邊緣運算安全

■作者：Silvano Di Ninno/ 恩智浦半導體安全軟體工程師
Nicusor Penisoara/ 恩智浦半導體安全、圖形和機器學習軟體研發部高級總監
Marc Vauclair/ 恩智浦半導體高級安全系統架構師、技術研究員

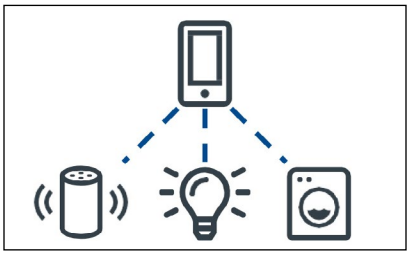
隨著越來越多的邊緣設備整合到系統中，邊緣設備的安全性也日益重要。本章探討功能和平臺安全方法，以解決在設計階段通過風險評估確定的薄弱點。

安全性：整體系統屬性

未來的系統將包含數十億的互連設備。它們將成為對攻擊者具有極大吸引力的目標。邊緣設備是這些系統的一部分。這些設備與遠端服務和基於雲的服務相連時，在更接近資料生成的位置處理資料。它們從感測器採集原始資料，分析和提取相關資訊，然後將其傳送到雲端並呈現給本地用戶。在大多數情況下，該資訊包含需要加以保護的敏感性資料。此外，一些邊緣設備配有執行器，用於控制可能會被濫用的公用設施和機器；因此，需要驗證從遠端服務收到的命令的真實性及其是否來自正確的控制實體。這使得邊緣設備成為攻擊者感興趣的目標。

安全保障是一種整體系統屬性；它不是一種附加屬性。系統的安全程度取決於攻擊者可觸及的最薄弱的組件。邊緣設備將在系統安全中扮演多種角色 ：

圖說：邊緣設備屬於優先攻擊目標 (攻擊者攻擊控制多個設備的邊緣設備，可獲得更高的投資回報)，所以這些設備必須得到相應的保護。



大多數保護網路設備的基本概念也適用於保護邊緣設備 ：

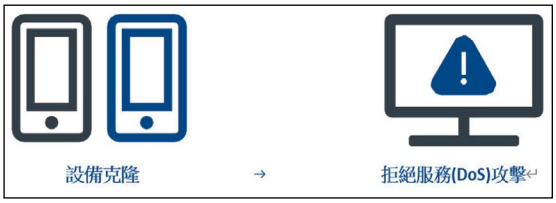
■傳輸中的資料
■存取控制機制
■靜態資料

圖說：通過將安全性較低的子系統進行必要的隔離或向其提供安全服務，有助於提升系統安全。

然而，儘管資料中心基礎設施 (通常部署雲服務的地方) 為存儲和處理的資料提供了一定的固有物理保護，但邊緣設備所在的位置幾乎沒有或完全沒有任何物理保護保障。這意味著需要為邊緣設備部署額外的保護和一定程度的入侵偵測功能。在系統晶片 (SoC) 層面，這些需求必須得到以下硬體功能的支援，並結合特定的軟體輔助技術來保護和應對此類攻擊：

■信任根 (RoT)
■安全啓動
■篡改檢測
■安全鎖區

邊緣設備及其資料保護能力是邊緣運算安全性能的一個關鍵組成部分，但保護網路和雲服務避免受邊緣設備轉化為遠端控制武器的影響也非常重要。邊緣設備需要以各種新方式處理日益增加的大量資料。多年來，為了解決這個問題，運算能力和可用網路頻寬都在增長。隨著邊緣人工智慧 (AI) 的出現以及 5G 和速度更快的 Wi-Fi 技術的部署，這種趨勢仍會延續。鑒於邊緣設備和雲基礎設施之間的交互越來越緊密，通過以下方法破壞雲服務的風險正在增加，其中包括 ：



邊緣設備安全包括兩個方面：

功能安全——這些安全原語 (大多數時候是密碼學特性) 確保敏感性資料根據需要保持安全 (機密性、完整性、真實性) 和私密。例如，在兩個邊緣設備之間使用加密消息是一種功能安全特性。

平臺安全——這些安全原語確

保出現遠端 (或有時為本地) 攻擊者時，能夠通過實施功能安全來保障安全。例如，對用於加密和解密兩個通信邊緣設備中任一設備消息的金鑰進行保護，就是一個平臺安全功能 (此安全功能的一個潛在實現是具有存取控制和隔離功能的安全存儲，因此即使對受攻擊邊緣設備的其他軟體部分進行遠端攻擊，也無法恢復此金鑰)。但是，如果將本機攻擊 (即對邊緣設備進行物理訪問的攻擊) 視為威脅模型的一部分，則應實施更先進的硬體安全功能，例如防旁路和故障攻擊。

針對這兩種安全功能，可利用一些基本的密碼學特性來保護邊緣節點。通過加密協議和金鑰管理技術，可以保護傳輸中的資料和靜態資料，並可確保邊緣設備啓動和運行真實且獲得授權的固件和軟體。此外，可以為設備構建一個唯一的身份標識，以便在邊緣設備端對其進行驗證和保護，同時在雲基礎設施端利用其進行存取控制。此唯一身份標識可說明檢測設備是否已受損並需要隔離以防 DoS 攻擊。

更先進的邊緣運算設備所需的另一個關鍵功能是，在單個 SoC 上創建和保護多個軟體執行環境的能力。這是隔離敏感軟體處理所需的功能。要將應用程式與協力廠商軟體棧整合為邊緣運算產品的一部分，必須確保這些不同軟體棧之間實現安全通信。

最後，安全功能和對策需要無縫集成到軟體棧中，以便能夠通過邊緣設備上運行的應用程式輕鬆部署。使用標準和 / 或完善的開源應用程式設計發展介面 (API)，對於盡可能重複利用軟體至關重要。

密碼學特性

密碼學特性依賴於密碼學演算法，通過在系統中提供以下一項或多項能力來確保其功能性 ：

完整性——接收到的資料與發送的資料相同。

機密性——偵聽通信的協力廠商無法解讀該消息。

真實性——消息接收方可以驗證消息的真實性。

不可否認性——消息發送方不能否認已經發送了該消息。

雜湊函數

雜湊函數是一個密碼學函數，它可以將任意數量的資料作為輸入，並提供固定位數的輸出。該函

表 1：演算法的安全屬性

演算	完整性	機密性	真實性	不可否認性
雜湊	✓			
對稱加密		✓		
消息認證碼	✓	✓		
公開金鑰	✓	✓	✓	✓

圖 1：雜湊函數的單向屬性

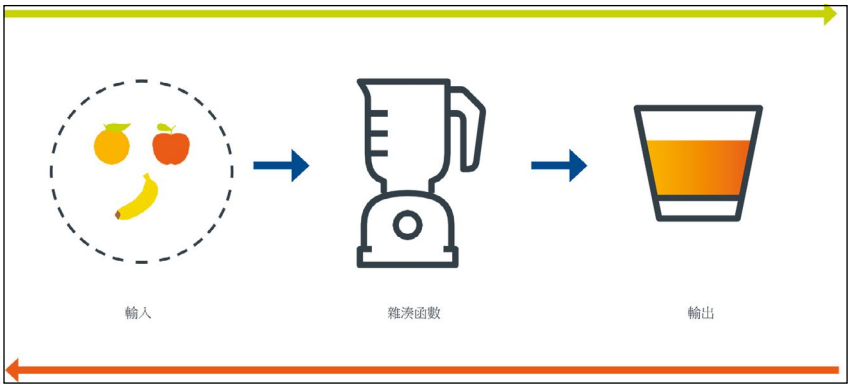
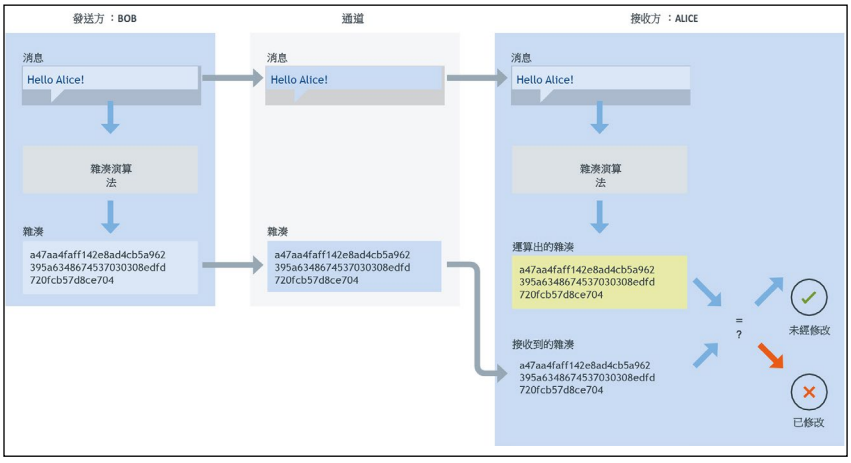


圖 2：使用雜湊函數保證消息的完整性

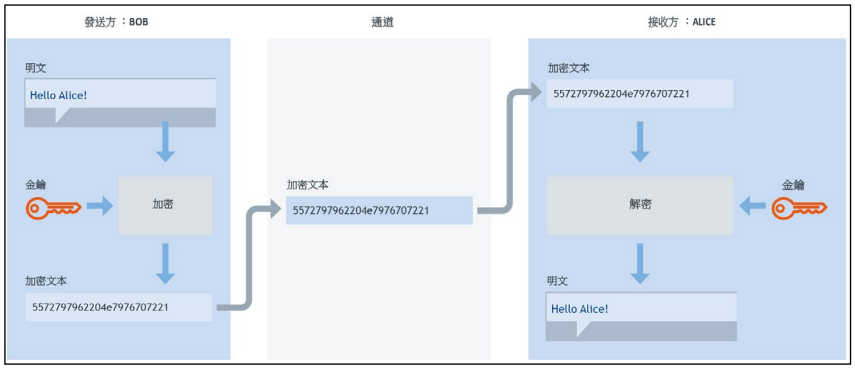


數 (見圖 1) 很容易由已知輸入生成輸出，但無法僅根據輸出結果逆推出輸入資料。另一個特性是，不可能輕鬆地創建兩個不同的輸入資料點，並給出相同的輸出結果。

此函數常用於以下目的 (見圖 3.2)：

有幾種可用的雜湊函數。像 MD5 和 SHA-1 等一些函數被認為是過時的弱雜湊函數。應該避免使用這些函數，盡管有些仍然被用於弱檔完整性檢查或檔識別。在這些情況下，不要依賴 SHA-1 的 MD5 來確定真實性。迄今 為止，推薦的標準化函數是來自 SHA-25 系列的函數；如果用於簽名和簽名驗證，這些是強制性的。

圖 3：使用對稱式金鑰密碼編譯來確保消息的機密性



對稱式金鑰密碼編譯

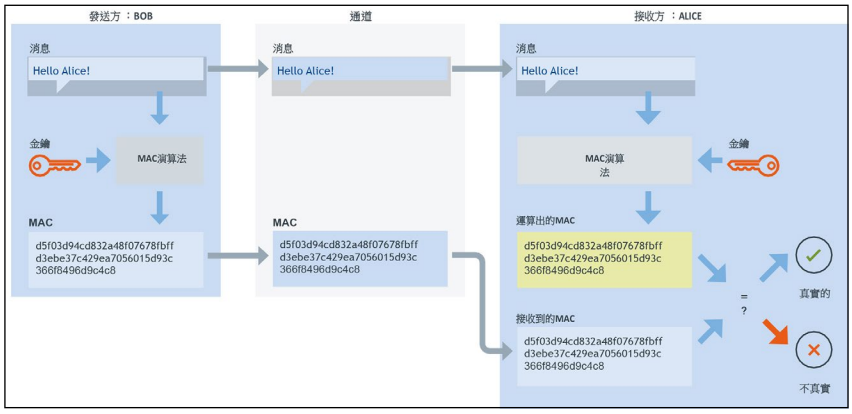
對稱式金鑰密碼編譯 (見圖 3) 使用發送方和接收方共用的金鑰來加密和解密資料。換句話說，Alice 和 Bob 使用相同的金鑰進行加密和解密。相對而言，該功能涉及的運算速度更快，因此使用者在交換大量資料時更喜歡使用它。該功能的主要缺點是，用於加密 / 解密資料的金鑰 (即對稱金鑰) 必須在通信雙方之間適當共用，以確保它不 會落入壞人之手。在兩個遠端方之間交換資料時，安全地生成和共用對稱金鑰會有問題。當需要在更多參與方之間共用機密資訊時，這個問題就更嚴重了。此外，基本加密 / 解密功能僅提供資料的機密性。此功能通常與公開金鑰加密一起用於金鑰分發，與身份驗證功能一起用於完整性驗證。衆多對稱金鑰演算法中最常見的演算法是 AES 系列 (AES-128、AES-192 和 AES-256)

對稱式金鑰密碼編譯常用於以下情況 ：

消息認證碼演算法

消息認證碼 (MAC)7 演算法是一個用於完整性和真實性檢查的加密函數。該函數類似於添加了共用金鑰的雜湊函數 (見圖 4)。任何人都可以驗證雜湊值，但只有知道共

圖 4：使用 MAC 保證消息的真實性



用金鑰的人才能驗證 MAC。MAC 演算法分為兩大類：雜湊 MAC 使用其中一種雜湊演算法作為操作的底層函數，基於密碼的 MAC 使用對稱式金鑰密碼編譯 (如高級加密標準或 AES) 作為底層函數。

此函數通常用於以下情況 ：

公開金鑰加密 (非對稱加密)

與使用單一金鑰進行加密和解密操作的對稱式金鑰密碼編譯相反，非對稱加密 (也稱為公開金鑰加密 (PKC)) 使用一個密鑰對。每個參與方都有一個由私密金鑰 (只有金鑰對的所有者知道) 和公開金鑰 (所有參與方都知道) 組成的金鑰對。這些金鑰可以採用不同方式使用。非對稱加密的下述兩個主要用途如圖 5 和 6 所示。

PKC 解決了一個基本問題，即允許兩個以前從未接觸過的實體之間建立真實的機密消息交換。

PKC 函數比對稱函數的運算量更大，因此通常用於處理少量資料，可支援以下用例 ：

■建立通信通道——數位簽章功能允許參與方識別通信中的各方，而加密功能則用作金鑰交換機制的一部分，以商定之後用於通信的新金鑰。這種用法的一個示例是 TLS 協議。

■安全啓動機制——數位簽章用於驗證二進位檔案的真實性。製造商使用其私密金鑰對軟體進行簽名，並分發其公開金鑰，以便任何其他實體都可以驗證該軟體的真實性。

■公開金鑰基礎設施 (PKI)——當使用 PKC 時，參與方的身份需要與其公開金鑰綁定，並且需要確保這些金鑰的新鮮度。這就是 PKI 的作用：通過發出證書和撤銷狀態資訊來認證這些連結。請參閱本文後面的“公開金鑰基礎設施”部分。

圖 5: 使用非對稱加密來保護少量消息的機密性

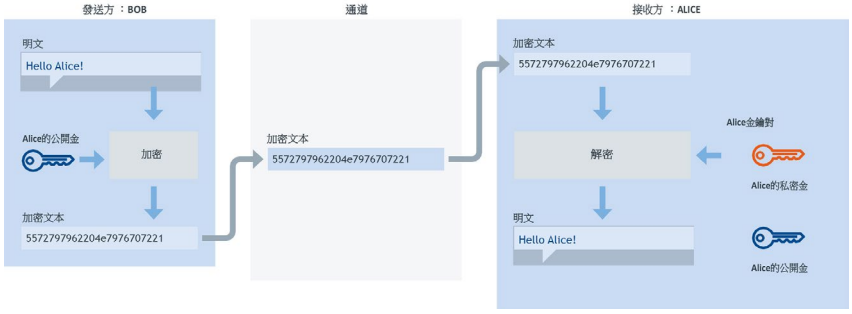
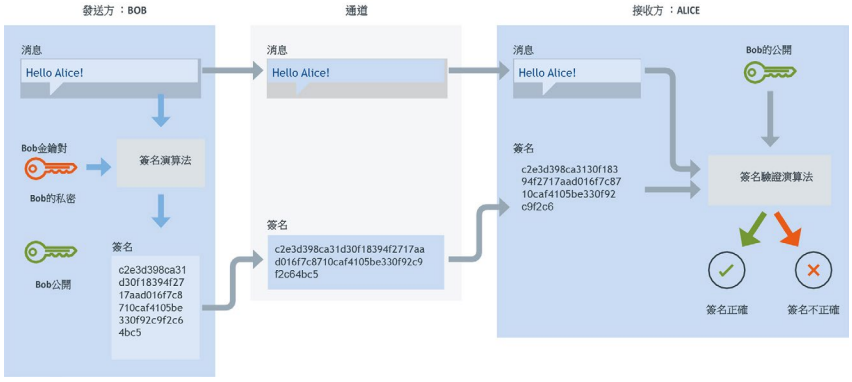


圖 6: 使用非對稱加密技術對消息進行數位簽章



加密以提供完整性和機密性

參與方使用消息預定接收方的公開金鑰加密消息，並且只有預定接收方能夠使用其私密金鑰解密消息。例如，
Rivest-Shamir-Adleman (RSA)加密演算法可用於實現此目的。

數位簽章以提供真實性和不可否認性

參與方使用其私密金鑰對消息進行簽名，並將消息與簽名一起發送。所有其他參與方都可以使用發送方的公 鑰來驗證簽名。例如，RSA演算法和橢圓曲線數位簽章演算法(ECDSA)可用於實現此目的。如前所述，數位簽 名演算法依賴於雜湊函數 ；它是簽名文檔的雜湊值。

■驗證服務——設備驗證意味著製造商已為設備提供了該設備特有的金鑰對。每當設備所連接的網路實體想要質詢設備的真實性時，該實體可以向設備發送質詢，並且設備可以發回質詢簽名，證明知道正確的 私密金鑰，從而驗證其真實性。

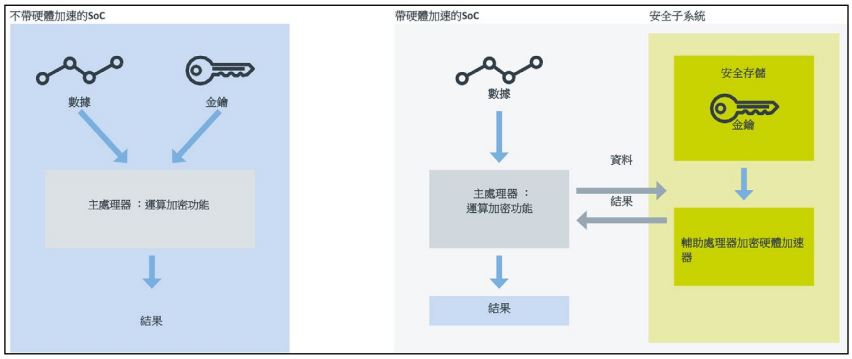
硬體加速

通過硬體加速，應用程式可以將某些運算任務分流至系統內的專用硬體元件。為了優化性能並保持低功耗，在許多邊緣運算設備上通常會對前面討論的加密功能進行硬體加速。

取決於邊緣運算設備，可能會有一個或多個硬體加速器。在某些設備中，主核心處理器利用特殊的加密運算加速代碼進行擴展；在許多其他邊緣設備上，將一個或多個附加輔助處理器集成到 SoC 中。

例如，恩智浦 i.MX 產品系列提供一個稱為加密加速器和保證模組 (CAAM) 的硬體模組，為密碼學運算提供硬體加速。除了加速功能之外，CAAM 還提供特殊的金鑰封裝功能，可保護敏感金鑰避免暴露。此金鑰封裝機制依賴於一組 SoC 主內核無法訪問的熔斷絲 (fuses)。這意味著即使攻擊成功破壞了主內核的行為，攻擊者也無法訪問封裝的金鑰，因為他們無法訪問 CAAM 的內部機制。即使在 SoC 上運行的軟體受損，SoC 的主內核也無法訪問圍繞這些功能構建的金鑰存儲。

圖 7：硬體加速



加密硬體加速器也是硬體隔離設備。加速器有自己的處理單元，包括寄存器和自己的本機存放區器。SoC 中的其他內核無法直接訪問這些資源。

使用硬體加速器並搭配設備驅動程式，將其集成到加密軟體庫中，以便從應用軟體角度抽象出密碼學運算。

硬體加速器還為各種密碼學演算法提供有保障的 (有時是經過認證的) 實現方案。此外，它們可以採納額外的安全對策，以防範旁路攻擊或故障注入攻擊，阻止試圖訪問加密運算期間使用的加密憑證的行為。

硬體亂數產生器

許多安全原語的安全性，更具體地說，許多密碼演算法和協定的安全性，依賴於在運算過程中處理的一些資料的真隨機性。例如，秘密頻道的網路通訊協定只有在通信雙方之間發送的質詢真正隨機時才是安全的 (否則，攻擊者 就可以猜出工作階段金鑰並監聽通信)。使用 ECDSA 的消息數位簽章僅當 ECDSA 在簽名運算過程中使用真亂數時才有效 (否則，攻擊者就可以猜出並運算簽名者的私密金鑰，並以其名義偽造新簽名)。產生真亂數的唯一方法是添加硬體真亂數產生器 (TRNG)，該生成器由軟體使用。現代 SoC 集成了一個硬體 TRNG，可以通過受保護的介面進行訪問。TRNG 主要用於為偽亂數產生器提供種子，因為每個時間單位 TRNG 生成的熵是有限的，而某些協定在每個時間單位內需要更多的隨機材料。

公開金鑰基礎設施

公開金鑰基礎設施 (PKI) 定義並證明公開金鑰與實體、個人和 / 或組織之間的綁定。通過引入一個或多個協力廠商，由其負責證明這些實體的身份以及與其所公佈公開金鑰的關係，PKI 可為兩個或多個使用 PKC 的實體之間的資料交換提供便利。這在邊緣運算中愈發重要，因為在某些架構中，邊緣設備可以充當這些協力廠商。它們具有必要的實施安全功能，可確保其不會被濫用。

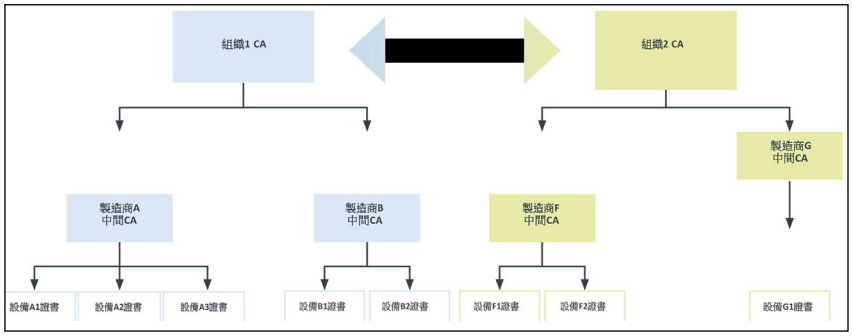
數位憑證可用於表示身份標識和公開金鑰之間的這種綁定關係。最常見的是 X.509 v3 證書 8。如圖 8 所示，它不僅包含與金鑰對相關聯的組織的身份標識資訊，還包含證書的發行方、簽署方和有效期等資訊。

圖 8：x509.3 證書內容示例

版本
證書序號
有效期
主體名稱
公開金鑰信息
發行方名稱
擴展部分（例如金鑰使用）
簽名演算法識別字
簽名

這些證書的頒發方稱為憑證授權 (CA)。CA 負責在發出數位憑證之前，檢查身份標識和公開金鑰之間的連結以及其他資訊。認證實體可接收這些證書，或者將證書發佈在任何人都可以訪問的公共存儲庫中。當另一個實體收到證書時，由於知道證書頒發方 (簽署證書的 CA) 的公開金鑰以及驗證實體對這個 CA 的信任，即可驗證證書的有效性。接收實體還可以檢查證書在其頒發後到當前使用期間是否已被撤銷。擁有一個適合所有用途的 CA 既不可行也不可取，其原因有多種：可擴展性、可用性、機密性等。這就是為何 PKI 不是一個 CA，而是由多個 CA 組成，這

圖 9：PKI 基礎設施示例



些 CA 通常按分層結構組織。頂層 CA 認證次級授權機構，而次級授權機構又認證實體。各個 CA 通常也會交叉認證彼此的證書。

證書的壽命是有限的；它們有一個開始使用日期和一個結束使用日期。此外，如果由於任何原因 (證書所有者消失、與公開金鑰關聯的私密金鑰安全性漏洞等) 導致證書撤銷，CA 將公佈此有效期到期。如果被撤銷的證書出現在數字交易中，首先檢查該證書是否已被撤銷的實體可以拒絕該證書。

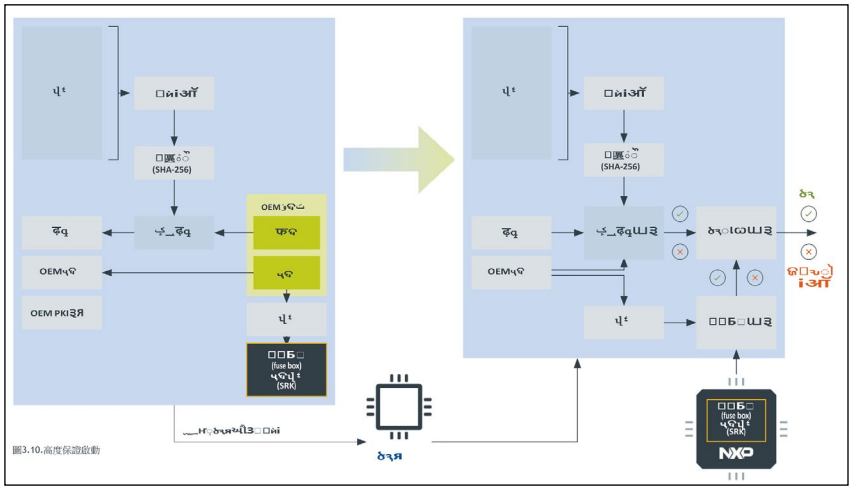
PKI 是一組硬體和軟體元件以及定義的策略。它提供以下服務 ：

不應將 PKI 視為管理所有可用證書的單一實體。PKI 不僅可以是資料中心中大規模的公共實現，也可以是私有的單一公司實現，甚至可以是單個 SoC 中的實現。

PKI 最常見的用途是在使用 “https” 網址流覽互聯網時，使用安全通訊端層 (SSL) 證書。邊緣設備也可以使用

“https” 網址來訪問互聯網。邊緣連接設備使用 PKI 的其他範例還有設備生命週期管理和安全啓動。

圖 10：高度保證啓動



安全啓動：信任鏈

構建安全系統的一個關鍵方面是保證真正的二進位檔案運行，並且擁有或提供系統的 OEM 信任這些二進位文件。安全啓動功能提供了這種保證。安全啓動通常從 SoC 內的一個不可變存儲區域開始，該區域本身始終可信，並且始終保持正確運行。SoC 的這個不可變部分及其相應功能有時被稱為信任根 (RoT)。通常使用 PKC 演算法作為支援系統安全啓動的底層技術。

OEM 或負責晶片上初始固件運行的實體將創建一個公開金鑰 / 私密金鑰對。此操作在安全設施中進行，私密金鑰以訪問受控方式實現安全存儲。通常使用專門的硬體安全模組 (HSM) 來生成、存儲和應用此金鑰 (即，金鑰永遠不會離開 HSM，只有授權使用者才能使用它)。

該金鑰的公共部分成為設備 RoT 的一部分。這是用於驗證任何其他軟體的首個可信元件。必須注意確保在設備中放置了正確的公開金鑰；此步驟通常在生產線的 “ 信任配置 ” 過程中實施 (參見第 6 章中的 “OEM 關閉狀態 ” 部分)。

私密金鑰用於簽署將在安全系統上執行的固件。此過程很敏感，需要由 OEM 正確控制對二進位檔案進行簽名的許可權。例如，如果出現錯誤並且對未準備好用於生產的二進位檔案進行了簽名，以

後使用這些二進位檔案可能會對部署的邊緣設備造成不利影響。

考慮多款異構 i.MX 應用處理器。它們使用一種類似于安全啓動過程的機制，稱之為高度保證啓動 (HAB)，如圖 10 所示。

安全和隔離

密碼學演算法和協定已經存在了幾十年。如果使用得當，可以說是強大且可靠。然而，多次網路攻擊表明，邏輯錯誤 (亦稱為 bug) 是邊緣運算系統安全保障的薄弱點。可用於破壞系統安全的錯誤稱為漏洞。代碼庫越大，系統上運行的軟體漏洞數量就越多。

在需要一定安全等級的系統中採用一種緩解技術，將敏感資訊的存儲和處理隔離到專用硬體中。例如，在智慧　卡行業中使用安全晶片來保護金鑰和敏感性資料。PC 行業也是如此，它使用可信平臺模組 (TPM) 技術來保證軟體運行的完整性。實踐證明這種方法行之有效，仍是當今所能提供的最佳保護之一。然而，對於某些用例來說，該解決方案的擴展性並不好，因為這些年來，對敏感操作處理能力的需求一直在增加。硬體擴展也集成到 SoC 的 CPU 中，以增加系統的隔離能力，同時充分利用 CPU 的性能優勢。這些隔離技術最初引入 PC、伺服器和移動設備中，但現在同樣適用於邊緣設備中使用的 SoC。

現代 SoC 實現了以下一種或多種隔離技術　：

可信執行環境

第一類硬體擴展旨在隔離可信功能。可信執行環境 (TEE) 的概念是在 21 世紀初為滿足這些新要求而引入的。TEE 在應用處理器內實現了一個安全區。大型作業系統 (例如 Linux 和 Android) 可能會暴露許多漏洞，因此被視為不可信，將在 TEE 之外稱為富執行環境 (REE) 的區域中執行。那些考慮到安全性而精心編寫且易於審計的軟體　將在 TEE 內運行。為了滿足這些需求，一些硬體供應商開發出 CPU 架構擴展，包括對這種可信和不可信軟體分區的硬體支援。這些擴展包括 Intel SGX9、AMD 安全加密虛擬化 (SEV)10 和 Arm TrustZone 11。有多種可用的軟體實現，包括專有選項和開源選項。這些實現方案大多都遵循類似的軟體架構，即在 TEE 中運行的軟體為在 REE 中運行的軟體提供一系列服務。TEE 軟體棧本身通常由以下兩個元件組成　：

■TEE 核心元件 (也稱為 TEE 內核) 提供核心隔離機制以及加密和存儲服務等基本服務。

■在 TEE 核心元件上運行的 TEE 應用程式為 REE 的專用服務提供定制化和使用者可程式設計性。在邊緣運算設備上，多個可信服務應用程式在 TEE 中運行，包括：

TEE 不僅可用於對處理器和記憶體進行分區，還可用於對週邊設備進行分區。某些 TEE 技術支援設備分離，從而使某些設備只能由 TEE 訪問，而不能由 REE 訪問。

(因文太長，本文後面還有軟體棧示例、虛擬化介紹，讀者可在網站瀏覽全文) CTA