I440351

(19)中華民國智慧財產局

(12)發明說明書公告本 **(11)證書號數：TW I440351 B**

(45)公告日： 中華民國 103 (2014) 年 06 月 01 日

(21)申請案號：099142160 (22)申請日： 中華民國 99 (2010) 年 12 月 03 日

(51)Int. Cl.： *H04L9/14 (2006.01)* *H04L9/28 (2006.01)*

H04L9/32 (2006.01) *G06F21/00 (2013.01)*

(30)優先權：2009/12/04 美國 61/266,948

2010/12/02 美國 12/958,570

(71)申請人：密碼研究公司 (美國) CRYPTOGRAPHY RESEARCH, INC. (US)

美國

(72)發明人：果奇保羅 C KOCHER, PAUL C. (US)；洛哈班齊 ROHATGI, PANKAJ (US)；傑菲喬夏 JAFFE, JOSHUA (US)

(74)代理人：陳長文

(56)參考文獻：

US 2008/0263363A1 US 2009/0070583A1

審查人員：柯建羽

申請專利範圍項數：21 項 圖式數：14 共 0 頁

(54)名稱

可檢驗的、對洩漏有抵抗性的加密與解密

VERIFIABLE, LEAK-RESISTANT ENCRYPTION AND DECRYPTION

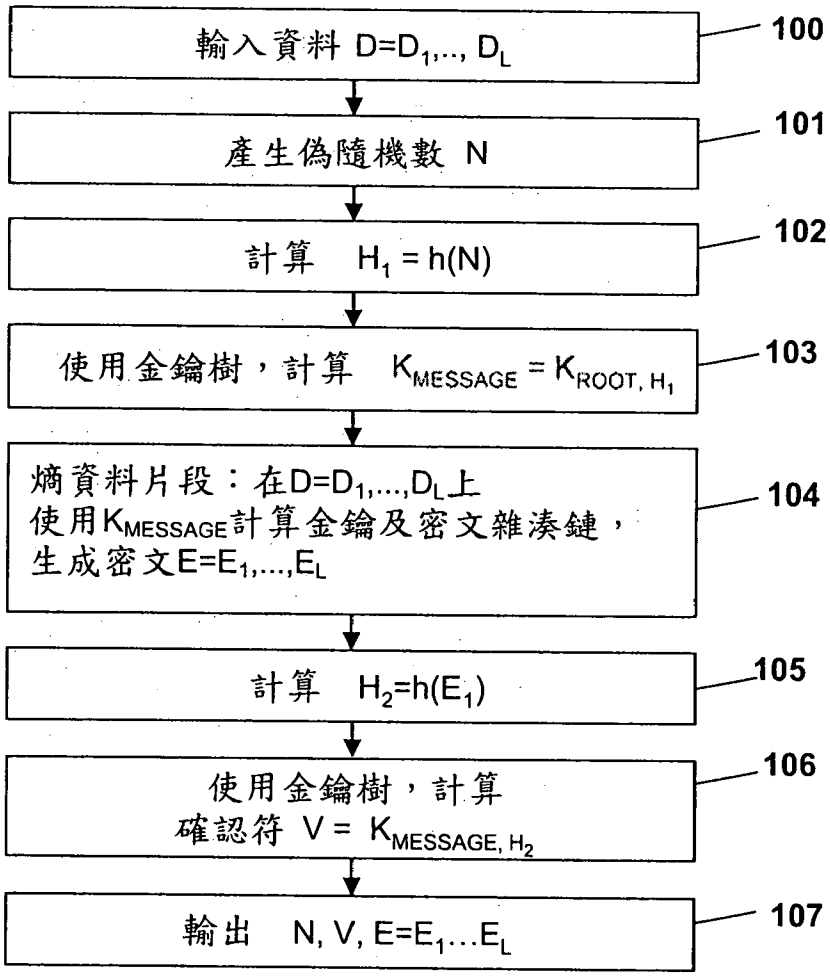
(57)摘要

本專利說明一種由裝置可使用以提供遠離外部監控攻擊的安全方式加密及解密敏感性資料的技術。加密裝置有權存取一基本機密密碼值(金鑰)，該值亦為解密裝置已知。敏感性資料被分解為片段，且各片段以從基本金鑰及一訊息識別符推導的一分開的加密金鑰加密，來創造一組加密的片段。加密裝置使用基本機密密碼值創造確認符(validator)，其證實用於此訊息識別符的加密的片段係由有權存取基本金鑰的一裝置所創造。解密裝置在接收到一加密的片段及確認符之後，立即使用確認符檢驗訊息識別符及加密的片段是否被修改，接著使用從基本金鑰及訊息識別符推導的一密碼金鑰解密片段。所推導的金鑰及確認符使用經設計的方法產生，因而即使密文及雜湊運算洩漏資訊，仍維持安全性。說明用於系統的實施例包括 SoCs、韌體載入、FPGAs 及網路通訊。

This patent describes techniques usable by devices to encrypt and decrypt sensitive data to in a manner that provides security from external monitoring attacks. The encrypting device has access to a base secret cryptographic value (key) that is also known to the decrypting device. The sensitive data are decomposed into segments, and each segment is encrypted with a separate encryption key derived from the base key and a message identifier to create a set of encrypted segments. The encrypting device uses the base secret cryptographic value to create validators that prove that the encrypted segments for this message identifier were created by a device with access to the base key. The decrypting device, upon receiving an encrypted segments and validator(s), uses the validator to verify the message identifier and that the encrypted segment are unmodified, then uses a cryptographic key derived from the base key and message identifier to decrypt the segments. Derived keys and validators are produced using methods designed to preserve security even

if cipher and hashing operations leak information. Embodiments for systems including SoCs, firmware loading, FPGAs and network communications are described.

100-107・・・步驟



第1圖：使用密文雜湊鏈的可檢驗抗洩漏加密

發明專利說明書 公告本

（本說明書格式、順序，請勿任意更動，※記號部分請勿填寫；惟已有申請案號者請填寫）

※ 申請案號：99142160

※ 申請日期：2010 年 12 月 3 日　　※IPC 分類：

H04L 9/14 2006.01
H04L 9/08 2006.01
H04L 9/32 2006.01
G06F 21/00 2013.01

一、發明名稱：（中文/英文）

可檢驗的、對洩漏有抵抗性的加密與解密 / VERIFIABLE, LEAK-RESISTANT ENCRYPTION AND DECRYPTION

二、中文發明摘要：

本專利說明一種由裝置可使用以提供遠離外部監控攻擊的安全方式加密及解密敏感性資料的技術。加密裝置有權存取一基本機密密碼值（金鑰），該值亦為解密裝置已知。敏感性資料被分解為片段，且各片段以從基本金鑰及一訊息識別符推導的一分開的加密金鑰加密，來創造一組加密的片段。加密裝置使用基本機密密碼值創造確認符（validator），其證實用於此訊息識別符的加密的片段係由有權存取基本金鑰的一裝置所創造。解密裝置在接收到一加密的片段及確認符之後，立即使用確認符檢驗訊息識別符及加密的片段是否被修改，接著使用從基本金鑰及訊息識別符推導的一密碼金鑰解密片段。所推導的金鑰及確認符使用經設計的方法產生，因而即使密文及雜湊運算洩漏資訊，仍維持安全性。說明用於系統的實施例包括 SoCs、韌體載入、FPGAs 及網路通訊。

三、英文發明摘要：

This patent describes techniques usable by devices to encrypt and decrypt sensitive data to in a manner that provides security from external monitoring

attacks. The encrypting device has access to a base secret cryptographic value (key) that is also known to the decrypting device. The sensitive data are decomposed into segments, and each segment is encrypted with a separate encryption key derived from the base key and a message identifier to create a set of encrypted segments. The encrypting device uses the base secret cryptographic value to create validators that prove that the encrypted segments for this message identifier were created by a device with access to the base key. The decrypting device, upon receiving an encrypted segments and validator(s), uses the validator to verify the message identifier and that the encrypted segment are unmodified, then uses a cryptographic key derived from the base key and message identifier to decrypt the segments. Derived keys and validators are produced using methods designed to preserve security even if cipher and hashing operations leak information. Embodiments for systems including SoCs, firmware loading, FPGAs and network communications are described.

四、指定代表圖：

(一)本案指定代表圖為：第（ 1 ）圖。

(二)本代表圖之元件符號簡單說明:

100-107 步驟

五、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

六、發明說明：

【發明所屬之技術領域】

本專利係關於用於處理加密的資料輸入的技術，且更明確而言，係關於針對外部監控的攻擊保護此等系統及資料。

【先前技術】

運算敏感資料的系統需要針對由攻擊者揭露或改變的此等資料的未授權的存取而作保護。得到對密碼金鑰及其他機密的存取的攻擊者可能竊取或竄改敏感資料，導致嚴重的後果，例如透過引進未授權的命令而顛覆系統的關鍵運算，及暴露機密的或所有人的資訊。一個受連累的元件亦可用於發動進一步攻擊，使得系統的其他元件遭到危險。更明確而言，先前研究已顯示一攻擊者可監控一裝置的外部特性，例如運算時間點、功率消耗及/或電磁輻射，且使用此額外的資訊以提取用於裝置中的機密金鑰。舉例而言，如 Kocher 等人所述（見 P. Kocher, J. Jaffe, B. Jun,「Differential Power Analysis」 Advances in Cryptology - Crypto 99 Proceedings, Lecture Notes In Computer Science Vol. 1666, Springer-Verlag, 1999），本領域中眾所周知一裝置的外部監控對不同資料使用相同組的金鑰實行一連串的密碼運算可導致金鑰的洩漏。

因為外部監控攻擊係典型地消極且非侵入性，所以傳

統的基於抑制實體存取或偵測不適當使用情形的抗竄改的防衛不足以或無法切實針對此等攻擊提供保護。舉例而言，使用實體保衛、良好防護的房間的管理機密金鑰的方法係在先前技術領域中為已知的。然而，在許多應用中，要求密碼系統在其給定的預期運算的環境中，以維持實體隔絕設備係並非可行的。此外，此等設備在建造及運作上係為昂貴的，且其能力可能仍非完美得避免洩漏至敵手的小量的資訊。

當然，先前技術領域已知的其他方法可避免監控攻擊而減輕資訊洩漏的問題，無須依賴實體防護。此包括用於減少資訊的量（或比率）從交易洩漏、修改密碼演算法實施以隨機計算中間值、及/或在功率消耗及運算時間點導入雜訊的方法。

舉例而言，美國專利號 6539092，名為「Leak-Resistant Cryptographic Indexed Key Update」，提供一種用於將一分享的主要金鑰及一索引值（例如，一計數器）轉換成一交易金鑰，其中推導係針對外部監控攻擊而作保護。該等方法在針對外部監控攻擊可貢獻至交易金鑰的推導的裝置的應用中很成功。舉例而言，'092 專利說明一智慧型卡可如何維持在各交易中增加的一索引計數器，接著使用索引計數器於金鑰推導中。

然而，存在某些應用，其中在一協定中的參與者應針對外部監控攻擊而被保護，但如'092 專利中所述，其缺乏儲存一連串計數器及更新的金鑰的能力。舉例而言，

考慮一裝置需要定期處理相同輸入資料的情況，例如包含一固定且不變的植入的金鑰的一裝置，其重複地被使用以任意的順序解密密文。韌體加密係為此一應用的一範例；可製造一微處理器在使用者空間檔案系統中具有一植入的金鑰，且在每一次重啟動中，微處理器需要從一非信任的外部快閃記憶體重新解密其韌體影像。韌體影像可偶然地被更新，但相同的密文亦可重複地被解密。因此，應用需求及實體製造限制兩者（例如歸因於一次性可程式化使用者空間檔案系統的使用以持有金鑰而無法修改儲存的金鑰）可使其對裝置限制解密金鑰將被使用的次數係為不切實際的。韌體發佈者可使用'092專利中所述的方法，其在每次釋出一新的加密的韌體影像時具有一新的索引值，但解密裝置無法在各重啟動上使用一不同的索引值，因為將索引值改變成不同於加密裝置所使用的一個值將導致一不正確的解密。因此，一攻擊者可潛在地供應具有竄改的資料組的解密裝置，接著嘗試在裝置處理（例如，解密等等）此等密文的同時藉由監控外部特性而恢復機密金鑰。統計側通道攻擊，例如差動功率分析（DPA），可推論來自一組量測的一機密金鑰，該組量測係在一裝置重複地使用相同金鑰以運算於不同輸入值時被收集（例如，在上述範例中的不同韌體密文或相同韌體密文的竄改的版本）。來自一單一長訊息的量測（例如，包含許多區塊加密（block cipher）輸入）或合理訊息的一收集（例如，多重韌體版本）亦

可提供足夠的資料用於一側通道攻擊，即使密文訊息並無被竄改。

當然，在一裝置對每一交易使用相同的金鑰的某些情況中，裝置理論上可實施一封鎖（lock-out）（例如，若超出一交易或失敗臨界，則藉由自我破壞而實施）以限制一敵手可觀察到的交易的數量。然而，封鎖機制導入數種實際的問題，例如可靠度問題及與儲存一失敗的計數器相關聯的困難（例如，許多半導體製造處理缺乏晶片上非揮發性儲存的保衛，且晶片外儲存係難以保衛）。

有鑑於所有以上所述，一種對裝置提供一可檢驗保衛方式以通訊且交換資料的方法，而針對外部監控攻擊保護且對裝置有能力拒絕非真實資料的方法將為有益的。

【發明內容】

本專利說明針對外部監控攻擊使用機密密碼金鑰保衛裝置的方式，以及針對蒐集關聯於裝置的內部運算的資訊的傳統密碼分析及其他攻擊（例如，DPA 及外部監控攻擊的其他形式）提供改善的安全性。用於加密敏感資料的各種範例實施例被揭露於說明書中。

儘管此等各種實施例在其細節中可能相當程度的改變，但其均包含於以下一般技術中，而可立即相對於說明書中所述的各種實施例被檢驗：相對於加密，待加密的各組資料係與一訊息識別符相關聯（例如，一交易/訊

息計數器、明文的一雜湊、一隨機值、或另一獨特或半獨特的值)。加密裝置使用訊息識別符及與加密裝置分享的一初始機密內部狀態推導一訊息金鑰。此推導以反覆的方式透過連續的一或多個中間金鑰而實行，從至少一部份分享的機密內部狀態開始且引導至訊息金鑰，其中，在各重複中，下一個金鑰取決於至少一個先前金鑰及至少一部份的訊息識別符。明文可被分解成一或多個片段。各明文片段以一或多個機密金鑰加密，機密金鑰可包括訊息金鑰或進一步從訊息金鑰推導出的金鑰，以創造相對應的加密的片段。典型地，各片段使用不同的金鑰（或不同組的金鑰）。

加密裝置接著使用與解密裝置分享的一機密金鑰（例如，訊息金鑰、機密內部機密、一不同的金鑰、從以上推導出的金鑰等等）以計算至少一個確認符。確認符的推導可使用與用於產生訊息金鑰類似的一重複處理而實行，其中一連串的變換式被應用至機密金鑰以產生連續值（舉例而言，各中間值的生成包括雜湊其親屬值)。

加密裝置輸出一或多個加密的片段及一或多個確認符。亦可如所需的輸出額外的資訊，以使得接受者可決定訊息識別符。

在相對應的加密處理期間，一解密裝置接收一或多個加密的片段、一或多個確認符、及相對應於加密的片段的訊息識別符。解密裝置接著使用一或多個確認符檢驗待解密的至少第一加密的片段並無被修改。確認符的檢

驗可包括從與加密裝置分享的一機密開始計算一連串連續的中間值，且其中各中間值係其親屬的雜湊（且特定雜湊運算取決於該加密的片段的一部份雜湊）。典型地，唯若檢驗出片段並未被修改時，准許進行對一加密的片段的解密處理。若檢驗成功，解密裝置使用與加密裝置分享的機密內部狀態，藉由跟隨由加密裝置所跟隨的相同反覆金鑰推導處理（即，從至少一部份分享的機密內部狀態開始，透過一連串中間金鑰，引導至最終訊息金鑰，其中於各步驟下一個金鑰取決於至少一部份的訊息識別符及至少一個先前金鑰），而計算訊息金鑰（若並未被推導）。各加密的片段（若決定為未被修改的）以從訊息金鑰推導出的一或多個相對應的機密金鑰解密，而恢復相對應的明文片段。

【實施方式】

此專利中所述的技術使得多方能夠針對外部監控攻擊而以增進的安全性通訊密碼保護的敏感資料。雖然範例實施例牽涉兩方說明，典型地稱為一「加密裝置」及一「解密裝置」，但是「裝置」一詞係為了方便起見而選擇，且無須直接相對應至一系統設計中的任何特定角色。舉例而言，加密及解密裝置兩者可均為可攜式硬體裝置。或者，加密裝置可為運算於一設備中運行於一伺服器上一軟體應用程式，同時解密裝置可為一可攜式硬體裝置

（反之亦然）。再者，雖然大多密碼運算牽涉兩方，但是此專利的技術當然可應用於僅牽涉一方的環境中（例如，在保衛記憶體或儲存系統中，其中兩者的角色係受到一單一方及/或裝置的控制，例如，第 8 圖中圖示的範例環境中），或應用於牽涉兩者以上的多方及/或裝置的環境中（例如，第 10 圖中圖示的範例實施例中）。

熵重新分佈運算

如此處所使用，一「熵重新分佈運算」（或「熵分佈運算」）係混合其輸入的一運算，使得關於輸入位元的未知資訊在輸出位元之間重新分佈。舉例而言，假設 x 位元密碼金鑰 K_0 係以一熵重新分佈運算 f 反覆地處理，使得金鑰 $K_i=f(K_{i-1})$，其中各個 $i>1$。接著，假設一敵手獲得關於 n 個不同金鑰 K_i 之各者的資訊的 y 位元（例如，被獲得為一嘗試的外部監控攻擊的部份），提供充分足夠的資訊以解出金鑰 K_0（例如，$y*n>x$）。熵分佈運算 f 的使用可使得此解決方案計算上為不可行的。一密碼雜湊函數 H 係運算的一範例，其可被使用作為一熵重新分佈運算。舉例而言，考慮產生一 256 位元結果的一強的雜湊函數 H。給定一隨機 256 位元初始金鑰 K_0，使得 $K_i=H(K_{i-1})$ 其中各個 $i>1$。具有（舉例而言）各 $K_0 \cdots K_{999,999}$ 的最不重要位元的知識的一敵手具有關於 K_0 的 1,000,000 位元資料。具有無限計算能力的一假設性敵手可藉由對 K_0 測試所有可能的 2^{256} 值而找到 K_0，以識別與最不重要的位元

的已知序列一致的一值。然而，實際敵手具有有限計算能力，且熵重新分佈運算避免在給定透過嘗試的外部監控攻擊所洩漏的資訊下被以一計算上實際方式解出 K_0（或任何其他 K_i）。

熵重新分佈運算可（而非限於）使用密碼雜湊函數實施，運算使用區塊加密（例如 AES）、擬隨機變換、擬隨機排列、其他密碼運算、或以上之結合而建構。為了方便起見，某些範例實施例以一雜湊敘述，但本領域中之技藝人士將瞭解依照以上敘述，亦可替代使用或額外使用其他熵重新分佈函數。

亦可從一基本運算建構多重熵重新分佈運算。藉由範例的方式，若需要兩個 256 位元的熵重新分佈運算 $f_0()$ 及 $f_1()$，則 $f_0()$可包含將 SHA-256 密碼雜湊函數應用至與對 $f_0()$的輸入連鎖的運算識別符字串「f0」，同時 $f_1()$可包含將 SHA-256 密碼雜湊函數應用至與對 $f_1()$的輸入連鎖的運算識別符字串「f1」。熵重新分佈運算可使用已知 AES 區塊加密（block cipher）建構。舉例而言，為了實施 $f_0()...f_{b-1}()$，各 $f_i()$可使用其輸入作為 AES-256 金鑰以加密在 0...b-1 的選擇中獨特的一對 128 位元輸入區塊，而產生 256 位元的輸出。基於雜湊函數及 MAC 建構的各種區塊加密（block cipher）亦為先前技術領域中已知且亦可被利用。

分享的密碼值及運算

此段說明某些密碼值及/或運算，由加密裝置及其相對應的解密裝置兩者分享，用於實行如此專利中所說明的可檢驗抗洩漏密碼運算。

加密裝置及解密裝置被設定，使得其各者可存取一基本分享的機密密碼狀態值，例如標示為 K_{ROOT} 的一機密金鑰。舉例而言，此機密狀態可儲存於 EEPROM、快閃、使用者空間檔案系統、或在一抗竄改晶片上的其他儲存之一或多者中，且可全部或部份從其他值或處理推導出，或可外部地獲得。此等裝置之各者獲得 K_{ROOT} 的方法可包括（但不限於）其各者以 K_{ROOT} 製造、裝置彼此直接溝通 K_{ROOT}，或透過第三方溝通（例如，使用利用 RSA、Diffie-Hellman、或其他公眾密碼技術的協定，或對稱技術）、透過一實體鍵入介面接收 K_{ROOT}、隨機產生 K_{ROOT}（例如，若加密及解密裝置係為相同的）等等。

此外，加密裝置及解密裝置兩者亦能夠計算一組非線性密碼熵重新分佈運算 $f_0(), f_1()...f_{b-1}()$，其中 $b>1$ 係為一正整數。此等 b 個熵重新分佈函數可配置於一樹狀結構中。舉例而言，具有高度為 Q 的一單純 b 相關的樹狀結構（即，具有從 0 至 Q 的 Q+1 層級）可使用 b 個相異熵分佈函數 $f_0()...f_{b-1}()$創造，以於樹的各節點代表此 b 相關的樹的 b 個可能分支，各節點代表一可能的推導的金鑰。在此一樹中，從一根密碼金鑰 K_{START} 開始（其於層級 0 處），可於層級 1 處計算 b 個可能推導的金鑰：最左邊的分支為 $f_0(K_{START})$；下一個分支為 $f_1(K_{START})$；且

繼續直到最右邊的分支為 $f_{b-1}(K_{START})$。於層級 2 處，可推導 b^2 個可能金鑰，因為 $f_0()...f_{b-1}()$之各者可應用至 b 個可能層級 1 金鑰之各者。當然，計算一特定層級 2 節點僅需要二的計算結果，而非 b^2（即，不在路徑上的節點並不被計算）。樹繼續用於連續的層級 1 至 Q，其中一先前層級的各可能金鑰（即，一不同的節點）可藉由依序應用 $f_0()...f_{b-1}()$而處理，以推導 b 個額外可能推導的金鑰。整體的金鑰樹具有 Q+1 個層級，以於層級 0 處的一單一節點開始，繼續於層級 i 處的 b^i 節點，且結束於層級 Q 處的 b^Q 節點。因此，存在從層級 0 處的根節點至層級 Q 處的 b^Q 最終節點的 b^Q 個可能路徑。各此等可能的路徑（相對應於應用於不同層級的一獨特連串函數）可以一連串 Q 個整數代表，各整數係選自 (0...b-1)。

舉例而言，在一範例實施例中，b＝2。因此，使用兩個熵重新分佈運算 $f_0()$及 $f_1()$（且可從一基本運算建構，例如上述）。若 Q＝128（即，高度係為 128），則可能有 2^{128} 個路徑且需要 128 個熵重新分佈函數計算結果以從層級 0 節點（即，開始金鑰）推導層級 Q 金鑰。

作為一變化，實施例可牽涉更多種 b 的選擇，例如在層級之間改變 b 的值，及/或基於一特定層級採取的路由改變 b。同樣地，亦可改變熵重新分佈運算，例如藉由使得熵重新分佈運算 $f_i()$於不同層級處不同，或使得此等運算取決於一特定層級所採取的序列。

加密及解密裝置亦能夠實行一密碼的、非線性金鑰鏈

運算 g()，其可與函數 f_i()相異（但並非必定的）。舉例而言，在一個實施例中，g()由一密碼雜湊運算組成。各種實施例可對 g()的不同應用使用不同的函數，包括從一基本函數建構的變數（例如，以一計數器或代表 g()的應用的另一值雜湊輸入資料）。

加密裝置及解密裝置亦具有一密碼的、抗衝突、單向雜湊函數 h()（例如，利用為一片段雜湊函數）而與運算 f_i()及 g()相異。

在一範例實施例中，運算 f_i()、g()及 h()之各者係藉由計算各運算為一運算識別符及輸入資料的密碼雜湊而從一通用密碼雜湊函數建構。舉例而言，運算識別符可為零終結字串，由「f# 」、「g」或「h」組成，其中 # 係為對給定的 f_i()的 i 的值，使得對 f_0()的運算識別符將為「f0」。使用輸入作為金鑰的一運算識別符的 HMAC 亦可被用於實施此等運算。可與此專利的技術一起使用的雜湊函數包括，但非限於 MD5、SHA-1、SHA-256、SHA-512、任何 SHA3候選、以及上述的結合及使用上述而建構（例如，HMAC）。如此處所使用，函數 BLAKE、Blue Midnight Wish、CubeHash、ECHO、Fugue、Grostl、Hamsi、JH、Keccak、LANE、Luffa、Shabal、SHAvite-3、SIMD及 Skein之各者係為一「SHA3 候選運算」。在其他實施例中，使用其他眾所周知的建構推導雜湊函數，例如但非限於 Matyas-Meyer-Oseas、Davies-Meyer、Miyaguchi-Preneel、Merke-Damgard 等等，其將諸如 AES、DEA 或其他加密的區塊加密（block

cipher）轉換成一雜湊函數。非抗衝突的變換（例如，MD5、雜湊變換的減少的迴圈變數、或其他混合的運算）亦可重新分佈呈現於輸入中的熵，但對用於單向函數 h() 將為較不具吸引力的。

儘管如此，其他實施例可在實施熵重新分佈運算 $f_0...f_{b-1}()$ 中使用串流加密（stream cipher），潛在地包括輕量及潛在地密碼弱串流加密。舉例而言，可利用串流加密 RC4，其中熵重新分佈運算輸入係用作 RC4 金鑰，且 RC4 輸出位元係用作（或被使用以形成）熵重新分佈運算輸出。

加密裝置及解密裝置具有一相對應解密函數 DEC() 的一機密金鑰加密函數（或函數組）ENC()。在某些實施例中，諸如具有固定長度的訊息中，ENC() 及 DEC() 可使用諸如 ECB 或 CBC 模式中的 AES 的傳統加密建構。對其他實施例的 ENC() 及 DEC() 建構係分別相對於第 13 圖及第 14 圖如後說明。

在第 1 及 2 圖中的範例實施例

此段說明用於可檢驗抗洩漏的加密及解密的大致技術的一範例實施例。此第一範例實施例使用金鑰鏈及密文雜湊鏈。

熵

為了方便起見，隨著密碼學中的傳統命名，吾人使用

「明文」一詞指稱待加密的資料。技藝人士將瞭解，此並非必定意味著輸入資料係為人可解讀的，且實際上，並不排除此資料本身在被此專利的技術保護之前，被壓縮、編碼、或甚至被加密。類似地，技藝人士將瞭解「資料」一詞包含經處理的任何數量，且可包括但非限於內容、資料、軟體、程式碼、及任何其他類型的資訊。

給定待保護的一敏感明文資料訊息 D，且具有一分享的基本機密密碼值 K_{ROOT} 的知識，加密裝置實行如第 1 圖中所概述的以下步驟。首先將敏感明文資料 D 分解成一連串 L 片段 D_1、...、D_L（步驟 100），其中(L≧1)，其各者足夠小以符合至記憶體中用於將片段輸入接收器中。此外，此等片段之各者的尺寸應足夠小以滿足應用及實施的洩漏要求。片段可為但非必定具有相同尺寸。此外，其他變數亦可藉由改變金鑰（例如，在 ENC()及 DEC()中）支援未限定尺寸的片段，如將顯示於以下相對於第 13 及 14 圖。

加密裝置亦產生（步驟 101）一偽隨機數 N（將顯示於以下），其可使用作為一訊息識別符（或一前導）用於與 D 的加密連接。舉例而言，偽隨機數（nonce）可使用一真隨機數產生器、一偽隨機數產生器、真及偽隨機數產生器的某些結合、一計數器值或其他（較佳地為獨特的或極少重複的）參數而產生，或藉由從金鑰及/或加密裝置可取得的資料（包括但不限於 D，例如，藉由設定 N 為部份或所有的 D 的雜湊）推導 N 而產生。在第 1 圖中，

對一給定的 K_{ROOT}，用於加密一特定訊息的 N 的值較佳地不用於加密任何其他訊息（或若然，則任何重複使用應被限制為不容易的及/或不常的）。

在以下的範例實施例中，使用偽隨機數 N 形成一訊息識別符 H_1。在最直截了當的實施中，N 供應作為訊息識別符，H_1 可簡單地等於 N。如另一範例，N 供應作為對訊息識別符的一前導，加密裝置可使用函數 h()計算 H_1（步驟 102）為 N 的雜湊。在吾人希望產生一固定尺寸的訊息識別符的情況中，雜湊係為有用的，舉例而言，當為了計算結果效率而運算於較短的數量時准許併入較長資料值（例如，文字字串），或為了計算結果簡化而將可改變長度的資料值轉換為一統一長度訊息識別符，或減少敵手可具有以影響 H_1 的選擇的任何可能性。當然，雜湊僅為產生訊息識別符的一種方式，且技藝人士將理解可利用不同於 h 的函數產生 H_1。

在計算 H_1 之後，加密裝置使用分享的基本機密密碼值 K_{ROOT} 及 H_1 作為輸入一抗洩漏、基於金鑰樹的金鑰推導處理而計算一訊息金鑰 $K_{MESSAGE}$（103）。為了討論的方便，金鑰推導處理在此處呈現為加密的上下文（例如，由加密裝置實行），且更具體而言，在第 1 圖的第一範例加密處理的上下文中。然而，相同的金鑰推導處理亦將被用於第 4 圖的第一範例解密處理中，在此情況中將藉由解密裝置實行。類似地，金鑰推導處理亦將被用於與其他處理連接，包括第 5、11 及 13 圖的範例加密處理，

及第 6、12 及 14 圖的範例解密處理。

一範例金鑰推導處理係圖解於第 2 圖中。處理從樹的一開始點起始，其標示為 K_{START}（201）及一路徑 $P_1...P_Q$（202）。舉例而言，在上述第 1 圖的步驟 103，K_{START} 係為分享的機密金鑰 K_{ROOT} 的值且路徑 $P_1...P_Q$（202）係藉由 H_1 決定。（以下討論將 H_1 轉換為 $P_1...P_Q$）。路徑表明應用至 K_{START} 的一連續的熵重新分佈運算。

在一範例實施例中，訊息識別符 H_1 被分解為 Q 個部份：P_1、P_2、...、P_Q。在一範例分解中，各部份 P_i 係為從 0 至(b-1)的一整數（例如，若 b＝4，則各 P_i 係為兩個位元的值(0、1、2、或 3)）。同樣地，若 b＝2，則各 P_i 係為一單一位元(0 或 1)。因此，路徑部份 $P_1...P_Q$ 可用於藉由應用函數 $f_0()$、$f_1()$、...、$f_{b-1}()$而產生多個中間金鑰引導至如下的 $K_{START,PATH}$ 來表明從 K_{START} 至 $K_{START,PATH}$ 的一特定路徑。首先，將函數 f_{P_1} 應用至 K_{START}（203）以生成一中間金鑰 K_{START,P_1}，接著藉由將 f_{P_2} 應用至 K_{START,P_1} 以生成中間金鑰 K_{START,P_1,P_2}（204），且依此類推，直到最終將 f_{P_Q} 應用至中間金鑰 $K_{START,P_1,P_2,...,P_{Q-1}}$（205）以生成最終推導的金鑰 $K_{START,P_1,P_2,...,P_Q}$（206）。注意各中間金鑰的推導取決於至少一個前趨金鑰（predecessor key）（例如，在第 2 圖的情況中的中間親屬）及訊息識別符的相關部份。為了方便起見，吾人將標示此最終推導的金鑰為標誌 $K_{START,PATH}$（指示由 K_{START} 開始且跟隨 PATH 所達成的金鑰）。同樣地，在第 1 圖步驟 103 的情況中，最終推導的金鑰（指

派至 $K_{MESSAGE}$ 的訊息金鑰）係標示為 K_{ROOT,H_1}，因為開始金鑰實際上係為 K_{ROOT}，且路徑實際上係為 P_1、P_2、...、P_Q，其係為 H_1 的簡單分解。（在替代實施例中，$K_{MESSAGE}$ 可從 K_{ROOT,H_1} 推導出，例如藉由 K_{ROOT,H_1}。以任一方式，$K_{MESSAGE}$ 係基於 K_{ROOT,H_1}）。

於步驟 104，資料片段使用基於該訊息金鑰 $K_{MESSAGE}$ 的至少一個密碼金鑰加密，從輸入片段 $D=D_1$、...、D_L 產生密文 $E=E_1$、...、E_L。用於步驟 104 的一範例實施例係顯示於第 3 圖中，其描繪牽涉計算加密的片段 E_1、...、E_L 的步驟及狀態。

第 3 圖的處理使用 $K_{MESSAGE}$ 計算 L 個個別片段加密金鑰 K_i($i=1$ 至 L)，各金鑰被用於加密機密訊息資料 D 的一相對應片段 D_i($i=1$ 至 L)。首先，將函數 g()應用至 $K_{MESSAGE}$ 以生成用於第一片段的加密金鑰 K_1（302）。接著，將函數 g()應用至 K_1 以生成用於第二片段的加密金鑰 K_2（303），且依此類推。最終，將函數 g()應用至 K_{L-1} 以生成用於最終片段的加密金鑰 K_L（305）。吾人稱此類型的處理為金鑰鏈，因為加密金鑰一個接著一個被鏈鎖。

在對加密 L 個片段決定 L 個金鑰 K_1、...、K_L 之後，片段的加密如下繼續進行。最終(L'th)片段係首先被進行，其中對 ENC()函數的明文輸入（306）係為 L'th 資料片段 D_L，與藉由密碼雜湊的整體明文 $D_1...D_L$ 計算的訊息完整值連鎖。（$D_1...D_L$ 的雜湊的包含係為可選地；實施例可省略此，或連鎖諸如一連串「0」位元或填充的某些其他形

式的其他資料）。此 L'th 明文片段係藉由金鑰 K_L 加密以生成加密的片段 E_L（307）。

接著，第 L-1'th 的片段係藉由將雜湊函數 h()應用至 E_L、將此雜湊值附加至資料片段 D_{L-1}、且使用該結果作為對 L-1'th 片段的加密輸入而於（308）處理。於（309），L-1'th 明文片段接著使用金鑰 K_{L-1} 加密以生成加密的片段 E_{L-1}。此處理對其他片段重複進行。舉例而言，相對應至第二明文片段的加密輸入（310）係由第二資料片段 D_2 跟隨著 $h(E_3)$及第三加密的片段的雜湊而構成，且輸入（310）接著使用金鑰 K_2 加密以生成加密的片段 E_2（311）。最終，相對應於第一明文片段的加密輸入（312）係由第一資料片段 D_1 跟隨著 $h(E_2)$及第二加密的片段的雜湊（311）而構成，且輸入（311）接著使用金鑰 K_1 加密以生成加密的片段 E_1（313）。（接續的片段雜湊無須被加密而作為以上的一變數，例如，E_i 可藉由加密 D_i 而接著將加密結果與 E_{i+1} 的雜湊連鎖而形成）。

加密的片段 $E_1...E_L$ 形成密文 E。第 1 圖中的步驟 104 接著被完成。使用 E_i 的計算結果中的各 E_{i+1} 的雜湊有效率地將加密的值鏈在一起，其供以使得解密裝置能夠在解密有缺陷的片段之前偵測經修改（或有缺陷）的密文片段。吾人將此稱之為「密文雜湊鏈」。在以上所顯示的範例中，各密文片段 $E_i(1<i<L)$取決於下一個密文片段的雜湊，例如，一確認符 V 係用以鑑別第一密文片段（E_1）的雜湊，接著 E_1（若為必須則在對 D_1 解密之後）生成預

期的 E_2 的雜湊。同樣地，E_2（若為必須則在解密之後）生成片段 E_3 的雜湊，且依此類推。

應注意，當所有的資料係在一個片段中（即，$L=1$）（例如，因為輸入訊息係為小的，或利用諸如第 13 圖中所顯示的處理的一加密處理 ENC()）時，第 3 圖的處理仍可實行。對於 $L=1$ 的情況，僅需要 K_1 且 $K_1=g(K_{MESSAGE})$。或者，$K_{MESSAGE}$ 可直接使用作為 K_1，而在此情況中運算 g()可全部一起被省略。如上所述，$D_1...D_L$ 的雜湊（其在此情況中，因為 $L=1$ 所以僅為 D_1）被包括係為可選地。因為此係僅有的片段，所以處理的結果 $E=E_1$。

回頭參照第 1 圖，在資料片段 D_i 已被計算之後，計算一確認符 V，其將使得加密的訊息的認證的接收器在解密之前能夠鑑別密文。首先，計算一值 H_2（105）作為第一加密的片段 E_1 的雜湊。回想第一片段 E_1 併入所有其他片段的雜湊。因此，E_1 的雜湊實際上反應所有片段的內容，包括片段 E_1，且可被用於檢驗並無片段被改變。（可選地，除了 E_1 之外，對雜湊產生的 H_2 的輸入亦可包括關於訊息的額外資訊，例如長度、版本號、發送者身份、N 的值等等）。

接著，加密裝置使用一機密金鑰以計算 V（106），其係為訊息識別符及密文片段 E_i 的一確認符。確認符 V 使用至少一個密文片段的雜湊（例如，雜湊 $H_2=h(E_1)$）及一初始機密（例如，$K_{MESSAGE}$，或如以下段落中所述的其他值）計算。V 的計算可使用第 2 圖中所述的抗洩漏、

基於金鑰樹的推導處理而實行，其具有開始金鑰 K_{START} 係為 $K_{MESSAGE}$ 且路徑係使用 H_2（106）決定。因此，V 的推導包括計算多個中間值引導至 V，其各者取決於至少一個前趨（predecessor）（例如，在第 2 圖的情況係為其親屬值）及雜湊的相關部份（例如，H_2）。應注意函數 $f_i()$、值 b 等等可為（但非必須為）與（103）中使用的相同。此處理導致金鑰 $K_{MESSAGE,H_2}$ 的推導，其係為（或進一步被處理以形成）確認符 V。

以上說明以 $K_{MESSAGE}$ 開始著手推導確認符，但替代實施例可以一不同的值開始。舉例而言，於步驟 104 的金鑰 $K_{MESSAGE}$ 及於步驟 106 的金鑰 $K_{MESSAGE}$ 可彼此不同，但兩者均從 K_{ROOT,H_1} 推導出。同樣地，於步驟 106 使用的金鑰可從於步驟 104 使用的 $K_{MESSAGE}$ 推導出，或反之亦然，或可利用不同的基本金鑰（除了 K_{ROOT} 之外）作為 K_{START}。當然，K_{ROOT} 本身亦可使用作為 K_{START}（例如，若 H_2 係為 N 的一雜湊及/或 H_1 及一或多個密文片段）。

如此專利所使用的確認符係一可檢驗密碼證明，其某些推定的密文係為與一特定訊息識別符相關聯的某些明文訊息資料的一加密的未修改版本，且已經藉由可存取一機密密碼值的一實體產生。建構於步驟 106 的確認符可藉由一接收器（例如一解密裝置）以避免對差動功率分析及相關的外部監控攻擊的易感受性的方式方而便地證實。此外，確認符創造處理（即，步驟 106 的性能）亦使得加密裝置能夠避免對差動功率分析及相關的外部

監控攻擊的易感受性。

在計算確認符之後，加密處理便完成。於步驟 107，結果係為輸出。輸出資料係由必須的資訊（若有任何，例如，偽隨機數 N）所組成，使得一接收器能夠推導訊息識別符、確認符 V、及加密的結果 E（包含加密的片段 E_1、...、E_N）。藉由結合金鑰鏈及密文雜湊鏈，此類型的加密處理能夠生成具有訊息認證的強的密碼的輸出，同時針對加密裝置促進差動功率分析及相關的攻擊的方式，以避免位於加密裝置中機密金鑰的重複使用。加密的結果係以一形式創造，該形式使得一解密裝置能夠以針對解密裝置促進差動功率分析及相關的攻擊的方式，實行解密而無須重複使用機密金鑰。金鑰樹處理限制金鑰以 $K_{MESSAGE}$ 及確認符 V 的形式重複使用，同時密文雜湊鏈方法限制資料加密中所使用的金鑰的使用。

下一段解釋輸出資料如何可接續地由解密裝置解密。

解密

第 4 圖顯示相對應於第 1 及 3 圖的範例加密處理的一範例解密處理。如早先所述，此需要解密裝置及加密裝置兩者具有推導相同的訊息識別符（例如，因為各裝置已知偽隨機數 N，所以其可計算 H_1）、基本機密密碼值 K_{ROOT}、密碼函數 f()、g()及 h()的能力。範例解密處理將使用第 2 圖中描繪的相同的金鑰推導處理（及金鑰鏈）。

範例加密處理於步驟 400 起始，該步驟為獲得（例如，

在一非信任的數位介面上）加密推定的結果（即，訊息識別符（例如，偽隨機數 N）、確認符 V、及包含片段 E_1、...、E_N 的加密的結果 E）。於步驟 401，裝置接著藉由雜湊接收到的偽隨機數 N 計算值 H_1。應注意，除非不正確地接收到偽隨機數，推導的 H_1 將等於用於加密處理中的 H_1。於步驟 402，解密裝置藉由雜湊片段 E_1 計算值 H_2（且若先前在加密期間使用，則關於訊息的其他資訊已經被併入 H_2 的推導中）。於步驟 403，裝置以 $K_{START}=K_{ROOT}$ 及 PATH= H_1，嘗試使用第 2 圖中所述的抗洩漏、基於金鑰樹的金鑰推導處理，計算訊息金鑰 $K_{MESSAGE}$。於步驟 404，裝置藉由使用與加密裝置相同的抗洩漏、基於金鑰樹的金鑰推導處理（例如，第 2 圖中的處理使用金鑰 $K_{START}=K_{MESSAGE}$ 及 PATH= H_2），計算預期的確認符 V'。於步驟 405，計算的值 V'與接收的確認符 V 作比較。若預期的確認符 V'並非與提供的確認符 V 相匹配，則處理以一錯誤終結（步驟 406），因為提供的資料可能已經被毀壞或惡意地修改，或已經發生某些其他錯誤。

若於步驟 405 的檢查係為成功的，則處理移至步驟 407，其中一計數器 i 係從值 1 初始，一金鑰記錄器 K 係以計算 $g(K_{MESSAGE})$ 的結果初始，其中 $g(K_{MESSAGE})$ 的結果係為用於解密第一加密的片段 E_1 的金鑰（即，K_1 的值，其在第 3 圖中標記為 302）。亦於步驟 407，一可變 H 係以 H_2 初始。以下運算接著以第 4 圖所顯示的一迴圈實行。首先，待解密的下一個密文片段的雜湊（即，$h(E_i)$）係

被計算且與期望的雜湊 H 作比較（步驟 408）。若比較失敗，解密的片段已經被改變，所以處理以一錯誤（409）終結且不實行進一步解密。若於步驟 408 的比較成功，則片段 E_i 於步驟 410 使用解密函數 DEC()，以金鑰 K 解密，來生成解密的片段，其被詮釋為含有跟隨著下一個密文片段的聲稱的雜湊的明文 D_i。H 被設定為此聲稱的雜湊值。接著於步驟 411，實行一檢查以確保是否所有的 L 片段已經被解密（即，是否計數器 i 等於 L）。若計數器並未達到 L，則於步驟 412，增加計數器 i 且藉由計算 $K = g(K)$ 更新記錄器 K 為用於下一個片段的解密金鑰，且處理從步驟 408 向前重複。若步驟 411 決定 i 已經達到 L，則於步驟 413 實行一檢查以確保是否 H 等於預期的填充資料（例如，$D_1...D_L$ 的雜湊）。若此檢查失敗，則解密以一失敗條件（414）結束。若檢查成功，則解密處理成功且恢復的解密的輸出 $D = D_1、...、D_L$ 於步驟 415 返回。

應注意在此實施例中，解密處理可以一串流的方式完成（即，解密裝置初始可獲得 N、V 及 E_1，且接著一次接收一個剩餘片段 $E_2、...、E_L$），且仍能夠執行以上概述的步驟。舉例而言，若解密裝置缺乏足夠的記憶體來持有整個訊息，或若解密的資料部份需要在所有資料被接收及解密之前被取得，則串流運算係為實用的。

第二範例實施例

此段說明用於可檢驗抗洩漏的加密及解密的大致技術的一第二範例實施例。相對於使用密文雜湊鏈的第一範例實施例，此第二範例實施例使用明文雜湊鏈。然而，在兩者情況中，金鑰的重複使用於加密裝置及解密裝置兩者被控制，以避免差動功率分析及相關的攻擊。

熵

第二範例實施例由加密裝置的加密係顯示於第 5 圖中，其為了簡潔起見，以一結合的處理圖表及狀態圖表描繪。加密裝置創造或獲得訊息以加密 D 及一訊息識別符 N，其可為一計數器、隨機產生的值、明文雜湊等等。

輸入訊息 D 被劃分成一連串片段 D_1、...、D_L（儘管准許 $L=1$），且此等片段被用於創造如以下的明文片段 B_1、...、B_L。首先，片段 B_1（501）藉由將訊息片段 D_1 與任何所欲的訊息資料的雜湊連鎖而形成（標示為 X，其可包括諸如長度 L、訊息識別符 N、一交易識別符或計數器等等的元素）。接著，藉由將 D_2 與 $h(B_1)$（即，B_1 的雜湊）連鎖而形成 B_2（502）。各個接續至 B_{L-1} 的 B_i 接著藉由將 D_i 與 B_{i-1} 的雜湊連鎖而形成。最終，最後的明文片段 B_L（504）係藉由將 D_L 與 $h(B_{L-1})$連鎖而形成。

處理的接下來的步驟（505-508）使用一金鑰鏈處理對明文片段之各者產生加密金鑰，使得（類似於第一範例實施例）各加密金鑰係直接或間接的基於訊息金鑰。在第二範例實施例中，第一加密金鑰係單純地設定成藉由

計算 h(N)推導出（505）訊息金鑰 $K_{MESSAGE}$ 的值，且接著使用如第 2 圖中所述的抗洩漏、基於金鑰樹的金鑰推導處理，以 $K_{START}=K_{ROOT}$ 及 PATH＝h(N)計算 $K_1=K_{MESSAGE}=K_{ROOT,\,h(N)}$。$i>1$ 的金鑰 K_i 被計算為 $g(K_{i-1})$，其中 g()。因此，第二金鑰 K_2 係為計算 $g(K_1)$（506）的結果。反覆此處理使得 L-1'th 的金鑰 (K_{L-1}) 係計算為 $g(K_{L-2})$（507），且最終片段金鑰 K_L 係計算為 $g(K_{L-1})$（508）。因此，每一金鑰 K_i 係基於訊息金鑰 $K_{MESSAGE}$（例如，等於或由訊息金鑰 $K_{MESSAGE}$ 推導出）。

處理中的下一步驟係以相對應金鑰 K_1、...、K_L 加密明文片段 B_1、...、B_L 之各者，以生成加密的片段 E_1、...、E_L。舉例而言，加密的片段 E_1 係藉由以 K_1 加密 B_1 而創造（509），E_2 係藉由以 K_2 加密 B_2 而創造（510），且依此類推，E_{L-1} 係藉由以 K_{L-1} 加密 B_{L-1} 而創造（511），E_L 係藉由以 K_L 加密 B_L 而創造（512）。加密的結果 E 係由片段 E_1、...、E_L 所組成。

處理中的下一步驟係對加密計算確認符 V（513）。首先，雜湊函數 h()係用以計算 $h(N\|E_1\|...\|E_L)\|h(B_L))$，其中「||」標示連鎖。接著，計算 $Z=h(N\|E_1\|...\|E_L\|h(B_L))$，接著使用抗洩漏基於金鑰樹的金鑰推導處理計算 $K_{ROOT,\,Z}$（例如，如第 2 圖中所述，以 $K_{START}=K_{ROOT}$ 及 PATH＝Z）。接著計算確認符 V 作為金鑰樹結果的雜湊（即，$h(K_{ROOT,Z})$）。最終，提供加密處理的結果，包含 N、$h(B_L)$、E 及確認符 V（514）。

以上的加密處理可利用於系統中，其中輸入資料 D 藉由串流達成，或因為其他原因而 D 無法一次全部被處理（例如，因為記憶體的限制）。在此情況中，加密裝置藉由獲得 N、h(X)及 K_1 而開始著手。此外一運行雜湊計算係以 N 初始。

1. 創造或獲得 N
2. 初始運行雜湊計算
3. 使得 $H = h(X)$
4. 使得 $K = K_{ROOT,h(N)}$
5. 以 N 更新運行雜湊計算
6. 使得 $i = 1$
7. 接收輸入資料 D_i（例如，串流進入）
8. 創造 $B_i = D_i$ 及 H 的連鎖
9. 使得 $H = h(B_i)$
10. 創造 $E_i = ENC(K, D_i)$
11. 以 E_i 更新運行雜湊計算
12. 輸出 E_i
13. 增加 i
14. 若存在更多輸入資料，則回到步驟 7
15. 以 H 更新運行雜湊計算
16. 最終運行雜湊計算且儲存於 Z
17. 計算 $V = h(K_{ROOT,Z})$
18. 輸出 H（其等於 $h(B_L)$）、N、V

解密

解密的處理係圖示於第 6 圖中。於步驟 600，解密裝置（典型地從一非信任介面）接收加密處理的聲稱的結果，即 E、$h(B_L)$、偽隨機數 N、及確認符 V。解密裝置將 E 劃分成 E_1、...、E_L，將一計數器 i 初始為 1，且將一記錄器 H 設定成接收的值雜湊 $h(B_L)$。亦接收或決定訊息 L 的長度（例如，若使用 1 千位元的一片段尺寸，但最後的片段可少於 1 千位元，則 L 係為以千位元捨進的訊息的長度）。於步驟 605，加密裝置計算 $Z=h(N \| E_1 \| ... \| E_L \| H)$，其中「‖」標示連鎖。於步驟（610），加密裝置使用第 2 圖中所述的抗洩漏基於金鑰樹的金鑰推導處理計算 $K_{ROOT,Z}$ 的值，其中根係為 $K_{START}=K_{ROOT}$ 且 PATH＝Z，且接著雜湊結果以生成 $h(K_{ROOT,Z})$。於步驟 620，將 $h(K_{ROOT,Z})$ 與接收的確認符 V 作比較。若結果不等於 V，則存在資料毀壞且處理停止於 611 而不實行任何解密。若檢查成功，則於步驟 620，解密裝置計算 h(N)，接著以計算 $K_{ROOT,h(N)}$ 的結果使用第 2 圖中所述的抗洩漏基於金鑰樹的金鑰推導處理初始金鑰記錄器 K，其中 $K_{START}=K_{ROOT}$ 且 PATH ＝h(N)，且將一計數器 i 設定成 1。

接下來，以下運算以一迴圈實行：於步驟 630，以在金鑰記錄器 K 中的金鑰加密片段 E_i 來產生由一資料片段 D_i 及一雜湊值組成的一明文片段 B_i。於步驟 640，檢查來自解密的目前片段的雜湊。對於第一片段（即，i＝1），雜湊針對 h(X)作比較，其中 X 由與 X 在加密期間相同的

欄位組成。對於在第一者之後的片段（即，i＞1），來自B_i的雜湊相對於先前片段的雜湊作比較（即，$h(B_{i-1})$）。若比較失敗。則解密處理於步驟 641 失敗。否則於步驟 650，B_i的訊息部份（即，D_i）被添加至輸出暫存器（例如，在 RAM 中），且金鑰記錄器 K 藉由計算 g(K)而被推進至下一片段金鑰，接著儲存結果於 K 中。計數器 i 亦增加 1。於步驟 660，i 的值與 L 作比較，且若 i 的值不超過 L，則解密處理迴圈回到步驟 630。否則，解密處理於步驟 670 完成，其中最後明文片段（即，$h(B_L)$）的雜湊係與接收的雜湊 H 作比較。若於步驟 670 的比較失敗（即，該等值並不相等），則發生一錯誤且解密失敗（步驟 671）。否則，結果資料 D_1、...、D_L 於步驟 680 輸出。

在此實施例中，明文的雜湊以含有明文 B_{i-1}的雜湊的明文片段 B_i鏈鎖住。儘管並非絕對必要的抗洩漏，此鏈提供可偵測在解密處理期間發生的任何缺陷額外的特性，因為明文被檢驗為與已經加密者相同。因此，此實施例係有益於用於毀壞的解密處理的潛在可能的環境中。

系統、應用、及變數

至目前為止，本專利已說明用於抗洩漏加密及解密的一大致技術，及與此技術一起的某些範例實施例。此段將說明某些範例系統及/或應用，其中可使用上述技術，以及上述的範例實施例的態樣的額外變數。

安全韌體的載入

第 7 圖顯示用於安全地在一中央處理單元（CPU）上載入敏感韌體的可檢驗抗洩漏密碼的應用，例如，稱之為一晶片上系統（SoC）的部份。為了方便起見，取決於上下文，元件符號可代表一處理中的步驟及/或代表由此等處理步驟使用（或產生）的數量。在此實施例中，SoC 係由一單一整合電路（700）組成，含有一 CPU（703）及各種類型的記憶體。記憶體可包括但非限於程式碼可執行於其上的隨機存取記憶體（RAM）（701）、含有信任的啟動程式碼的唯讀記憶體（ROM）（704）、及持有一分享的密碼機密 K_{ROOT} 的一機密狀態儲存記憶體（702）。金鑰儲存記憶體可使用各種技術實施，例如但非限於使用者空間檔案系統/抗使用者空間檔案系統、備有電池的 RAM、及 EEPROM。SoC 可具有一外部功率輸入（707），其可從一非信任的來源接收功率（例如，潛在地受到敵手的控制及/或觀察）。亦可接收一外部供應的時鐘（708）（且可與 PLLs 一起使用以形成額外的時鐘）。SoC 具有一 AES 引擎的一密碼硬體組件（705）用於資料加密及解密、諸如但非限於基於 SHA-1 或 SHA-256 或 AES 的雜湊函數引擎的一雜湊函數引擎、及抗洩漏的一實施、基於第 2 圖的基於金鑰樹金鑰推導處理，具有使用雜湊函數及/或 AES 函數或其他變數實施的函數 $f_0()$、...、$f_{b-1}()$。對技藝人士而言，在其他實施例中，密碼硬體組件（705）的整體功能或其某些子集可藉由在軟體中（例如，藉由

CPU）實行應為顯而易見的。

在從 ROM 中的信任的啟動程式進行啟動程式之後，SoC 在一非信任介面（706）上從一外部、非信任儲存裝置載入其敏感軟體/資料，其在此實施例中係為快閃記憶體（709）。為了保護敏感軟體/資料避免揭露或未授權的修改，藉由製造商或其他程式碼發行者使用分享的機密密碼值 K_{ROOT}，而使用可檢驗抗洩漏技術加密（例如，如第 1 或 5 圖中所顯示）。加密結果係儲存於快閃記憶體（709）中。SoC 首先從快閃記憶體（709）將加密的程式碼/資料載入其內部 RAM（701）。SoC 接著實行抗洩漏解密（例如，如第 4 圖中所顯示），其中處理係實施於 ROM（704）密碼硬體組件（705）中的信任的啟動程式的程式碼庫中，且使用來自金鑰庫（702）的分享的機密金鑰 K_{ROOT} 實行。若成功，則此處理在 RAM 記憶體（701）中創造一確認的且加密的敏感程式碼/資料影像，其可接著被執行。在解密處理失敗的情況中，RAM 中加密的程式碼/資料（及任何部份解密的程式碼/資料）被刷新且當需要時運算從起始處重新開始。

在此實施例的一可選增強中，安全性係藉由在使用者空間檔案系統中儲存一最小可接受軟體版本號碼、備有電池的記憶體、或其上載入軟體的裝置的其他本端儲存而完成。待載入裝置中的所有軟體載有一版本號碼，且裝置將僅接受具有比最小值更大的一版本號碼的軟體。此外，某些軟體版本應明確地指示 SoC 更新最小可接受

軟體版本號碼，從而避免軟體惡意轉返（rollback）至視為無法接受的一先前版本。上述抗轉返方法可獨立於（即，為附加的）可檢驗抗洩漏運算而實施。或者，抗轉返方法可實施為訊息識別符、確認符、或用於可檢驗抗洩漏運算中的其他安全的數量的部份。

該等技藝人士將輕易地認知 SoC 應用並非限於此處所呈現的特定架構，且可保護具有不同於第 7 圖呈現的實施例的內部架構及/或組件的 SoCs 或其他裝置。

舉例而言，第 8 圖顯示對一安全處理器架構（800）的可檢驗抗洩漏密碼的應用。為了方便起見，取決於上下文，元件符號可代表一處理中的步驟及/或由此等處理步驟使用（或產生）的數量。在此設定中，此裝置包含一 CPU、持有包括一基本機密密碼金鑰 K_{ROOT} 的內部機密狀態的一金鑰庫。諸如但非限於使用者空間檔案系統（801）的非揮發性儲存可用於儲存內部機密狀態。密碼硬體子組件（804）加密及/或整體保護及/或重演保護移出晶片上資料/指令快取（803）至外部非安全 RAM 記憶體（806）的所有資料，且解密及/或整體檢查及/或重演檢查從外部非安全 RAM 記憶體所取得的所有資料。此外，所有程式碼均以形成於非安全快閃（805）中的加密及整體保護而儲存，且當被帶入晶片上資料/指令快取（803）時被解密及整體檢查。先前技術的範例處理器架構（其安全性可透過額外的可檢驗抗洩漏密碼改善）包括但非限於來自 IBM 的 Secure Blue 設計（2006 年 6 月 6 日發佈於 IBM

媒體名為「IBM Extends Enhanced Data Security to Consumer Electronics Products」）及來自 MIT 的 AEGIS 設計（敘述於 AEGIS：用於 Tamper-evident and Tamper-resistant Processing 的架構，超級計算的第 17 次年度年會的活動，2003 年第 160-171 頁）。

可檢驗抗洩漏密碼的使用實質上藉由針對監控攻擊提供保護而改善現有處理器設計的安全性。具體而言，此實施例增強密碼硬體子組件（804）以包括一雜湊函數及一金鑰樹處理能力，其重複使用（例如，AES）一現有安全處理器設計的加密能力，且實施第一範例實施例的步驟及方法，以創造一安全抗洩漏的安全處理器。具體而言，從快取（803）對 RAM 記憶體（806）寫入的任何資料使用抗洩漏加密處理（例如，如第 1 圖中所顯示）加密，且從非信任快閃（805）及非信任 RAM 讀取的程式碼使用第 4 圖中概述的抗洩漏解密處理解密。當資料被寫入一特定片段時，增加相對應於片段的計數器，且計數器的值被併入片段的加密及/或整體檢查創造處理，從而使得牽涉舊資料取代的攻擊能夠被偵測。

FPGA 位元串流載入

載入一現場可程式閘陣列（FPGA）的邏輯通常包含高度敏感商業機密、密碼機密、及/或需要被保護避免揭露或複製的其他敏感資訊。此載入的邏輯或更新的邏輯典型地供應至 FPGA 作為來自一外部來源的一位元串流，

例如但非限於一快閃記憶體裝置或一 CPU 或某些其他來源（907）。某些 FPGAs 含有用於儲存配置資料的非揮發性記憶體，而其他則必須在每一次晶片通電時重新被載入。現有的 FPGAs 具有解密位元串流的能力，典型地使用持有一備有電池的記憶體或本端儲存的一金鑰（例如，使用晶片上快閃、EEPROM、或使用者空間檔案系統）。在將其安裝至呈現於 FPGA 中的可程式化層中之前（或同時），FPGA 解密供應的加密的位元串流。差動功率分析攻擊及相關的外部監控攻擊可針對位元串流解密處理作嘗試，引起如一成功的攻擊可導致的位元串流解密金鑰的揭露及/或位元串流本身的一連串安全性風險。

參照第 9 圖，可檢驗抗洩漏密碼可用於在一 FPGA 上創造一安全位元串流解密功能。在解密之前，敏感位元串流藉由一外部裝置（使用軟體、硬體或某些其結合）使用一抗洩漏加密處理（例如，如第一範例實施例中所述）而加密，產生加密的位元串流。加密的位元串流可被載入（907）一非信任記憶體，例如一外部快閃或硬碟，或從諸如一 CPU 等等的一非信任來源檢索。

在 FPGA 之中，用於抗洩漏解密的密碼機密 K_{ROOT} 被保持於金鑰庫（902）中，其儲存內部機密狀態，且其可使用諸如但非限於使用者空間檔案系統、備有電池的 RAM（902、903）、EEPROM、快閃等等的技術實施。FPGA（900）在介面（906）上接收加密的位元串流。舉例而言，此位元串流可能已經使用第一實施例或第二範例實

施例（相對應於第 1 及 5 圖）任一者加密。

若使用第 1 圖的實施例加密，則 FPGA 首先接收偽隨機數 N、確認符 V、長度 L、及初始片段 E_1。E_1 被儲存於加密的片段暫存器（905）中。使用如上述的一抗洩漏解密處理（例如，見第 4 圖），計算 E_1 的雜湊，且以 K_{ROOT}、L、及雜湊檢驗確認符 V，（若成功）生成 $K_{MESSAGE}$ 或一致命錯誤（在此情況中處理停止）。若成功，則 FPGA 使用片段解密處理組件（904）以在 E_1 上實行抗洩漏解密處理。E_1 的解密生成片段 E2 的雜湊，其被載入、檢驗、且解密。處理一次在一個片段上繼續，直到最終片段被解密且檢驗。若一錯誤發生，則處理停止且所有部份 FPGA 解密的資料被去除。（在失敗時處理可從起始再次重新開始）。使用一或多個狀態記錄器 910 以追蹤位元串流的狀態的載入處理（例如，追蹤處理是否正在進行處理、失敗、或完成）。狀態亦可被輸出用於診斷的目的且由外部組件使用。一旦所有的片段已經被成功的載入，則 FPGA 即刻被配置且可被使用（例如，FPGA 即刻可准許 I/O、時鐘等等應用至載入的位元串流影像）。可防止 FPGA 運算直到位元串流完全被載入（例如，避免透露關於不完整的 FPGA 影像的資訊且避免從不正確 FPGA 配置所生的整體電路的不可預期行為）。

若第 5 圖的第二實施例被用於加密，則 FPGA 首先接收 E、V、N、及 $h(B_L)$，且儲存 E 於一暫存器中。FPGA 的片段解密處理組件 904 接著使用第 6 圖中所述的方法

以檢驗及解密所提供的加密的片段。狀態記錄器 910 被用於追蹤位元串流載入、確認、及解密處理的狀態，及處理停止所導致的任何嚴重的錯誤，及任何部份解密的資料的去除。

網路通訊及其他基於封包的應用

第 10 圖顯示可檢驗抗洩漏密碼的應用，以保護網路通訊避免外部監控攻擊。在此實施例中，諸如裝置 A（1000）、裝置 B（1030）及裝置 C、D、E 等等（1040）的多重網路裝置彼此在一網路（1020）上通訊。某些或所有此等通訊可含有敏感資訊，使其值得用於加密及鑑別資料。再者，某些此等裝置（例如，此實施例中的裝置 A）需要保護其密碼計算結果及金鑰避免外部監控攻擊。

裝置 A 具有一金鑰庫（1001）以儲存必須與其通訊的其他裝置分享的密碼根金鑰的一表。此等金鑰早先可能已經被儲存，或可被協商（例如，使用公用金鑰密碼）。使用公用金鑰密碼以協商金鑰的方法係為先前技術領域中眾所周知的，且被用於諸如 SSL 及 IPSEC 的協定中。此實施例可輕易地被整合至此等或其他協定中。

待加密的帶外封包或資料片段原始來自一應用程式、操作系統、驅動程式、或其他組件（1002）且進入明文封包暫存器（1003）中。各封包接著使用片段加密/解密處理組件（1004）處理，其使用一可檢驗抗洩漏加密方

法（例如，如第 1 圖中所述）加密。用於此加密的根金鑰係為裝置 A 及目的裝置分享的金鑰，其從金鑰庫（1001）獲得。對此處理，訊息識別符偽隨機數 N 可為任何（較佳地）獨特值，包括一計數器。舉例而言，偽隨機數可等於一封包識別符、具有可能併入額外最重要位元以避免溢流的一 TCP 數列、一值的雜湊、一隨機值等等。對各封包，抗洩漏加密運算產生一加密的片段及一確認符 V。偽隨機數可被傳送或可為隱含的（例如，基於早先接收的封包號）。加密的片段、V、及任何其他所需的資料被聚集成一外出封包，且移至網路介面組件（1006），且接著至網路（1020）用於路由至適當的目的裝置。

對於帶內加密的封包，假設發送裝置已經實行如上述的加密。此等封包藉由網路介面組件（1006）從網路（1020）被接收，且接著移至密文封包暫存器（1005）。各封包接著藉由片段加密/解密處理組件（1004）處理，其中實行一抗洩漏解密處理（例如，如第 4 圖中所述）。對於此解密處理，(i)介於接收及發送裝置之間的分享的金鑰（例如，K_{ROOT}或用於推導 K_{ROOT}的一前導）係從金鑰庫（1001）獲得，(ii)偽隨機數 N 係從封包恢復，否則被決定，(iii)確認符係針對 N 及加密的封包檢驗，(iv)若確認符正確，則封包資料被解密。裝置 A 及發送裝置之間的分享的密碼機密可使用作為 K_{ROOT}。若解密或確認符失敗，則封包被中斷。否則，在成功解密之後，解密

的結果可被提供至應用程式、操作系統、驅動程式等等。

此處理係概述於第 11 及 12 圖中。第 11 圖圖示可檢驗封包層級抗洩漏加密處理，且第 12 圖圖示相對應的解密處理。可檢驗封包層級抗洩漏加密處理係為以下步驟：給定具有分享一基本密碼值的 K_{ROOT} 來源及目的之一輸入封包資料 D（1100），則在步驟 1101 中產生一訊息識別符（例如，使用在封包 D 中呈現的一隨機來源及/或資訊，及/或諸如與通訊協定相關聯的一序號某些封包識別符）。對於 TCP/IP 通訊，N 可從一分區識別符、序號（可選地附加額外地最重要位元以避免翻覆）、來源埠、目的埠、及/或其他值而建構。接著於步驟 1102，計算 N 的雜湊。（可選地，此步驟可被省略，且可使用 N 取代在 $K_{MESSAGE}$ 推導中的 h(N)）。接續在步驟 1103 中，使用第 2 圖中所述的抗洩漏基於金鑰樹的金鑰推導處理以 $K_{START} = K_{ROOT}$ 及 PATH＝h(N)計算 $K_{MESSAGE} = K_{ROOT, h(N)}$。輸入封包資料 D 以金鑰 $K_{MESSAGE}$ 加密來生成加密的結果 E（1104）。

接著計算 E 的雜湊（1105）（例如，使用 SHA-256）。接著，使用第 2 圖中概述的抗洩漏基於金鑰樹的推導處理，以 $K_{START}=K_{MESSAGE}$ 及 PATH＝h(E)計算用於加密的確認符 V 作為 $K_{MESSAGE,h(E)}$（1106）。最終，形成包括 V、E、及 N 的輸出封包（或任何其他資訊，若為任何必須而能夠接收以修復 N）（1107）。輸出資料 E 接著以一封包傳輸至一遠端裝置（例如，在網際網路上的一遠端電腦）。

如一可選的最佳化，若加密裝置具有多重封包被暫存

用於發送，則其同時可加密多重封包，使得僅需要單一確認符用於所有的封包。舉例而言，可實行加密處理如第 3 圖中所顯示，其中各片段 D_i 係為一封包。以此方式結合封包減少金鑰樹必須用於發送者及接收者兩者的運算的次數。

第 12 圖中圖示一相對應可檢驗封包層級抗洩漏解密處理。給定包括 V、E、N 的一加密的封包（或足夠修復 N 的資料，例如，一序號）及分享的密碼機密 K_{ROOT}（1200），解密處理如以下進行：首先，計算 h(N)的值（1201）（或，若加密裝置直接使用 N，則此步驟被省略）。接著計算 E 的雜湊（1202）。接下來於步驟 1203，使用第 2 圖中所繪的抗洩漏基於金鑰的方式，以 $K_{START}=K_{ROOT}$ 及 PATH=h(N)計算 $K_{MESSAGE}=K_{ROOT, h(N)}$（1204）。接下來使用第 2 圖中概述的抗洩漏金鑰樹處理，以 $K_{START}=K_{MESSAGE}$ 及 PATH=h(E)計算 $V'=K_{MESSAGE, h(E)}$。接續地，解密裝置檢查是否 V'＝V（1205）。若其並非相等，則處理停止於此封包且封包被丟棄（1206）。若檢查成功，則 E 以 $K_{MESSAGE}$ 解密來生成明文封包 D（1207）（例如，使用第 14 圖所顯示的 DEC()處理）。

智慧卡運用

可檢驗抗洩漏加密及解密可實施於智慧卡中（例如，與協定連接，其中智慧卡必須以保衛差動功率分析及相關的外部監控攻擊的方式實行加密及/或解密）。此等系

統及協定的範例包括但不限於用於付費電視訊號的解密的金鑰（控制碼字）的推導、付費（包括離線付費）、識別/確認網路登入、行動電話 SIM 卡、及轉乘車票。此專利中所揭露的範例密碼技術可用於確保智慧卡中的機密金鑰被保護以避免外部監控攻擊，同時實行此協定。智慧卡亦可用於實施較大系統中所利用的部份或全部的抗洩漏加密或解密處理，例如，若智慧卡實施第 3 圖的基於金鑰樹的金鑰推導處理，使得 K_{START} 絕對無須離開智慧卡。

互相鑑別應用

在許多應用中，兩個以上的裝置需要彼此鑑別及/或在其之間交換敏感資訊。此等協定的範例應用包括但不限於：(i)一印表機及一墨水匣之間的鑑別以確保兩者裝置均為真的且並非偽冒的；(ii)一數控器（set-top box）及一智慧卡之間的鑑別以確保組件係為真實的（例如，避免偷取的視訊解密金鑰的導入）；(iii)一車庫門及一開啟器之間的鑑別；(iv)無金鑰入口系統（例如，可用於汽車中），其鑑別金鑰（例如，在解鎖門或發動引擎之前）；(v)由經常被偷的物品實行的鑑別協定（例如，汽車音響、GPS 單元、手機等等）以避免被偷或從操作中被竄改裝置；及 (vi)入口系統，例如可在安全建築物中找到者，其在准許進入之前鑑別金鑰/票卷。在此等應用中，介於裝置之間的挑戰回應協定傳統上被用於互相鑑

別及設定一分享的機密金鑰兩者，用於敏感資訊的交換。單純的協定實行此等鑑別，同時抗拒的 DPA 可使用此專利的方法建構以實行任何所需的加密及解密運算。舉例而言，一裝置可透過其能力使用此專利中揭露的技術證實其鑑別性以供應一有效的確認符及/或解密一訊息。

以內部片段金鑰改變的片段加密及解密

此段說明 ENC()及 DEC()運算的範例變數，其可代替傳統加密處理（例如，在 ECB 或 CBC 模式中的 AES）被用於實施範例實施例（例如，如第 3 圖的步驟 320、第 4 圖的步驟 410、第 5 圖的步驟 509、第 6 圖的步驟 630、第 11 圖的步驟 1104、第 12 圖的步驟 1207 所顯示）。分別顯示於第 13 及 14 圖的 ENC()及 DEC()中的變數，密碼金鑰經常改變，用於更甚的安全性。特定而言，額外的密碼金鑰更新在將一資料片段 D_i 加密為 E_i 時發生（反之亦然）。因此，吾人將此等變數稱之為實施內部片段金鑰改變。

除了對 ENC()及 DEC()的改變之外，第一及第二範例實施例中運算的餘數可如先前所述實施。舉例而言且非限制，涉及初始訊息金鑰 $K_{MESSAGE}$、確認符 V、等等的運算無須被改變。

第 13 圖顯示用於加密資料片段的一 ENC()運算的一範例實施例。第 14 圖顯示一 DEC()運算的一相對應範例實

施例。在此實施例中，使用區塊密文 AES 以密文區塊鏈（CBC）模式建立運算，但技藝人士應清楚亦可使用其他區塊密文或原始加密/解密或加密模式。

對片段 i 加密處理的輸入係為片段金鑰 K_i（1301）及資料片段 D_i（1310）。輸入資料片段 D_i（1310）被劃分為子片段 $D_{i,1}$（1311）、$D_{i,2}$（1312）等等。第 13 及 14 圖顯示資料片段 D 被劃分為 3 AES 區塊的子片段，然而亦可使用其他尺寸，且當然亦可利用有別於 AES 的其他演算法。（較小的子片段增加總運算結果，然而較大的子片段造成金鑰被用於更多運算，增加資訊洩漏的可能性）。片段金鑰 K_i 以一雜湊運算 m()變換，生成 $K_{i,1}$（1302），其係為第一子片段 $D_{i,1}$ 的金鑰。若使用一初始向量(IV)（1314），則其以 $D_{i,1}$ 的第一 AES 區塊 XORed。（若不使用 IV，則此 XOR 步驟可省略。若使用 IV，則其可被鑑別，例如，藉由將其併入確認符運算，或藉由從諸如一訊息識別符的一確認的值推導的 IV）。(D_i XOR IV)的第一位元使用片段金鑰 $K_{i,1}$（1302）以 AES（1315）加密，形成密文子片段 $E_{i,1}$（1320）的第一部份。此密文部份亦以子片段 $D_{i,1}$（1311）的下一個位元 XORed，而生成另一 AES 輸入，其接續著使用片段金鑰 $K_{i,1}$（1302）加密，以產生子片段 $D_{i,1}$（1311）的下一個部份。實行一相似的密文區塊鏈以形成第三 AES 加密的輸入，其亦以金鑰 $K_{i,1}$ 實行。此等三個 AES 運算的結果係為密文子片段 $E_{i,1}$（1320）。第四個 AES 運算係實行於下一個資料子片段

$D_{i,2}$（1312）的第一區塊上，且使用標示為 $K_{i,2}$（1303）的一新的金鑰，其藉由將 m()應用至 $K_{i,1}$（1302）而推導出。從處理 $D_{i,1}$的最終密文成為用於 $D_{i,2}$（1312）的第一部份的 IV（1317）。加密處理繼續直到已經加密所有 s 資料子片段的所有區塊，終極生成加密的子片段 $E_{i,2}$（1321）、...、$E_{i,s}$（1322），且其中對各子片段使用 m()推導一新的金鑰。最終，密文子片段被集合以形成最終密文片段 E_i（1330）。

參照第 14 圖，解密處理 DEC()係為相反的 ENC()處理。子金鑰 $K_{i,1}$（1402）、$K_{i,2}$（1403）等等係從金鑰 K_i（1401）透過如上述用於加密的相同處理使用 m()推導出。加密的片段 E_i被劃分成子片段，其各者包含以子金鑰解密的一或多個 AES 輸入。在各解密運算之後，適當的 IV（若有）或先前密文以資料 XORed。最終資料被集合以形成子片段（1420、1421、1432 等等），其依序集合以形成 D_i（1430）。

以上的 ENC()及 DEC()處理係為涉及驟然金鑰改變的範例，以便提供更大的洩漏忍耐度。可使用其他片段加密及解密方法，包括在 ECB、CBC、或計數器（例如，Galois 計數器）模式中串流密文及/或區塊密文（例如，RC4、SEAL、AES、DES、三重 DES 等等）的應用。對於將相同的金鑰應用至一片段中的所有資料的此等運算，其可具有在加密之前限制各片段的尺寸的益處（例如，藉由如第 3 圖所顯示將資料劃分成子片段），以便限制以各

金鑰實行的運算的次數，從而減少一敵手可以各金鑰實行而觀察的運算的次數。

通訊通道

此處所述的資料交換可以一廣泛的可能方式完成。舉例而言但非限制，可使用傳統匯流排/介面（例如，I2C、JTAG、PCI、串聯 I/O（包括 USB）、PCI 快速模組、乙太網路等等）、無線協定（例如，802.11 群、藍牙、細胞服務電話協定、ISO14443 等等）、及晶片內連接（例如，APB、與其他正反器直接連接等等）所有。對於以上之各者，發送裝置及接收裝置將具有適當的介面（例如，上述類型的介面），其可發送、接收、或（適當地）發送及接收。

解密之前資料確認的替代形式

至此呈現的範例實施例已使用抗洩漏基於金鑰樹的金鑰推導處理（例如，如第 2 圖所圖示）來計算密文的一確認符，其可在解密之前安全地被檢驗。儘管此處理係良好合適於一廣範圍的應用，用於創造一值的其他技術可供相似的作用，且在某些設定中可為適當的。舉例而言在某些實施例中，加密處理無須抵抗外部監控（但解密處理需要此抵抗力）及/或可呈現演算法層級對策用於公共金鑰數位簽署處理（例如美國專利 6,304,658 中所述者）。對於此等系統，數位簽署（數位簽章）運算可用以

建構在解密時可被檢驗的一值，來確保密文未被修改。舉例而言，數位簽章可鑑別訊息識別符及至少一個加密的片段。公共金鑰數位簽署演算法的範例包括但非限於RSA、DSA、及橢圓曲線 DSA 變數（包括但非限於EC-DSA）。一數位簽章的檢驗無須任何敏感資訊，且因此可在解密之前實行。然而，此彈性必須付出在加密裝置中需要公共金鑰簽署邏輯且在解密裝置中需要公共金鑰檢驗邏輯的代價。一確認符（或確認符替代物）亦可能包含多重對稱確認符、公共金鑰簽章、或其他元素。

非連續的片段金鑰推導

片段金鑰（例如，第 3 圖中的 K_1、K_2、...、K_L）及子片段金鑰（第 13 圖中的 $K_{i,1}$、$K_{i,2}$等等）並非必須被連續推導。舉例而言，金鑰可以一階層式金鑰圖案推導，或更一般地可為任何先前金鑰的一函數，或可使用金鑰樹建構從 K_{ROOT}獨立地推導出，或金鑰可使用其他金鑰及金鑰樹建構的某些結合而推導出。

資料變換及計算的記錄

資料變換及運算的順序可被改變。舉例而言，在第 1、3 及 4 圖中所述的第一範例實施例顯示加密處理從最後片段 D_L 進行至第一片段 D_1，其具有各片段 D_i 含有 i+1'th 片段的加密結果 E_{i+1} 的雜湊。對第一加密的片段 E_1 計算一分開的確認符（例如，見步驟 106）。此方式可具有對

第 4 圖中所顯示的解密裝置的益處，因為其無須在解密之前暫存整體的加密結果，然而加密裝置卻必須如此。

或者，加密裝置可從 D_1 開始加密片段且結束於 D_L，其中各片段 D_{i+1} 含有早先片段的加密 E_i 的雜湊。在此範例中，片段 D_1（舉例而言）由等於雜湊函數的輸出長度的尺寸的 0 的一字串延伸，以表明其為第一片段。接著使用 $PATH = h(E_L)$ 計算使用金鑰樹創造的一確認符。對於此變數，解密處理係與第 4 圖相似，但以相反的方向從最後加密的片段至第一者而進行。因此，加密裝置不再需要暫存資料片段，儘管解密裝置如今卻必須如此。

用於雜湊的額外確認符的替換

儘管某些範例顯示雜湊為鑑別接續加密的片段的資料片段，但接續的片段可取而代之的承載其自身的獨立確認符。舉例而言，第 3 圖顯示第一資料片段（312）承載一雜湊 $h(E_2)$ 以確認片段 E_2 未被改變。然而，此雜湊並非總是必須的，且在某些情況中可被省略（例如，若下一個片段取而代之的承載一確認符）。此某種程度上簡化加密，但增加計算時間，因為需要計算且檢查更多確認符。在串流應用或若儲存/記憶體被限制，則額外的付出計算可為有道理的，而給予避免接續資料必須可取得且暫存的優點。

雜湊的變化

在某些圖示中，諸如第 3 圖中的 h()的一單一運算被多次運用及/或被用於不同用途。此等運算一般無須均為相同函數。舉例而言，不同步驟可利用不同雜湊函數。

雜湊函數的輸出可被縮短，與其他雜湊函數輸出結合，或者透過後處理修改。舉例而言，SHA-2 產生一 256 位元輸出雜湊，但欲使用一較短的訊息識別符（例如，160-、128-、80-或 64-位元）。函數 h()內部可使用 SHA-2 且僅返回其結果的某些位元。

運算順序的變化

某些範例實施例標明資料元素被連鎖或結合的一特定順序。舉例而言，第 3 圖中的步驟 303-312，資料 D_i 以雜湊 $h(E_{i+1})$連鎖。資料片段在被雜湊之前連鎖成一連串的其他範例包括第 5 圖，元素 501-504 及 513、第 3 圖的步驟 306。此等特定順序僅為一可能順序的一個範例，且可在替代實施例中使用各種其他資料順序。

基於金鑰樹的推導的變化

若運算（例如 f_i）係為不可逆轉的，則可能使用除了金鑰樹的頂端的一值作為開始值。相似地，計算的值可被快取（例如，若訊息識別符係一計數器，則初始運算通常將不從一個訊息改變至下一者，所以無須重新計算）。

錯誤偵測及/或修正

本技術領域中眾所周知在一密碼裝置的運算中產生為注入缺點的一結果的不正確輸出可生成有關敏感資料及金鑰的資訊。在實際操作中，可檢查密碼運算以幫助避免不正確計算結果的釋放，其可洩漏機密。舉例而言，一簡單且有效的技術係為實行密碼運算兩次，理想地使用兩個（或更多）獨立硬體處理及實施，具有一比較器檢驗兩者（或所有）產生完全相同的結果。若由單元產生的結果不匹配，則比較器將避免任一結果被使用及/或觸發其他錯誤條件。在個別密碼運算中（例如雜湊步驟），錯誤偵測及/或錯誤修正邏輯亦可被利用以幫助避免或偵測密碼運算不正確地實行的情況。

本專利所揭露的技術可額外地針對在加密及解密處理上的缺點注入攻擊的某些類型提供某些固有的抵抗。在加密處理期間，於基於金鑰樹的金鑰推導處理期間導入的一限制或部份缺點，歸因於在此處理中的熵重新分佈函數的使用，而將產生隨機、不可預期的結果。具體而言，訛誤的中間值將典型地藉由接續熵重新分佈函數混合，其將限制敵手使用缺陷結果發動攻擊的能力。

相同的，在解密期間，在密文或訊息識別符處理中導入的缺點或錯誤將大致導致確認符被拒絕。具有明文雜湊鏈的第二實施例提供進一步抵抗，因為明文片段被獨立地鑑別在輸出之前修正。當然，可額外地使用運算的檢查及其他眾所周知的缺點偵測技術。

亦可併入諸如一 POST(通電自我測試)及隨機號碼測試的自我診斷函數，以檢驗密碼函數及隨機號碼產生能力未受損。

額外主機環境及形式因素

以上已經說明使用可檢驗抗洩漏密碼的數種範例系統及應用。然而技藝人士將理解，上述的技術並非限於特定主機環境或形式因素。更確切地，其可被用於廣泛的各種應用，包括但非限於：特定應用積體電路(ASICs)、現場可程式閘陣列(FPGAs)、晶片上系統(SoC)、微處理器、安全處理器、安全網路裝置、所有類別的密碼智慧卡(包括但非限於與 ISO 7816-1、ISO 7816-2、及 ISO 7816-3(「ISO 7816-相容的智慧卡」)實質上相容的智慧卡)；非接觸且基於靠近的智慧卡及密碼卷(包括但非限於與 ISO 14443 實質上相容的智慧卡)；儲存值的卡及系統；密碼安全的信用及金融卡；顧客忠誠卡及系統；密碼鑑別的信用卡；密碼加速器；賭博及賭注系統；安全密碼晶片；抗竄改的微處理器；軟體程式(包括但非限於用於個人電腦、處理器等等上的程式，及可載入或安裝於密碼裝置上的程式)；金鑰管理裝置；銀行金鑰管理系統；安全網頁伺服器；防衛系統；電子付款系統；微付款系統及計量器；電話預付卡；密碼識別卡及其他識別檢驗系統；用於電子匯款的系統；自動出納機器；銷售站的點；發證系統；電子證章；門入口系統；使用密碼金鑰的所有

類別的實體鎖；用於解密電視訊號的系統（包括但非限於廣播電視、衛星電視、及有線電視）；用於解密加密的音樂及其他音頻內容的系統（包括在電腦網路上分佈的音樂）；用於保護所有類型視頻訊號的系統；內容保護及複製保護系統（例如，用於避免未授權複製或使用電影、音頻內容、電腦程式、電子遊戲、影像、文字、資料庫等等）；手機拌碼及鑑別系統（包括電話鑑別智慧卡）；安全電話（包括用於此電話的金鑰儲存裝置）；密碼PCMCIA卡；可攜式密碼卷；及密碼資料審核系統。

以上所有描繪範例實施例及可檢驗抗洩漏密碼的應用，其中相關的變化、加強及修改將從所揭露的精神及範疇的上下文而為顯而易見的。因此，由此專利所保護的發明不應限於以上揭露，而應由此處隨附的申請專利範圍推斷。

【圖式簡單說明】

第1圖顯示使用金鑰及密文雜湊鏈的可檢驗、抗洩漏加密的整體處理的一範例實施例。

第2圖顯示從一分享的密碼機密 K_{START} 開始，且繼續通過一路徑 $P_1...P_Q$ 的一抗洩漏、基於金鑰樹的金鑰推導處理的一範例實施例。第2圖的金鑰推導處理係可用於與第1及3圖的第一範例加密處理及第4圖的第一範例解密處理連接。其亦可用於與第5、11及13圖的其他範

例加密處理及第6、12及14圖的其他範例解密處理連接。

第3圖顯示用於加密的一抗洩漏金鑰及雜湊鏈處理的一範例實施例（例如，包含第1圖中所顯示的整體加密處理的部份）。

第4圖顯示使用相對應於第1圖（及第3圖）的加密處理的金鑰及密文雜湊鏈的一可檢驗、抗洩漏解密處理的一範例實施例。

第5圖顯示使用金鑰及明文雜湊鏈的可檢驗、抗洩漏加密的一範例實施例。

第6圖顯示使用相對應於第5圖的加密處理的金鑰及明文雜湊鏈的可檢驗、抗洩漏加密的一範例實施例。

第7圖顯示一環境，其中可檢驗、抗洩漏密碼運算被用於將韌體載入一晶片上的一系統。

第8圖顯示一環境，其中可檢驗、抗洩漏密碼運算被用於一保衛 CPU 晶片之中，其中諸如快閃記憶體及/或RAM的外部記憶體係為非信任的。

第9圖顯示一環境，其中可檢驗、抗洩漏密碼運算被用於將一位元流影像載入一現場可程式閘陣列。

第10圖顯示一環境，其中可檢驗、抗洩漏密碼運算被用於基於封包的網路通訊裝置中。

第11圖顯示用於可檢驗封包層級抗洩漏加密的一處理的一範例實施例，其可用於第10圖所述的環境中，以及其他實施例中。

第12圖顯示相對應於第11圖所述的加密處理的可檢

驗封包層級抗洩漏加密的一處理的一範例實施例。

第 13 圖顯示一範例 ENC()運算的一範例實施例，使用具有內部片段金鑰變化的加密區塊鏈（cipher block chaining，CBC）。

第 14 圖顯示一範例 DNC()運算的一範例實施例，相對應於第 13 圖的加密運算，使用具有內部片段金鑰變化的加密區塊鏈（CBC）。

【主要元件符號說明】

700 整合電路

701 隨機存取記憶體

702 機密狀態儲存記憶體

703 CPU

704 唯讀記憶體

705 密碼硬體組件

706 非信任介面

707 外部功率輸入

708 外部供應時鐘

709 快閃記憶體

800 安全處理器架構

801 使用者空間檔案系統

803 晶片上資料/指令快取

804 密碼硬體子組件

805 非安全快閃

806 非安全 RAM 記憶體

900 FPGA

902 RAM

903 電池

904 片段解密處理組件

905 片段暫存器

906 介面

907 其他來源

910 狀態記錄器

1000 裝置 A

1001 金鑰庫

1002 其他組件

1003 明文封包暫存器

1004 片段加密/解密處理組件

1005 密文封包暫存器

1006 網路介面組件

1020 網路

1030 裝置 B

1040 裝置 C、D、E

七、申請專利範圍：

103年1月16日修正替換頁(本) P.1-6

1. 一種用於加密明文資料同時限制密碼金鑰的重複使用的裝置，其包含：

一硬體單元，其經配置以

(a) 藉由計算多個連續的中間金鑰，從一內部機密狀態及一訊息識別符推導一訊息金鑰，該多個連續的中間金鑰係以至少一部分該內部機密狀態開始且引導至該訊息金鑰，待推導的各連續金鑰係基於至少一部分的該訊息識別符及一先前金鑰；

(b) 基於至少該訊息金鑰使用該一或多個密碼金鑰，以加密該明文資料的一或多個片段，來產生一或多個加密的資料片段；

(c) 從至少一個該加密的資料片段計算一密碼雜湊；

(d) 從至少一機密值及該密碼雜湊推導一確認符，其中該推導之步驟包括以下步驟：從該機密值開始計算多個連續中間值，各連續值係為至少基於該等值的一先前一個及該密碼雜湊的一部分；及

(e) 輸出該一或多個加密的資料片段及該確認符。

2. 如申請專利範圍第 1 項之裝置，其中在(a)步驟中的各該中間金鑰的該推導經配置成包括以下步驟：選擇一熵分佈運算，接著應用該熵分佈運算至該先前金鑰。

3. 如申請專利範圍第 2 項之裝置，配置成該熵分佈運算係為取決於以下至少一者的一密碼雜湊運算：(i)該先前金鑰；及(ii)相對應於至少一部分該訊息識別符的一值。

4. 如申請專利範圍第 1 項之裝置，配置成(a)步驟中的該訊息金鑰係藉由跟隨一路徑通過一金鑰樹（key tree）而計算，該路徑包含該訊息識別符。

5. 如申請專利範圍第 1 項之裝置，配置成(b)步驟包括以下步驟：從該訊息金鑰藉由實行多個熵分佈運算而推導多個該等密碼金鑰；且配置成：(i)該等密碼金鑰之一者係藉由應用一熵分佈函數至該訊息金鑰而計算；及(ii)多個連續的密碼金鑰之各者係藉由應用一熵分佈函數至該等密碼金鑰之一先前一者而計算。

6. 如申請專利範圍第 1 項之裝置，配置成各該連續的中間值係為一密碼雜湊運算的結果，該密碼雜湊運算僅由以下作決定：(1)該連續中間值的該直系親屬值；及(2)該加密的資料片段雜湊的一或兩個位元。

7. 一種用於加密明文資料同時限制密碼金鑰的重複使用的裝置，其包含：

一硬體單元，其經配置以

(a) 藉由計算多個連續的中間金鑰，從該內部機密狀態及一訊息識別符推導一訊息金鑰，該多個連續的中間金鑰係以至少一部分該內部機密狀態開始且引導至該訊息金鑰，待推導的各連續金鑰係基於至少一部分的該訊息識別符及一先前金鑰；

(b) 基於至少該訊息金鑰使用該一或多個密碼金鑰，以加密該明文資料的一或多個片段，來產生一或多個加密的資料片段；

(c) 至少基於一或多個該加密的資料片段使用一機密金鑰以計算一密碼檢驗值，且可用於檢驗一或多個該加密的資料片段；及

(d) 輸出該一或多個加密的資料片段及該密碼檢驗值。

8. 如申請專利範圍第 7 項之裝置，配置成該密碼檢驗值係為一數位簽章，其鑑別該等加密的資料片段之至少一者。

9. 一種用於加密資料同時限制密碼金鑰的重複使用的裝置，其包含：

一硬體單元，其經配置以

(a) 接收一或多個加密的資料片段及一密碼檢驗值，且獲得與其相對應的一訊息識別符；

(b) 檢驗該密碼檢驗值以決定訊息識別符或至少一個該等加密的資料片段是否已經被修改，包括以下步驟：

(i) 從至少一個該加密的資料片段計算一密碼雜湊；

(ii) 從至少一機密值及該密碼雜湊推導出一預期的確認符，其中該推導的步驟包括以下步驟：從該機密值開始計算多個連續的中間值，其中各連續值係至少基於該等值的一先前一者及該密碼雜湊的一部分；及

(iii) 將該推導出的預期的候選確認符與該接收的密碼檢驗值作比較；

(c) 藉由計算多個連續的中間金鑰，從該內部機密狀態

及該訊息識別符推導一訊息金鑰，該多個連續的中間金鑰係以至少一部分該內部機密狀態開始且引導至該訊息金鑰，其中各連續金鑰係基於至少一部分該訊息識別符及一先前金鑰而推導出；及

(d) 基於至少該訊息金鑰使用該一或多個密碼金鑰以解密該加密的資料的一或多個經檢驗的片段，來產生一或多個明文資料片段。

10. 如申請專利範圍第 9 項之裝置，其中(c)步驟中的各該中間金鑰的該推導係配置成包括以下步驟：選擇一熵分佈運算，接著應用該熵分佈運算至該先前金鑰。

11. 如申請專利範圍第 10 項之裝置，配置成該熵分佈運算係為取決於以下至少一者的一密碼雜湊運算：(i)該先前金鑰；及(ii)相對應於至少一部分該訊息識別符的一值。

12. 如申請專利範圍第 9 項之裝置，配置成(c)步驟中的該訊息金鑰係藉由跟隨一路徑通過一金鑰樹（key tree）而計算，該路徑包含該訊息識別符。

13. 如申請專利範圍第 9 項之裝置，配置成(d)步驟包括從該訊息金鑰藉由實行多個熵分佈運算而推導多個該等密碼金鑰；且配置成：(i)該等密碼金鑰之一者係藉由應用一熵分佈函數至該訊息金鑰而計算；及(ii) 多個連續的密碼金鑰之各者係藉由應用一熵分佈函數至該等密碼金鑰之一先前一者而計算。

14. 如申請專利範圍第 9 項之裝置，配置成各該連續的中間值係為一密碼雜湊運算的結果，該密碼雜湊運算僅由以

下作決定：(1)該連續中間值的該直系親屬值；及(2)該加密的資料片段雜湊的一或兩個位元。

15. 如申請專利範圍第 9 項之裝置，配置成存在僅一個明文資料片段及僅一個加密的資料片段，且配置成該密碼雜湊計算結果進一步包括一資料片段長度。

16. 如申請專利範圍第 9 項之裝置，配置成該預期的確認符確認一雜湊，其中：(i)該雜湊係取決於併入一第二加密的資料片段的該雜湊的一第一加密的資料片段；及(ii)該等第二及接續的加密的資料片段係各配置成併入該下一個加密的資料片段的該雜湊。

17. 如申請專利範圍第 9 項之裝置，其中該裝置係一 FPGA，且其中該裝置係配置成該明文資料包括一 FPGA 位元流。

18. 如申請專利範圍第 9 項之裝置，進一步配置成使用收容於該裝置中的一處理器以執行至少一部分該明文資料。

19. 如申請專利範圍第 9 項之裝置，其中該裝置係包含一處理器的一晶片，且其中該裝置係配置成藉由該晶片保衛待載入的資料避免一外部記憶體進入一晶片上快取，且其中該資料被載入為由該晶片先前加密者。

20. 一種用於加密明文資料同時限制密碼金鑰的重複使用的裝置，其包含：

一硬體單元，其經配置以

(a) 藉由計算多個連續的中間金鑰，從該內部機密狀態及一訊息識別符推導一訊息金鑰，該多個連續的中間金鑰係以至少一部分該內部機密狀態開始且引

導至該訊息金鑰，待推導的各連續金鑰係基於至少一部分的該訊息識別符及一先前金鑰；

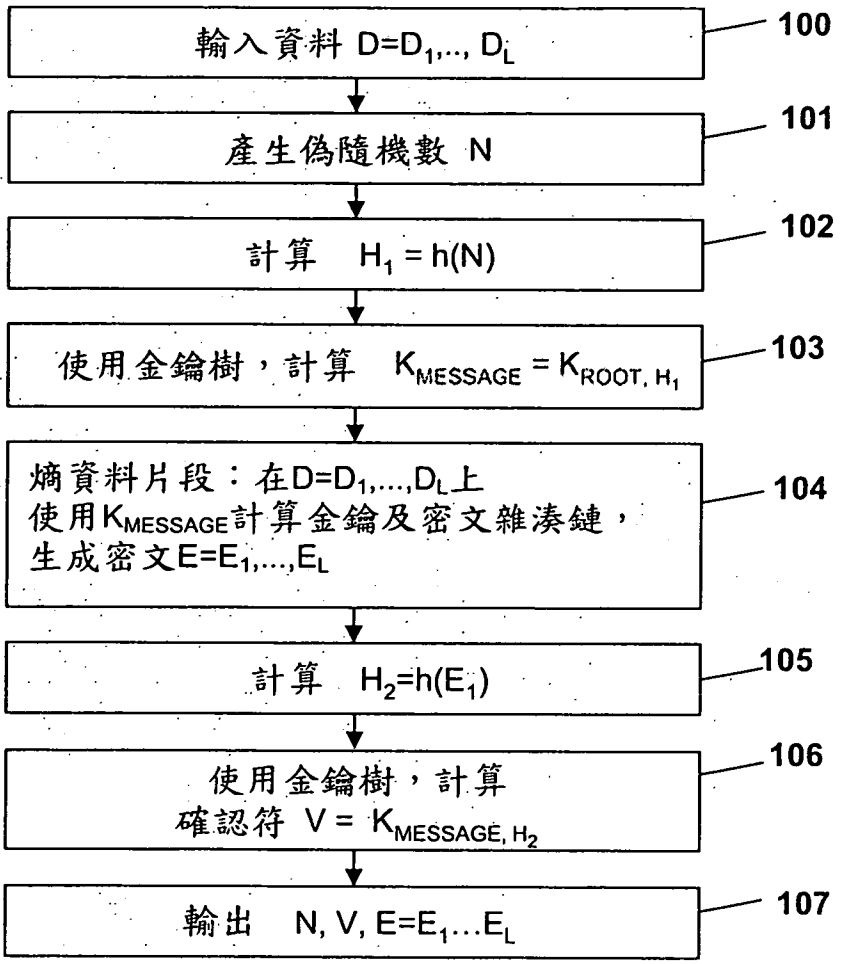
(b) 基於至少該訊息金鑰使用該一或多個密碼金鑰，以加密該明文資料的一或多個片段，來產生一或多個加密的資料片段；

(c) 至少基於一或多個該加密的資料片段使用一機密金鑰以計算一密碼檢驗值，且可用於檢驗一或多個該加密的資料片段；及

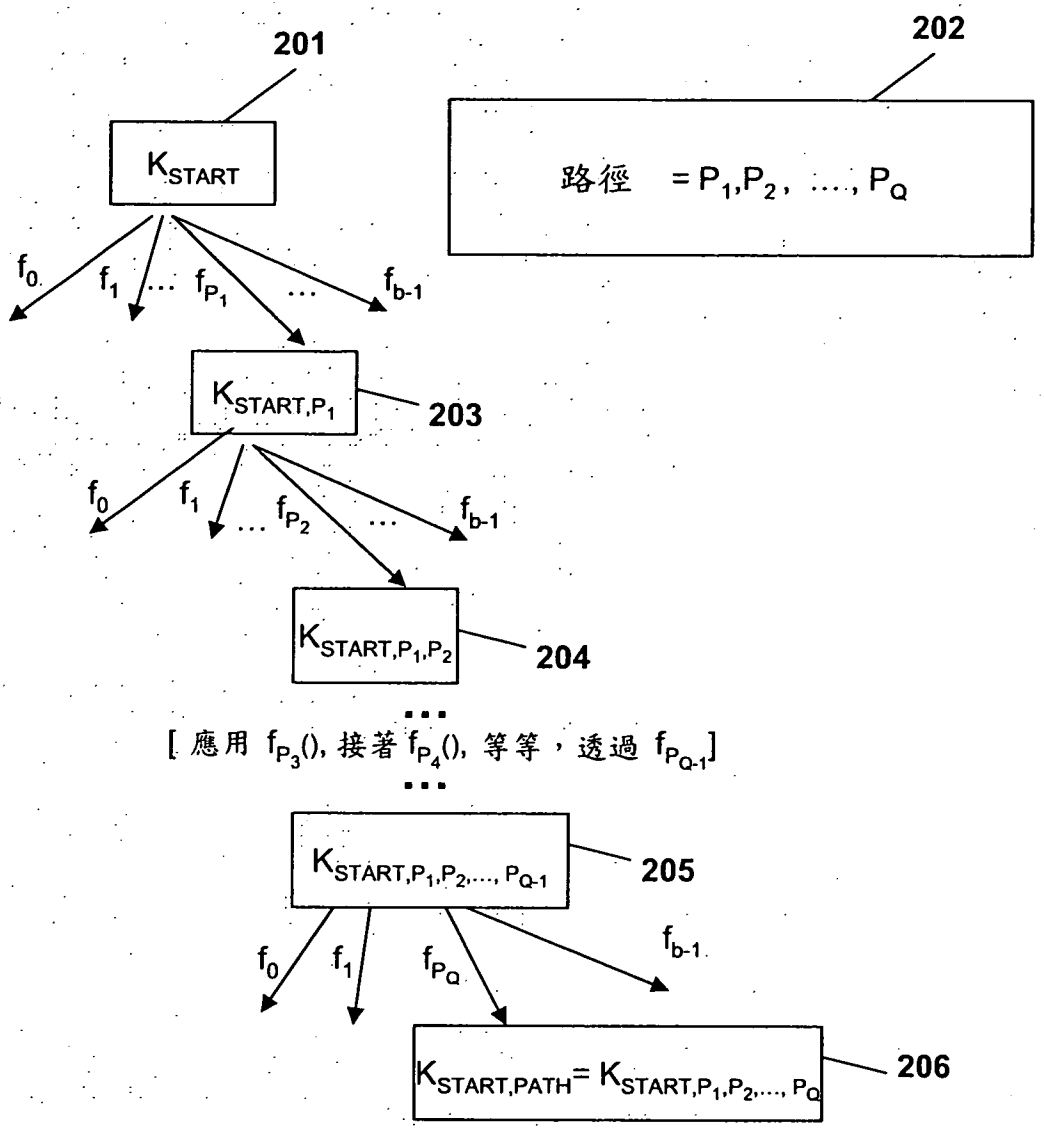
(d) 輸出該一或多個加密的資料片段及該密碼檢驗值。

21. 如申請專利範圍第 20 項之裝置，配置成該密碼檢驗值係為一數位簽章，其鑑別該等加密的資料片段之至少一者。

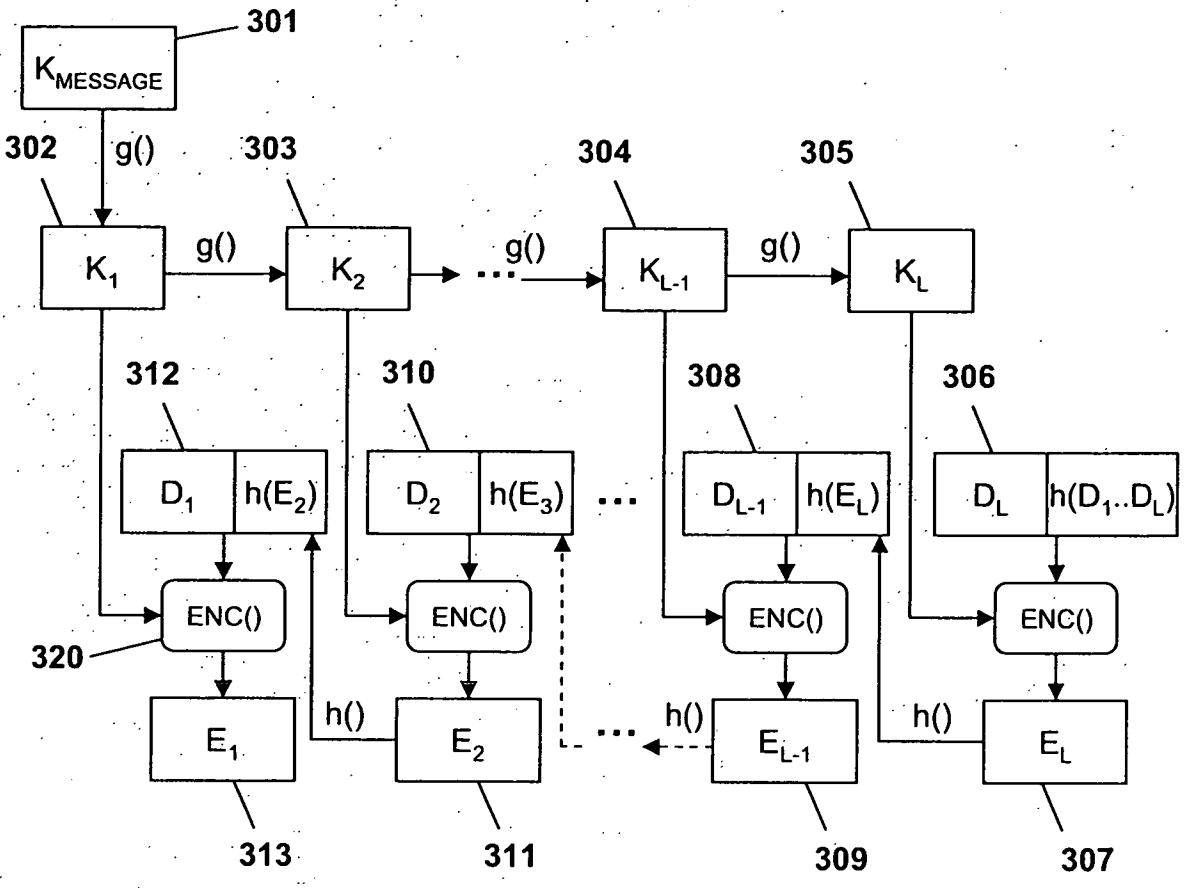
八、圖式：


輸入資料 D=D1,.., DL
100
產生偽隨機數 N
101
計算 H1 = h(N)
102
使用金鑰樹，計算 KMESSAGE = KROOT, H1
103
熵資料片段：在D=D1,...,DL上
使用KMESSAGE計算金鑰及密文雜湊鏈，
生成密文E=E1,...,EL
104
計算 H2=h(E1)
105
使用金鑰樹，計算
確認符 V = KMESSAGE, H2
106
輸出 N, V, E=E1...EL
107

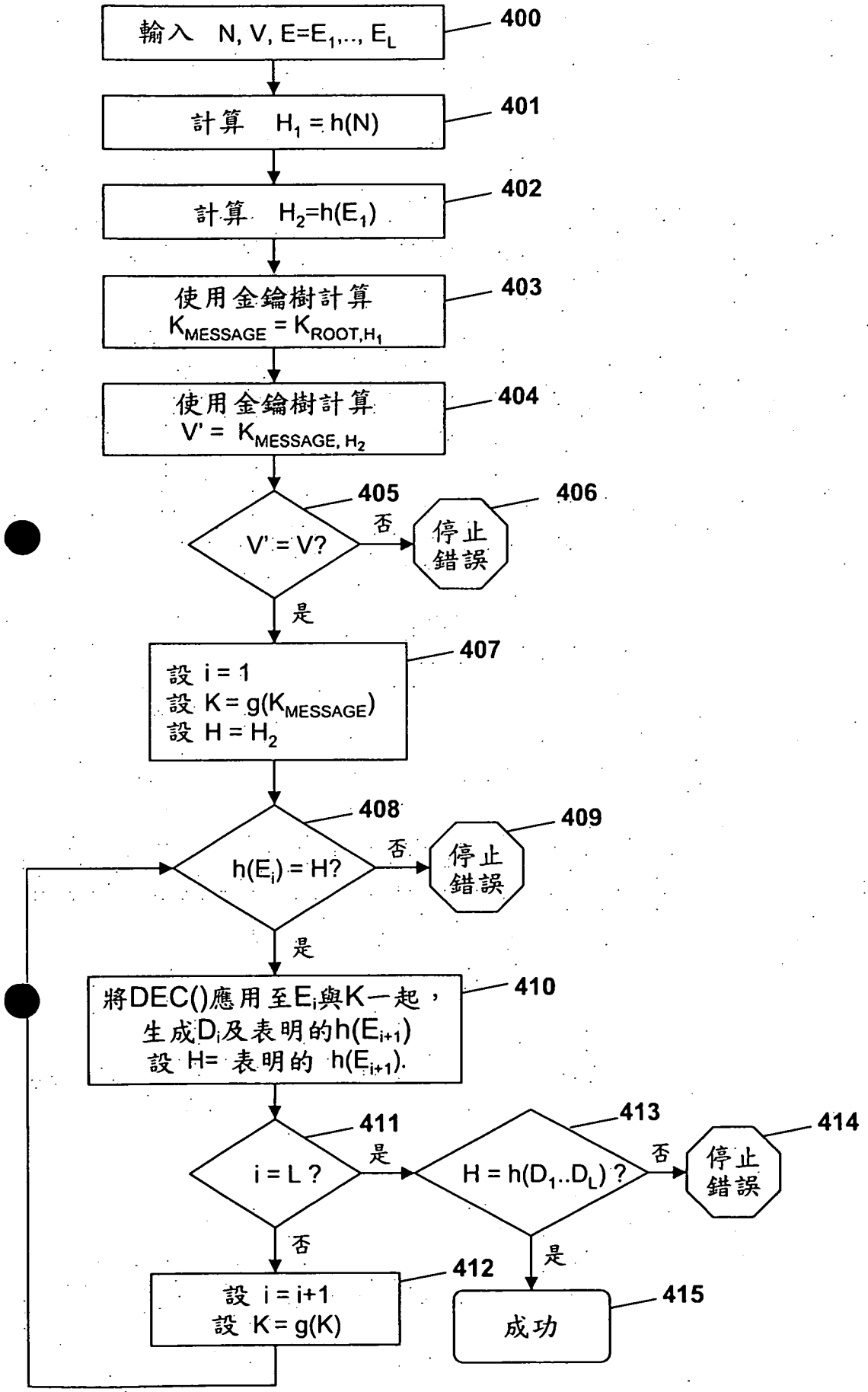
第1圖


201
202
K_{START}
路徑 $= P_1, P_2, \ldots, P_Q$
f_0 f_1 … f_{P_1} … f_{b-1}
K_{START,P_1}
203
f_0 f_1 … f_{P_2} … f_{b-1}
K_{START,P_1,P_2}
204
[應用 $f_{P_3}()$, 接著 $f_{P_4}()$, 等等，透過 $f_{P_{Q-1}}$]
$K_{START,P_1,P_2,\ldots,P_{Q-1}}$
205
f_0 f_1 f_{P_Q} f_{b-1}
$K_{START,PATH} = K_{START,P_1,P_2,\ldots,P_Q}$
206

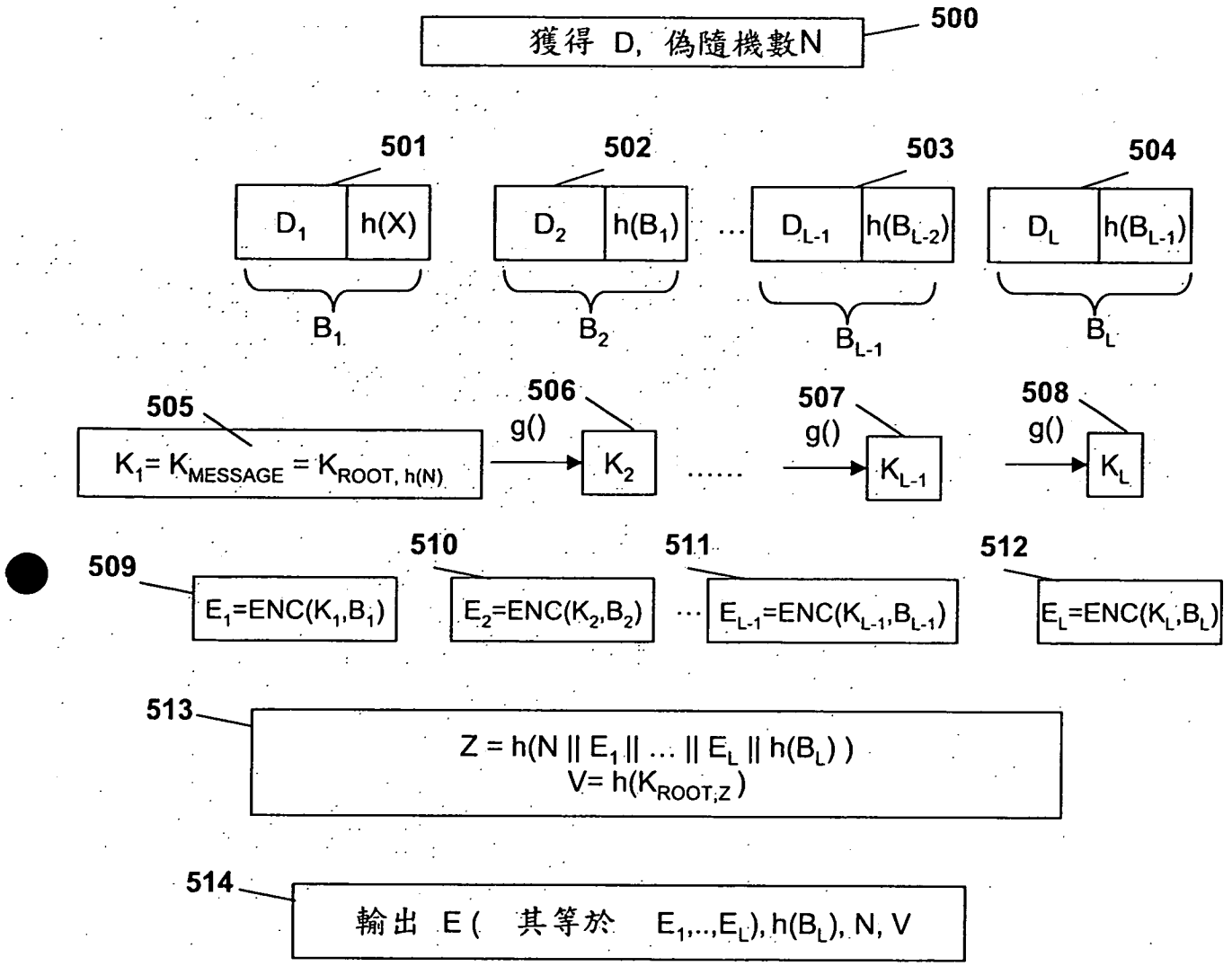
第2圖


301
$K_{MESSAGE}$
302
g()
303
304
305
K_1
g()
K_2
g()
K_{L-1}
g()
K_L
312
310
308
306
D_1 $h(E_2)$
D_2 $h(E_3)$
D_{L-1} $h(E_L)$
D_L $h(D_1..D_L)$
ENC()
ENC()
ENC()
ENC()
320
h()
h()
h()
E_1
E_2
E_{L-1}
E_L
313
311
309
307

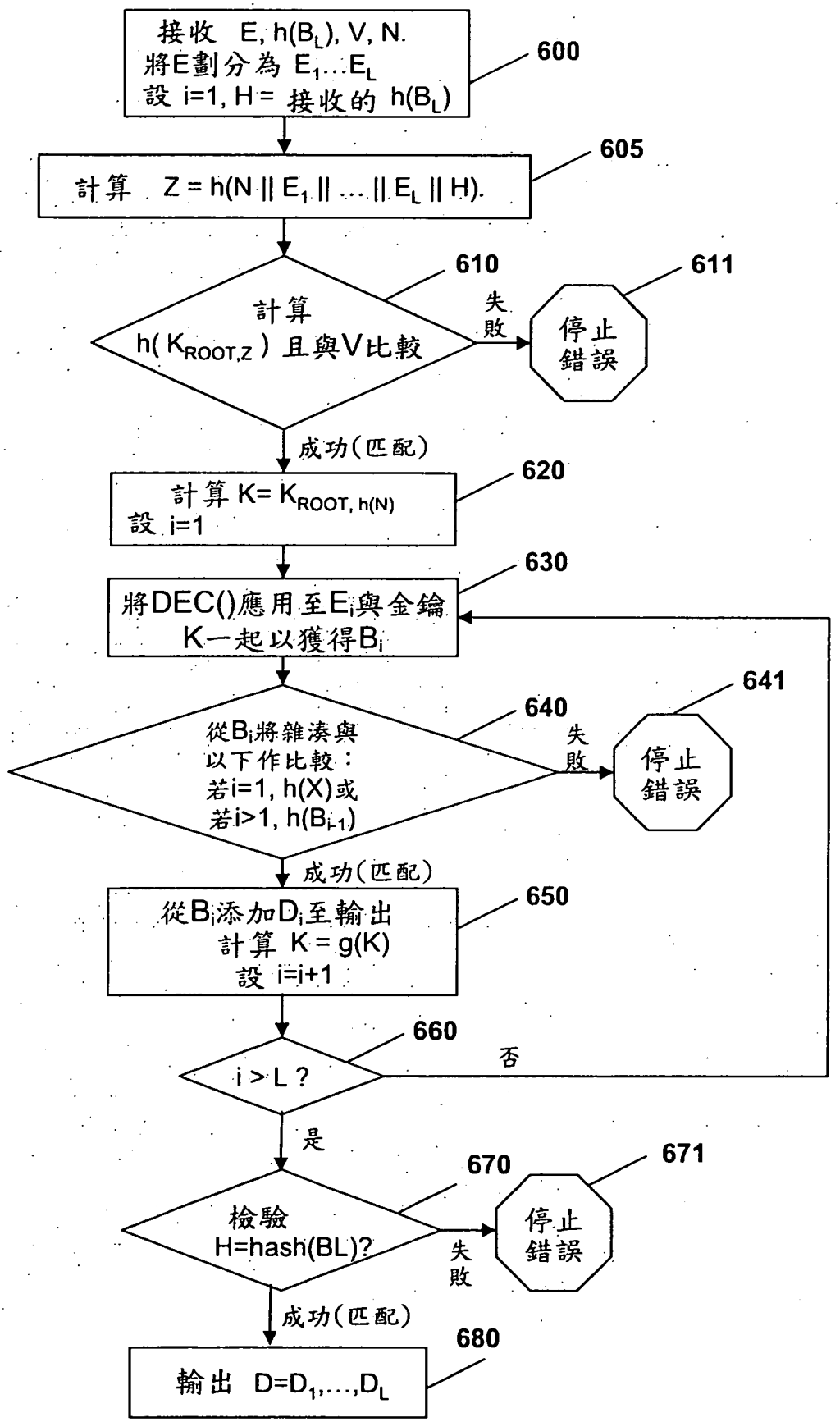
第3圖


輸入 N, V, E=E1,.., EL
400
計算 H1 = h(N)
401
計算 H2=h(E1)
402
使用金鑰樹計算
KMESSAGE = KROOT,H1
403
使用金鑰樹計算
V' = KMESSAGE, H2
404
V' = V?
405
否
停止錯誤
406
是
設 i = 1
設 K = g(KMESSAGE)
設 H = H2
407
h(Ei) = H?
408
否
停止錯誤
409
是
將DEC()應用至Ei與K一起，
生成Di及表明的h(Ei+1)
設 H= 表明的 h(Ei+1).
410
i = L ?
411
是
H = h(D1..DL) ?
413
否
停止錯誤
414
否
設 i = i+1
設 K = g(K)
412
是
成功
415

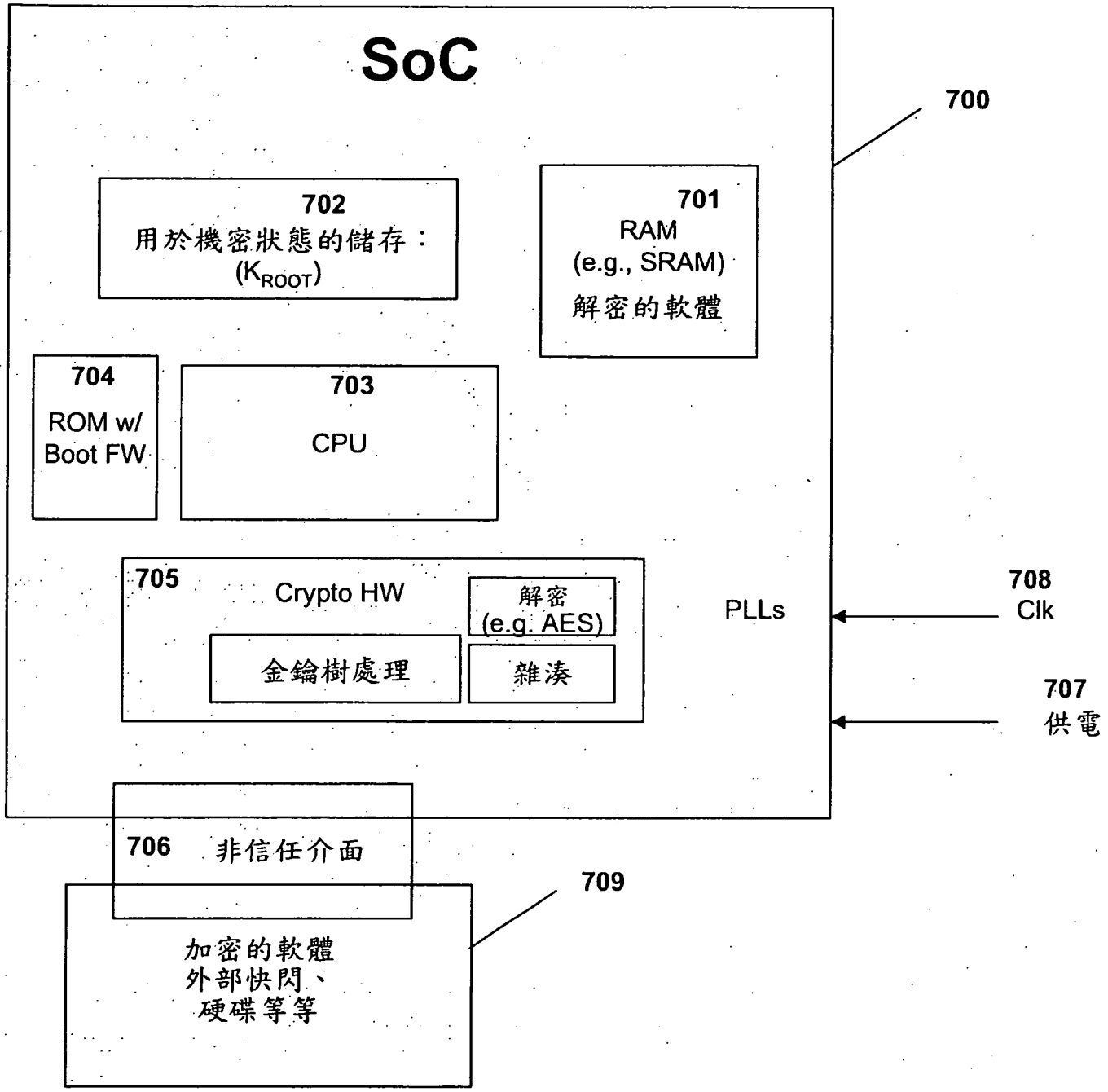
第4圖


500
獲得 D, 偽隨機數N
501
502
503
504
D1
h(X)
D2
h(B1)
DL-1
h(BL-2)
DL
h(BL-1)
B1
B2
BL-1
BL
505
K1= KMESSAGE = KROOT, h(N)
506
g()
K2
507
g()
KL-1
508
g()
KL
509
E1=ENC(K1,B1)
510
E2=ENC(K2,B2)
511
EL-1=ENC(KL-1,BL-1)
512
EL=ENC(KL,BL)
513
Z = h(N || E1 || ... || EL || h(BL))
V= h(KROOT,Z)
514
輸出 E(其等於 E1,..,EL), h(BL), N, V

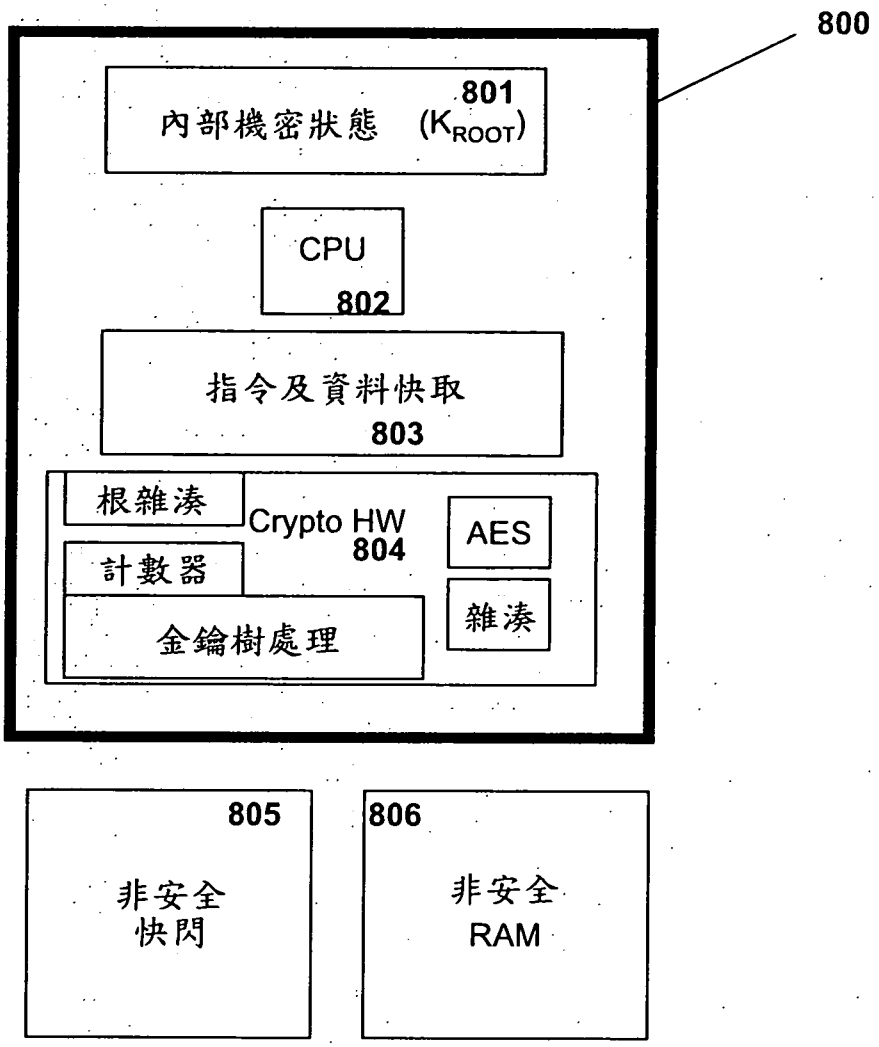
第5圖


接收 E, h(B_L), V, N.
將E劃分為 E_1...E_L
設 i=1, H = 接收的 h(B_L)
600
計算 Z = h(N || E_1 || ... || E_L || H).
605
計算
h(K_ROOT,Z) 且與V比較
610
失敗
停止
錯誤
611
成功(匹配)
計算 K= K_ROOT, h(N)
設 i=1
620
將DEC()應用至E_i與金鑰
K一起以獲得B_i
630
從B_i將雜湊與
以下作比較：
若i=1, h(X)或
若i>1, h(B_i-1)
640
失敗
停止
錯誤
641
成功(匹配)
從B_i添加D_i至輸出
計算 K = g(K)
設 i=i+1
650
i > L ?
660
否
是
檢驗
H=hash(BL)?
670
失敗
停止
錯誤
671
成功(匹配)
輸出 D=D_1,...,D_L
680

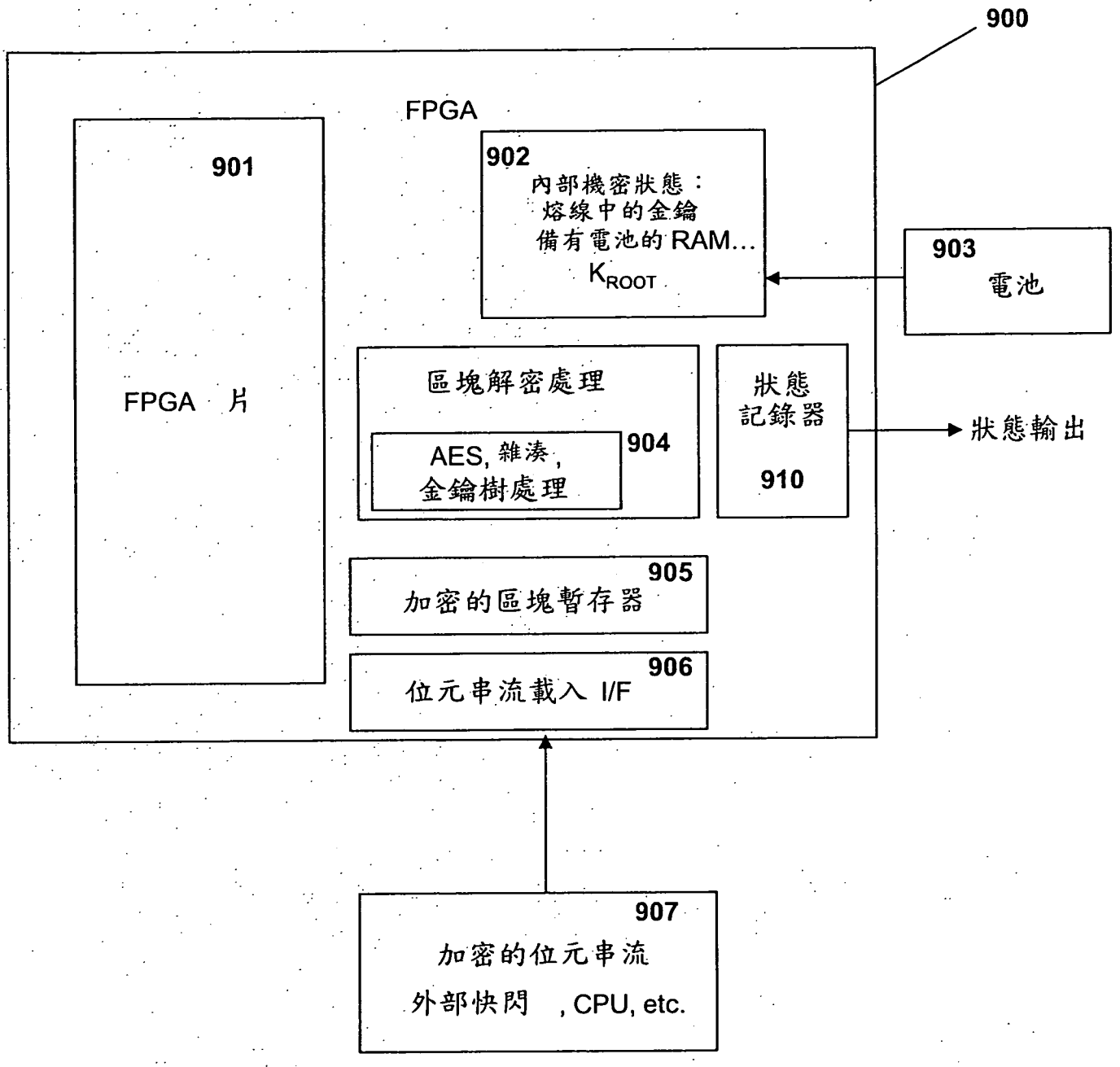
第6圖


SoC
700
702
用於機密狀態的儲存：
(K_{ROOT})
701
RAM
(e.g., SRAM)
解密的軟體
704
ROM w/
Boot FW
703
CPU
705
Crypto HW
解密
(e.g. AES)
金鑰樹處理
雜湊
PLLs
708
Clk
707
供電
706
非信任介面
709
加密的軟體
外部快閃、
硬碟等等

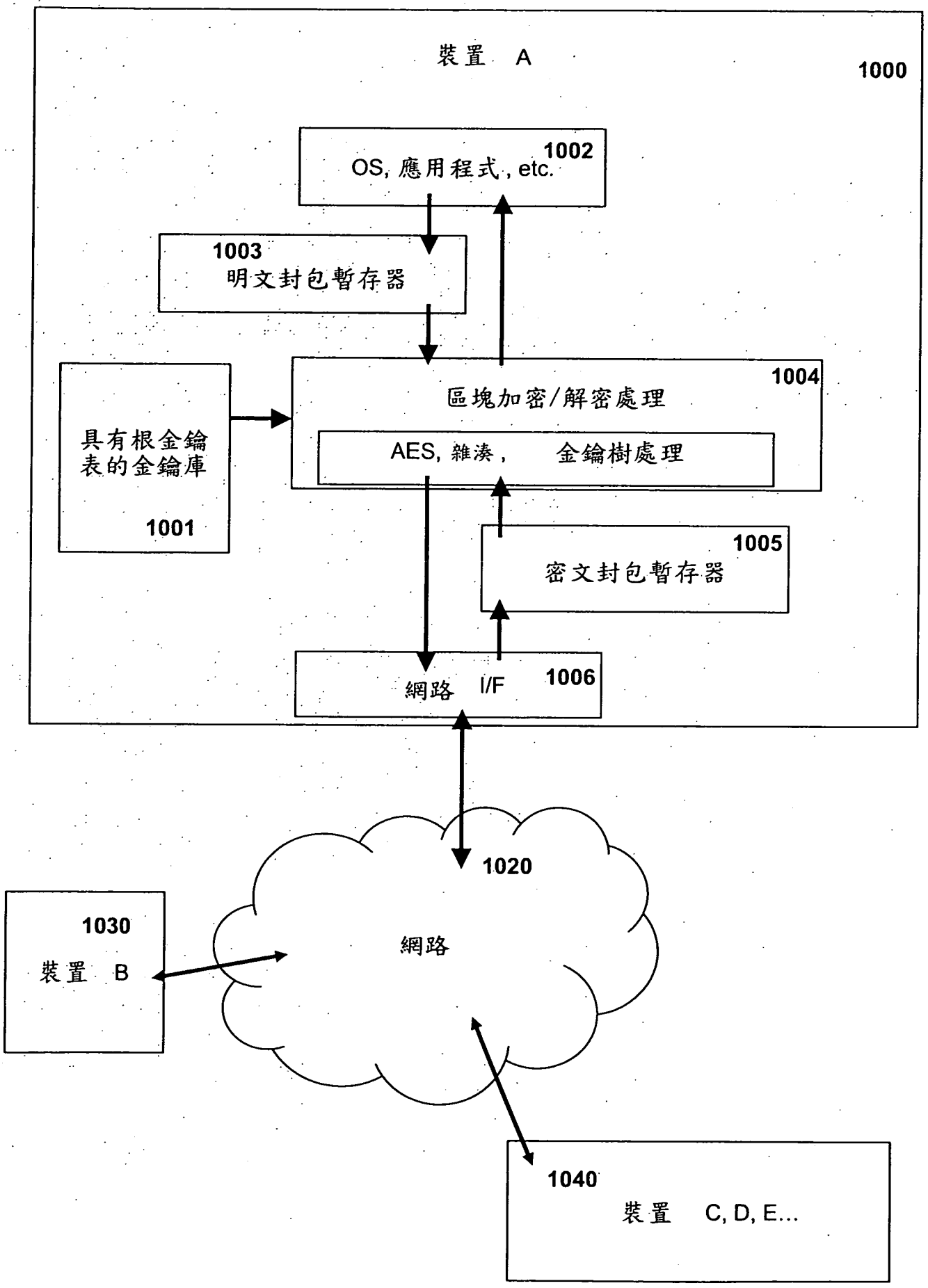
第7圖


800
801
內部機密狀態 (K_{ROOT})
CPU
802
指令及資料快取
803
根雜湊
Crypto HW
804
AES
計數器
雜湊
金鑰樹處理
805
非安全
快閃
806
非安全
RAM

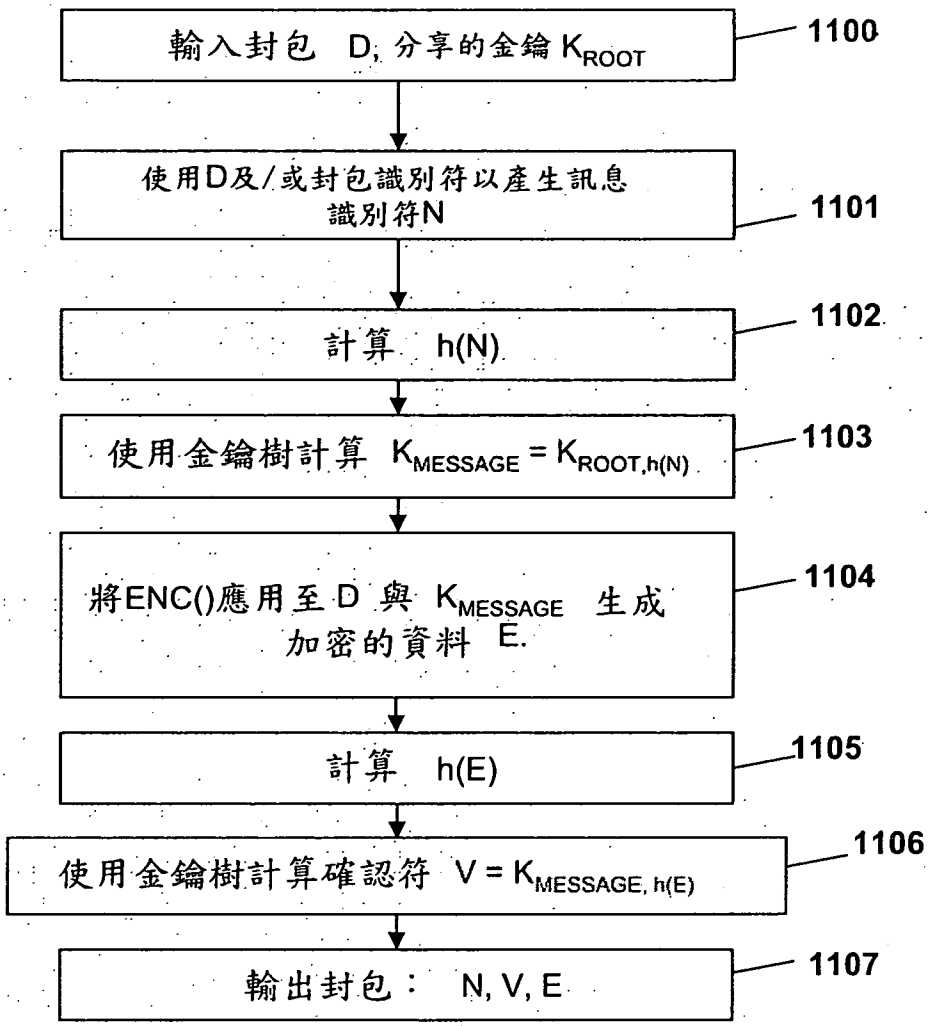
第8圖


900
FPGA
901
FPGA 片
902
內部機密狀態：
熔線中的金鑰
備有電池的 RAM...
K_{ROOT}
903
電池
區塊解密處理
AES, 雜湊,
金鑰樹處理
904
狀態
記錄器
910
狀態輸出
905
加密的區塊暫存器
906
位元串流載入 I/F
907
加密的位元串流
外部快閃 , CPU, etc.

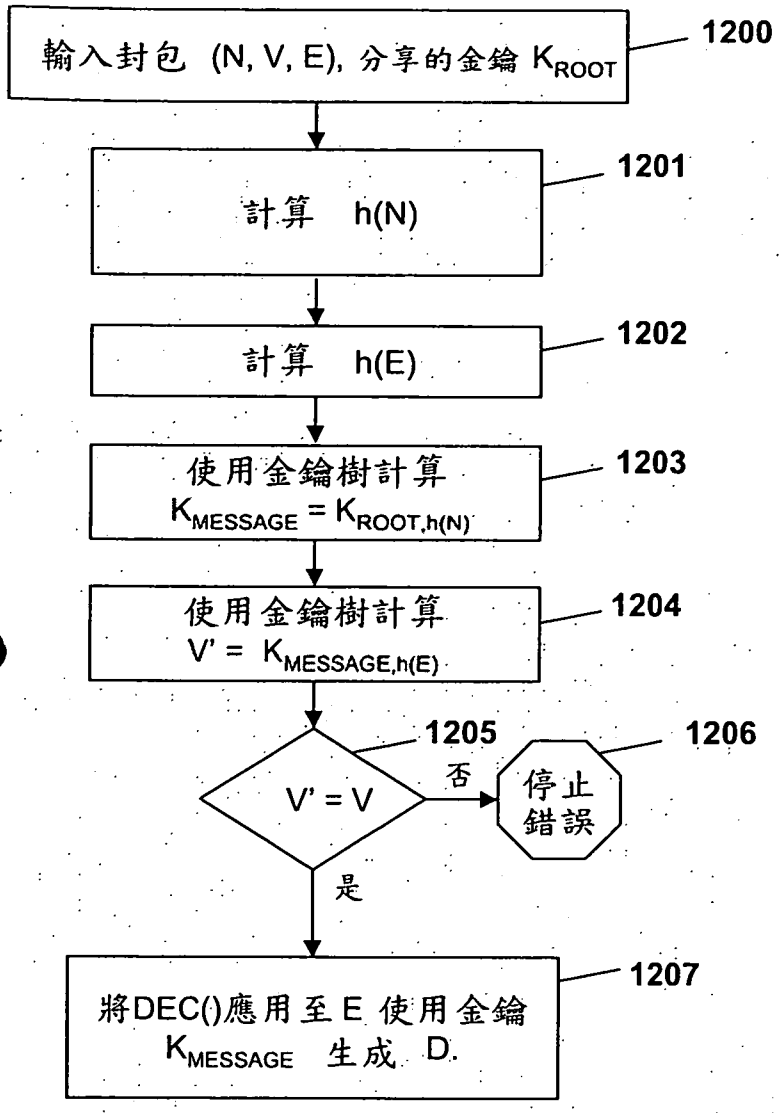
第9圖


裝置 A
1000
1002
OS, 應用程式, etc.
1003
明文封包暫存器
1004
區塊加密/解密處理
AES, 雜湊, 金鑰樹處理
具有根金鑰表的金鑰庫
1001
1005
密文封包暫存器
網路 I/F
1006
1020
網路
1030
裝置 B
1040
裝置 C, D, E...

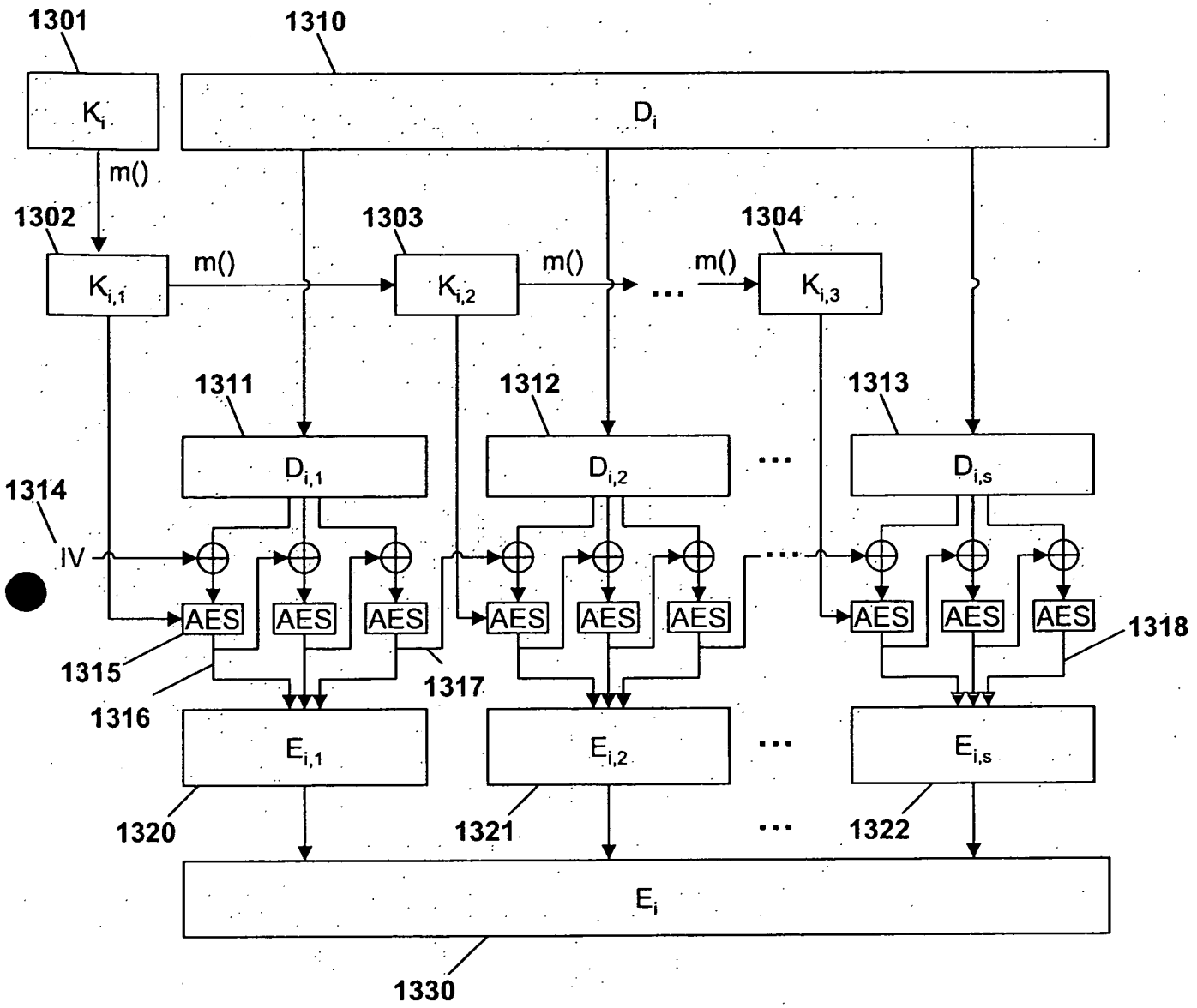
第10圖


輸入封包 D, 分享的金鑰 K_{ROOT}
1100
使用D及/或封包識別符以產生訊息識別符N
1101
計算 h(N)
1102
使用金鑰樹計算 $K_{MESSAGE} = K_{ROOT,h(N)}$
1103
將ENC()應用至D 與 $K_{MESSAGE}$ 生成加密的資料 E.
1104
計算 h(E)
1105
使用金鑰樹計算確認符 $V = K_{MESSAGE, h(E)}$
1106
輸出封包： N, V, E
1107

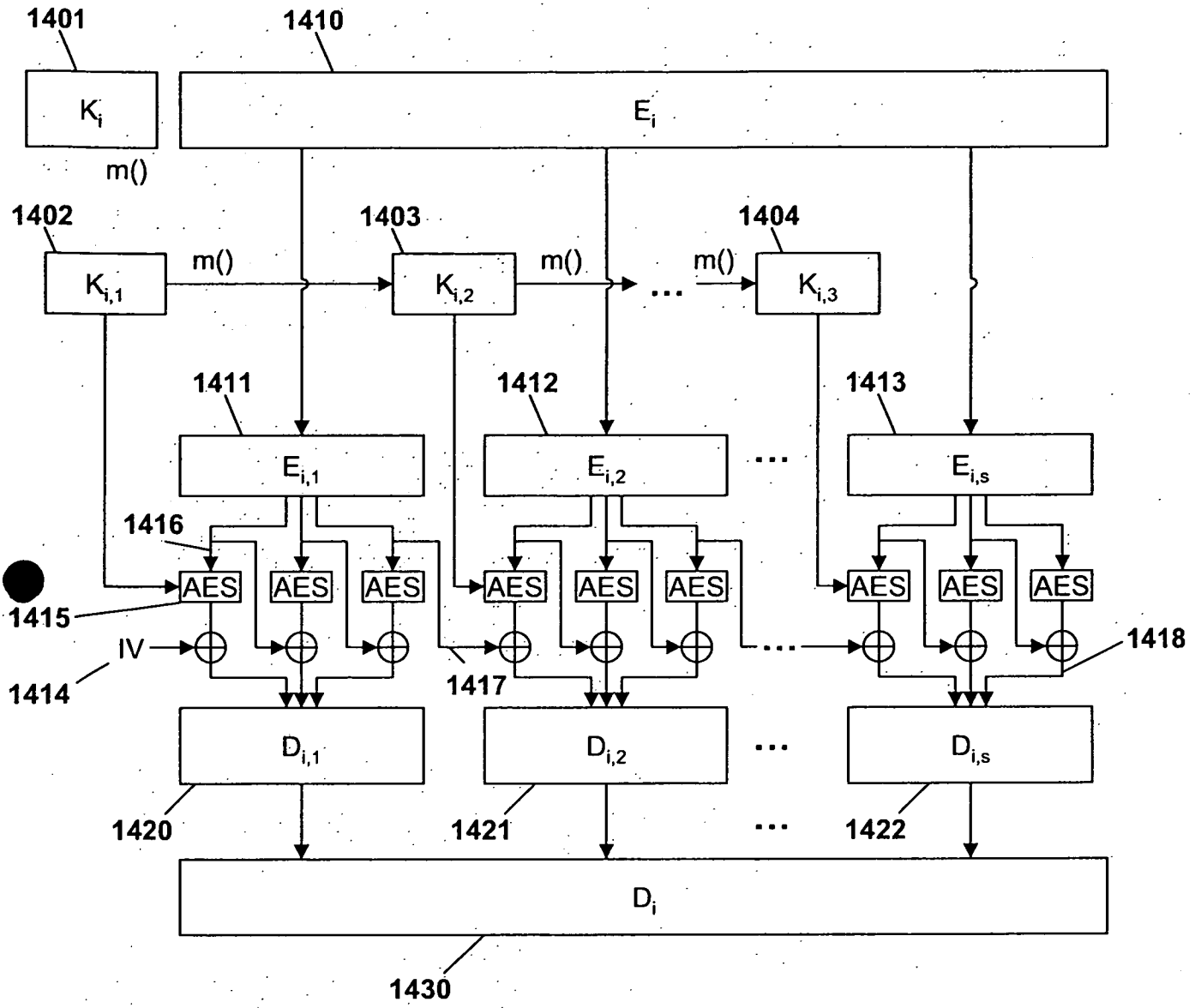
第11圖


輸入封包 (N, V, E), 分享的金鑰 K_{ROOT}
1200
計算 h(N)
1201
計算 h(E)
1202
使用金鑰樹計算
$K_{MESSAGE} = K_{ROOT,h(N)}$
1203
使用金鑰樹計算
$V' = K_{MESSAGE,h(E)}$
1204
V' = V
1205
否
停止
錯誤
1206
是
將DEC()應用至 E 使用金鑰
$K_{MESSAGE}$ 生成 D.
1207

第12圖


1301
1310
K_i
D_i
m()
1302
1303
1304
$K_{i,1}$
$K_{i,2}$
$K_{i,3}$
1311
1312
1313
$D_{i,1}$
$D_{i,2}$
$D_{i,s}$
1314
IV
AES
1315
1316
1317
1318
$E_{i,1}$
$E_{i,2}$
$E_{i,s}$
1320
1321
1322
E_i
1330

第13圖


1401
1410
K_i
E_i
m()
1402
1403
1404
$K_{i,1}$
$K_{i,2}$
$K_{i,3}$
1411
1412
1413
$E_{i,1}$
$E_{i,2}$
$E_{i,s}$
1416
AES
1415
IV
1414
1417
1418
$D_{i,1}$
$D_{i,2}$
$D_{i,s}$
1420
1421
1422
D_i
1430

第14圖