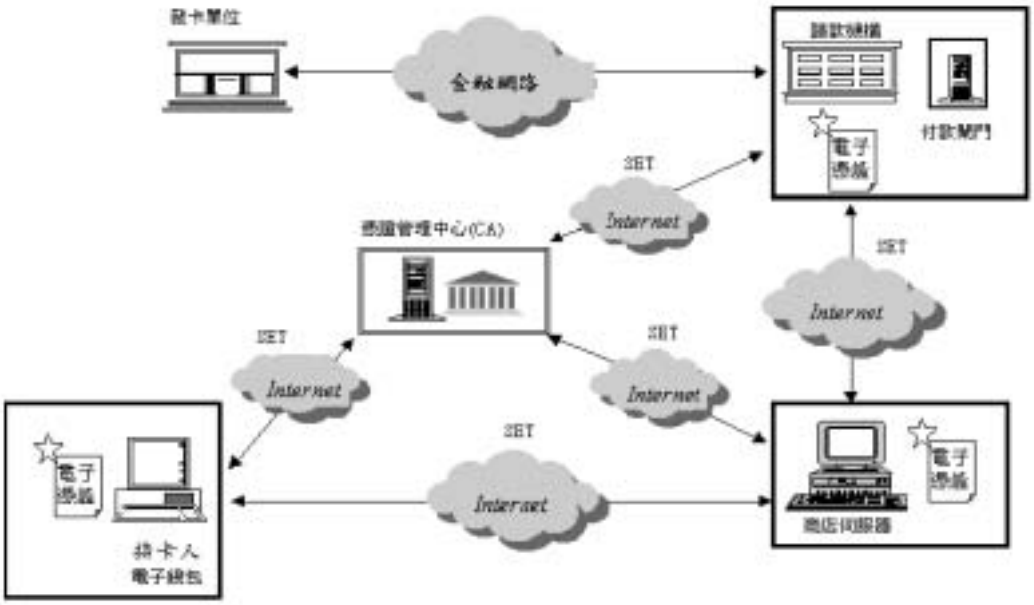
2． 一般安全性資源共用平台

一般安全性資源共用平台主要是針對如何達到快速共用資源、快速交換資料情況下又能確保一定的安全性。在此利用 Jini 的委託人(principal)跟存取控制名冊(access control list)的技術實現具備一定安全程度的資源共用平台。

首先安裝 Jini 在爪哇虛擬機器（Java Virtual Machine）上，然後 Jini 利用網路配接的分散式計算技術跟遠端函式呼叫達到在網路上擁用隨插即用的功能，如此一來當新的服務出現就能夠快速地加入組織並立即地提供服務。

在用戶端也能夠很快速地搜尋到此組織下有提供何種服務並使用它。委託人在安全上的功能是代表某些實體存取 Jini 組織的服務，在某些時侯服務端本身也會需要存取其它的服務端，而這種情況就是依照委託人此安全概念來實現的。而存取控制名冊在安全上的功能即當有用戶端想存取資源時，必須依照存取名冊的規定來行動，不可採取違反存取名冊的行為。

3． 高安全性交易平台



SET 架構圖

圖四、SET 架構圖

在需要高度安全性的情況下，例如電子交易時。採取高安全性交易平台進行連線，利用系統晶片中內建之 RSA 電路用自己的密鑰加密（sign），接收方在得到加密過的摘要後如果能用傳送者的公鑰正確解密（verification），那麼接收方就可以百分之百確信這個文件的確來自傳送方。而傳送者傳送文件前先透過一個 Hash 函數將資料轉換成一段訊息摘要，再將此訊息摘要和文件一齊送出。如果中途有人修改過此文件，那麼接收者將被修過的文件經過 Hash 函數重新計算過後，算出來的摘要和傳送者送來的摘要不相吻合，能順利偵測到別人的偽造行為。

在 SET 之架構下除了需要公鑰、密鑰外，尚需要一數位憑證(Digital Certificate)，原因是公鑰是在公眾網路上讓別人作為傳送資料給自己的加密工具，如果公鑰被偽造散佈，那麼偽造者就可以假冒你攔截傳送給你的資料，並用對等的密鑰將資料解開。所以我們在傳送資料之前也必須先識別對方身分，確認公鑰確實是接收者的，才可將資料送出。為了做到公鑰的確認，可經由公鑰的認證來完成，此即數位憑證。

除此之外我們還需要一位可信賴的第三者來做公鑰的認證工作。這個第

三者就是所謂的憑證管理中心(Certificate Authority，簡稱 CA)，而 CA 必須是可被傳送者和接收者信賴的角色，他會依據合法申請者的請求發出數位憑證，數位憑證裡面包含了申請人的辨識資料(姓名)、公鑰及 CA 對這把公鑰的簽章，有 CA 的簽章背書後，我們就可以信賴這把公鑰。在網路上只要透過 CA 的公鑰的驗證我們就可以辨識對方的身份，如此一來，不同的個體就可以在高度的信賴下進行高安全性的交易。

四、 結論與未來展望

硬體、作業系統、軟體、加解密技術、網路環境只要任何一環節出現漏洞，則安全性就產生漏洞。所以要做到真正安全的環境，只有在硬體、作業系統、軟體、加解密技術、網路環境全都是可靠的、安全的才有可能達到。因此在這個計畫我們利用卡片和 RSA 裝置確保本人持有，再加上 AES 安全演算法用來傳輸資料以及 Jini 來達到資源共享時的一般安全性，最後利用 SET 來達到更高安全性的需求，尤其適合使用於電子交易環境上。這是我們初步提出系統晶片的安全環境。

然而，雖然許多的加密演算法在其理論基礎上是不容至疑的，但一旦導入硬體產品的開發時，就可能產生出新型態的安全性問題仍值得我們探討。未來可能持續進行的研究有：

a. 在 FPGA 板上實作內含 RSA 和 AES 演算法之晶片
b. 針對傳統加密法的新型態硬體錯誤攻擊法之研究
c. 針對 AES 現有及未來可能之硬體錯誤攻擊法之研究
d. 能量耗損監控攻擊法之研究
e. 新型態之硬體錯誤攻擊模式之研究

五、 參考文獻

[1]賴溪松, 韓亮, 張真誠, "*近代密碼學及其應用* ",旗標,民國 92.

[2]W. Stallings, "*Cryptography and Network Security: Principles and Practices*", Prentice-Hall,third Edition,2003

[3] Mastercard and Visa, Secure Electronic Transaction（SET）Specification Book1：Business Description, Version1.0, May 31, 1997.

[4] Mastercard and Visa, Secure Electronic Transaction（SET）Specification Book2：Programmer's Guide, Version1.0, May 31, 1997.

[5] Mastercard and Visa, Secure Electronic Transaction（SET）Specification Book3：Formal Protocol Definition, Version1.0, May 31, 1997.

[6] 張真誠, 林祝興, 江季翰, "*電子商務安全*", 松崗電腦圖書資料股份有限公司,pp4-2~4-5, 2000.

[7] N. Bannow, "*Java-processor for SmartCards and small embedded system*", (in German) Diploma thesis, Institute of Applied microelectronics and computer engineering, University of Rostock, Dec. 2000.

[8] M, Constantinos, "*Java Card Technical and Security*", Information Security Technical Report, vol.3, no.2, 1998, pp.82-89.

[9] 李卓俊, "*一個具有支援Java Card之智慧卡作業系統雛形的設計與實作*", 國立成功大學工程科學研究所碩士論文, June 2001.

[10] Scott Oaks , "*Java Security*", 2nd Edition, May 1, 2001

[11] Jan Newmarch, " *A Programmer's Guide to Jini Technology*",November, 2000

[12] 探矽工作室, "*2002 嵌入式系統開發聖經*",學貫行銷股份有限公司,2002

[13] Jean J. Labrosse, "*MicroC/OS-II : the real-time kernel*",CMP Books