(19)中華民國智慧財產局
(12)發明說明書公開本 (11)公開編號：TW 201426540 A
(43)公開日： 中華民國 103 (2014) 年 07 月 01 日

(21)申請案號：103105439 (22)申請日： 中華民國 100 (2011) 年 05 月 24 日
(51)Int. Cl.： *G06F9/30 (2006.01)* *G06F21/71 (2013.01)*
(30)優先權：2010/05/25 美國 61/348,127
2011/04/21 美國 13/091,487
2011/04/21 美國 13/091,547
2011/04/21 美國 13/091,641
2011/04/21 美國 13/091,698
2011/04/21 美國 13/091,785
2011/04/21 美國 13/091,828
(71)申請人：威盛電子股份有限公司 (中華民國) VIA TECHNOLOGIES, INC. (TW)
新北市新店區中正路 533 號 8 樓
(72)發明人：亨利 G 葛蘭 HENRY, G. GLENN (US)；派克斯 泰瑞 PARKS, TERRY (US)；
比恩 布蘭特 BEAN, BRENT (US)；克理斯賓 湯姆士 CRISPIN, THOMAS A. (US)
(74)代理人：洪澄文；顏錦順
申請實體審查：有 申請專利範圍項數：23 項 圖式數：32 共 93 頁

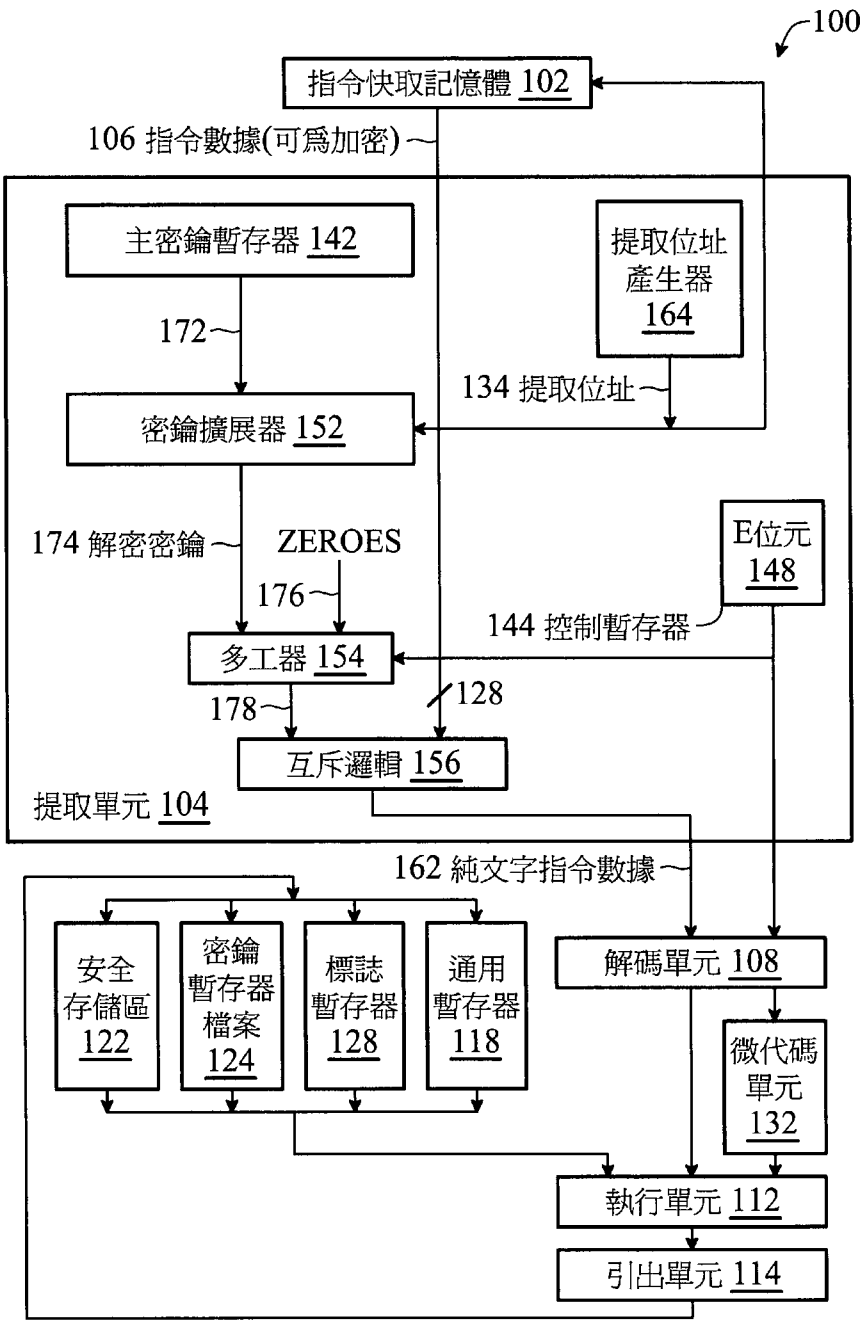
(54)名稱

解密密鑰產生裝置與方法

APPARATUS AND METHOD FOR GENERATING A DECRYPTION KEY

(57)摘要

微處理器之提取單元係使用第一解密密鑰數據提取並且解密一分支與切換密鑰指令，並在分支方向不被採用時以第一解密密鑰數據提取並且解密該分支與切換密鑰指令之後的接續指令，並在分支方向被採用時採用第二解密密鑰數據提取並且解密該分支與切換密鑰指令的一目標指令；其中包括指向解密密鑰數據、或參考目標位址至解密密鑰數據之映射等技術。加密程式係在加密一程式前將傳統之程式內分塊的分支指令以分支與切換密鑰指令取代。加密程序中所使用之資訊更將該程式劃分為一序列之複數塊，各塊為一序列之複數指令、且對應不同的加密密鑰數據。



第 1 圖

100：微處理器
102：指令快取記憶體
104：提取單元
106：指令數據(可為加密)
108：解碼單元
112：執行單元
114：引出單元
118：通用暫存器
122：安全存儲區
124：密鑰暫存器檔案
128：標誌暫存器
132：微代碼單元
134：提取位址
142：主密鑰暫存器
144：控制暫存器
148：E 位元
152：密鑰擴展器
154：多工器
156：互斥邏輯
162：純文字指令數據
164：提取位址產生器
172：兩組密鑰
174：解密密鑰
176：多位元的二進位零值
178：多工器 154 的輸出
ZEROES：多位元的二進位零值

(19)中華民國智慧財產局
(12)發明說明書公開本 (11)公開編號：TW 201426540 A
(43)公開日： 中華民國 103 (2014) 年 07 月 01 日

(21)申請案號：103105439 (22)申請日： 中華民國 100 (2011) 年 05 月 24 日
(51)Int. Cl.： *G06F9/30 (2006.01)* *G06F21/71 (2013.01)*
(30)優先權：2010/05/25 美國 61/348,127
2011/04/21 美國 13/091,487
2011/04/21 美國 13/091,547
2011/04/21 美國 13/091,641
2011/04/21 美國 13/091,698
2011/04/21 美國 13/091,785
2011/04/21 美國 13/091,828
(71)申請人：威盛電子股份有限公司 (中華民國) VIA TECHNOLOGIES, INC. (TW)
新北市新店區中正路 533 號 8 樓
(72)發明人：亨利 G 葛蘭 HENRY, G. GLENN (US)；派克斯 泰瑞 PARKS, TERRY (US)；
比恩 布蘭特 BEAN, BRENT (US)；克理斯賓 湯姆士 CRISPIN, THOMAS A. (US)
(74)代理人：洪澄文；顏錦順
申請實體審查：有 申請專利範圍項數：23 項 圖式數：32 共 93 頁

(54)名稱

解密密鑰產生裝置與方法

APPARATUS AND METHOD FOR GENERATING A DECRYPTION KEY

(57)摘要

微處理器之提取單元係使用第一解密密鑰數據提取並且解密一分支與切換密鑰指令，並在分支方向不被採用時以第一解密密鑰數據提取並且解密該分支與切換密鑰指令之後的接續指令，並在分支方向被採用時採用第二解密密鑰數據提取並且解密該分支與切換密鑰指令的一目標指令；其中包括指向解密密鑰數據、或參考目標位址至解密密鑰數據之映射等技術。加密程式係在加密一程式前將傳統之程式內分塊的分支指令以分支與切換密鑰指令取代。加密程序中所使用之資訊更將該程式劃分為一序列之複數塊，各塊為一序列之複數指令、且對應不同的加密密鑰數據。

201426540

發明摘要

※ 申請案號：102105439（TW201426540）
※ 申請日：102.2.7
※IPC 分類：G06F 9/30 (2006.01)
G06F 21/71 (2013.01)

【發明名稱】 解密密鑰產生裝置與方法

APPARATUS AND METHOD FOR GENERATING A DECRYPTION KEY

【中文】

微處理器之提取單元係使用第一解密密鑰數據提取並且解密一分支與切換密鑰指令，並在分支方向不被採用時以第一解密密鑰數據提取並且解密該分支與切換密鑰指令之後的接續指令，並在分支方向被採用時採用第二解密密鑰數據提取並且解密該分支與切換密鑰指令的一目標指令；其中包括指向解密密鑰數據、或參考目標位址至解密密鑰數據之映射等技術。加密程式係在加密一程式前將傳統之程式內分塊的分支指令以分支與切換密鑰指令取代。加密程序中所使用之資訊更將該程式劃分爲一序列之複數塊，各塊爲一序列之複數指令、且對應不同的加密密鑰數據。

【英文】

A microprocessor includes a fetch unit that fetches and decrypts an (atomic) branch and switch key instruction using first decryption key data. If the branch direction is not taken, the fetch unit fetches and decrypts the next sequential instruction after the branch and switch key instruction using the first decryption key data. If the direction is taken, the fetch unit

fetches and decrypts a target instruction of the branch and switch key instruction using second decryption key data that is different from the first decryption key data. The instruction points to the decryption key data; alternatively, the microprocessor consults a mapping of target address ranges to decryption key data. An encryption program replaces conventional inter-program-chunk branch instructions with branch and switch key instructions before encrypting the program using information that divides the program into a sequence of chunks each chunk being a sequence of instructions and having distinct associated encryption key data.

【代表圖】

【本案指定代表圖】：第 1 圖 。

【本代表圖之符號簡單說明】：

100~微處理器；　　102~指令快取記憶體；

104~提取單元；　　106~指令數據(可爲加密)；

108~解碼單元；　　112~執行單元；

114~引出單元；　　118~通用暫存器；

122~安全存儲區；　　124~密鑰暫存器檔案；

128~標誌暫存器；　　132~微代碼單元；

134~提取位址；　　142~主密鑰暫存器；

144~控制暫存器；　　148~E位元；

152~密鑰擴展器；　　154~多工器；

156~互斥邏輯；　　162~純文字指令數據；

164~提取位址產生器；　　172~兩組密鑰；

174~解密密鑰；　　176~多位元的二進位零值；

178~多工器154的輸出；

ZEROES~多位元的二進位零值。

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

無 。

發明專利說明書

（本說明書格式、順序，請勿任意更動）

【發明名稱】 解密密鑰產生裝置與方法

APPARATUS AND METHOD FOR GENERATING A DECRYPTION KEY

【技術領域】

【0001】 本發明係有關於微處理器(microprocessor)領域，特別用於增加微處理器所執行的程式之安全性。

【先前技術】

【0002】 很多軟體程式在面臨破壞電腦系統安全的攻擊時，通常是脆弱不堪的。例如，駭客可藉由攻擊一運行中程式的緩衝溢位區漏洞(buffer overflow vulnerability)植入不當程式碼、並轉移主控權給該不當程式碼。如此一來，所植入的程式碼將主導被攻擊的程式。一種防範軟體程式遭攻擊的方案爲指令集隨機化(instruction set randomization)。概略解釋之，指令集隨機化技術會先將程式加密(encrypt)爲某些形式，再於處理器將該程式自記憶體提取後，於該處理器內解密(decrypt)該程式。如此一來，駭客便不易植入惡意指令，因爲所植入的指令必須被適當地加密(例如，使用與所攻擊程式相同的加密密鑰或演算法)方會被正確地執行。例如，參閱文件「Counter Code-Injection Attacks with Instruction-Set Randomization, by Gaurav S. Kc, Angelos D. Keromytis, and Vassilis Prevelakis, CCS ' 03, October 27-30, 2003, Washington, DC, USA, ACM 1-58113-738-9/03/0010」，其中敘述Bochs-x86 Pentium模擬器

(emulator)之改良版本。相關技術的缺點已被廣泛討論。例如，參閱資料「Where' s the FEEB? The Effectiveness of Instruction Set Randomization, by Ana Nora Sovarel, David Evans, and Nathanael Paul, http://www.cs.virginia.edu/feeb」。

【發明內容】

【0003】 本發明一種實施方式揭露一種微處理器。該微處理器包括一提取單元，使用第一解密密鑰數據提取並且解密一分支與切換密鑰指令。該微處理器更包括微代碼。上述微代碼在該分支與切換密鑰指令的方向不被採用的狀況下，令該提取單元採用上述第一解密密鑰數據提取並且解密該分支與切換密鑰指令之後的接續指令。該微代碼更在該分支與切換密鑰指令方向被採用的狀況下，令該提取單元採用不同於上述第一解密密鑰數據的第二解密密鑰數據提取並且解密該分支與切換密鑰指令的一目標指令。

【0004】 本發明另外一種實施方式揭露一方法，以一微處理器處理一加密程式。該方法包括使用第一解密密鑰數據提取並且解密一分支與切換密鑰指令。此方法更包括，在該分支與切換密鑰指令的方向不被採取的狀況下，以上述第一解密密鑰數據提取並且解密該分支與切換密鑰指令之後的接續指令。該方法更包括，在該分支與切換密鑰指令的方向被採取的狀況下，以不同於上述第一解密密鑰數據的第二解密密鑰數據提取並且解密該分支與切換密鑰指令的一目標指令。

【0005】 本發明另外一種實施方式亦揭露一方法，用於加密一程式，以供用於解密與執行加密程式的一微處理器日後執

行。該方法包括接收一非加密程式的一目的檔，其中包括傳統分支指令，所指示的目標位址可於該微處理器執行該程式前判定。該方法更包括分析該程式以獲得塊資訊。上述塊資訊將該程式劃分成一序列多個塊。各塊包括一序列多個指令。上述塊資訊更包括各塊相關的加密密鑰數據。各塊對應的加密密鑰數據不相同。該方法更包括將上述傳統分支指令中目標位址與自身坐落不同塊者各自以一分支與切換密鑰指令取代。該方法更包括基於上述塊資訊加密該程式。

【0006】 本發明另外一種實施方式亦揭露一方法，用於加密一程式，以供用於解密與執行加密程式的一微處理器日後執行。該方法包括接收一非加密程式的一目的檔，其中包括傳統分支指令，所指示的目標位址僅能在該微處理器執行該程式時判定。該方法更包括分析該程式以獲得塊資訊。上述塊資訊將該程式劃分成一序列多個塊。各塊包括一序列多個指令。上述塊資訊更包括各塊相關的加密密鑰數據。各塊對應的加密密鑰數據不相同。該方法更包括將上述傳統分支指令各自以一分支與切換密鑰指令取代。該方法更包括基於上述塊資訊，加密該程式。

【圖式簡單說明】

【0007】

第1圖爲一方塊圖，圖解根據本發明技術實現的一微處理器；

第2圖爲一方塊圖，用以詳細說明圖解第1圖的提取單元；

第3圖爲一流程圖，根據本發明技術，圖解第2圖提取單元

之操作；

第4圖爲一方塊圖，根據本發明技術，圖解第1圖標誌暫存器的欄位；

第5圖爲一方塊圖，根據本發明技術，圖解一密鑰載入指令的格式；

第6圖爲一方塊圖，根據本發明技術，圖解一密鑰切換指令的格式；

第7圖爲一流程圖，根據本發明技術，圖解第1圖微處理器的操作，其中執行第6圖之密鑰切換指令；

第8圖爲一方塊圖，根據本發明技術，圖解一加密程式的記憶體用量，該加密程式包括多個第6圖所揭露的密鑰切換指令；

第9圖爲一方塊圖，根據本發明技術，圖解一分支與切換密鑰指令的格式；

第10圖爲一流程圖，根據本發明技術，圖解第1圖微處理器的操作，其中執行第9圖之分支與切換密鑰指令；

第11圖爲一流程圖，根據本發明技術，圖解一後處理器的操作，由軟件工具實現，可用於後部處理一程式、且加密之，以由第1圖微處理器執行；

第12圖爲一方塊圖，圖解本發明另外一種實施方式的分支與切換密鑰指令的格式；

第13圖爲一方塊圖，根據本發明技術，圖解塊位址範圍表；

第14圖爲一流程圖，根據本發明技術，圖解第1圖微處理器的操作，其中執行第12圖之分支與切換密鑰指令；

第15圖爲一方塊圖，圖解本發明另外一種實施方式的分支與切換密鑰指令的格式；

第16圖爲一方塊圖，根據本發明技術，圖解塊位址範圍表；

第17圖爲一流程圖，根據本發明技術，圖解第1圖微處理器的操作，其中執行第15圖之分支與切換密鑰指令；

第18圖爲一流程圖，圖解本發明技術另外一種實施方式，其中敘述一後處理器的操作，用於後部處理一程式、且加密之，由第1圖微處理器執行；

第19圖爲一流程圖，根據本發明技術，圖解第1圖微處理器的操作，用於應付一任務切換，切換於一加密程式以及一純文字程式之間；

第20圖圖解一流程圖，根據本發明技術，圖解第1圖微處理器所執行的系統軟體之操作；

第21圖圖解一方塊圖，根據本發明另外一種實施方式，圖解第1圖標誌暫存器的欄位；

第22圖爲一流程圖，根據本發明技術，圖解採用第21圖之標誌暫存器的第1圖微處理器之操作，用於應付一任務切換，切換於多個加密程式之間；

第23圖爲一流程圖，根據本發明技術，圖解採用第21圖之標誌暫存器的第1圖微處理器之操作，用於應付一任務切換，切換於多個加密程式之間；

第24圖爲一方塊圖，根據本發明另外一種實施方式，圖解第1圖密鑰暫存器檔案中的單一個暫存器；

第25圖爲一流程圖，根據本發明另外一種實施方式，圖解

採用第21圖標誌暫存器以及第24圖密鑰暫存器檔案的第1圖微處理器之操作，以應付一任務切換，切換於多個加密程式之間；

第26圖爲一流程圖，根據本發明另外一種實施方式，圖解採用第21圖標誌暫存器以及第24圖密鑰暫存器檔案的第1圖微處理器之操作，以應付一任務切換，切換於多個加密程式之間；

第27圖爲一方塊圖，圖解第1圖微處理器100部分內容的其他實施方式；

第28圖爲一方塊圖，根據本發明技術，詳細圖解第27圖的分支目標位址快取記憶體(BTAC)；

第29圖爲一方塊圖，根據本發明技術，詳細圖解第28圖之BTAC各單元之內容；

第30圖爲一流程圖，根據本發明技術，圖解第27圖微處理器採用第28圖BTAC的操作；

第31圖爲一流程圖，根據本發明技術，圖解第27圖微處理器採用第28圖BTAC的操作；以及

第32圖爲一流程圖，根據本發明技術，圖解第27圖微處理器對一分支與切換密鑰指令的操作。

【實施方式】

【0008】 參閱第1圖，一方塊圖圖解根據本發明技術所實現的一微處理器100。微處理器100包括一管線(pipeline)，其中包括一指令快取記憶體(instruction cache)102、一提取單元(fetch unit)104、一解碼單元(decode unit)108、一執行單元(execution

unit)112、以及一引出單元(retire unit)114。微處理器100更包括一微代碼單元(microcode unit)132，用以提供微代碼指令(microcode instructions)給該執行單元112。微處理器100更包括通用暫存器(general purpose registers)118以及標誌暫存器(EFLAGS register)128，以提供指令運算元(instruction operands)給執行單元112。而且，透過引出單元114，將指令執行結果更新於通用暫存器118以及標誌暫存器128。在一種實施方式中，標誌暫存器128是由傳統x86標誌暫存器修改實現，詳細實施方式將於後續篇幅說明。

【0009】 提取單元104自指令快取記憶體102提取指令數據(instruction data)106。提取單元104操作於兩種模式：一爲解密模式(decryption mode)，另一爲純文字模式(plain text mode)。提取單元104內一控制暫存器(control register)144的一E位元(E bit)148決定該提取單元104是操作於解密模式(設定E位元)、或操作於純文字模式(清空E位元)。純文字模式下，提取單元104視自該指令快取記憶體102所提取出的指令數據106爲未加密、或純文字指令數據，因此，不對指令數據106作解密。然而，在解密模式下，提取單元104視自該指令快取記憶體102所提取出的指令數據106爲加密指令數據，因此，需使用該提取單元104的一主密鑰暫存器(master key register)142所儲存的解密密鑰(decryption keys)將之解密爲純文字指令數據，詳細技術內容將參考第2圖以及第3圖進行討論。

【0010】 提取單元104亦包括一提取位址產生器(fetch address generator)164，用以產生一提取位址(fetch

address)134，以自該指令快取記憶體102提取指令數據106。提取位址134更供應給提取單元104的一密鑰擴展器(key expander)152。密鑰擴展器152自主密鑰暫存器142中選取兩組密鑰172，並對其實施運算以產生一解密密鑰174，作爲多工器154的第一輸入。多工器154的第二輸入爲多位元的二進位零值(binary zeroes)176。E位元148控制多工器154。若E位元148被設定，多工器154選擇輸出該解密密鑰174。若E位元148被清除，多工器154選擇輸出多位元的二進位零值176。多工器154的輸出178將供應給互斥邏輯156作爲其第一輸入。互斥邏輯156負責對提取的指令數據106以及多工器輸出178施行布林互斥運算(Boolean exclusive-OR，XOR)，以產生純文字指令數據162。加密的指令數據106乃預先以互斥邏輯將其原本的純文字指令數據以一加密密鑰進行加密，其中該加密密鑰之數值與該解密密鑰174相同。提取單元104的詳細實施方式將配合第2圖以及第3圖內容於稍後敘述。

【0011】 純文字指令數據162將供應給解碼單元108。解碼單元108負責將純文字指令數據162之串流解碼、並分割爲多個X86指令，交由執行單元112執行。在一種實施方式中，解碼單元108包括緩衝器(buffers)或佇列(queues)，以在解碼之前或期間，緩衝存儲的純文字指令數據162之串流。在一種實施方式中，解碼單元108包括一指令轉譯器(instruction translator)，用以將X86指令轉譯爲微指令microinstructions或micro-ops，交由執行單元112執行。解碼單元108輸出指令時，更會針對各指令輸出一位元值，該位元值乃伴隨該指令沿所述管線結構一路行

進而至，用以指示該指令是否爲加密指令。該位元值將控制該執行單元112以及該引出單元114，使之根據該指令自該指令快取記憶體102取出時是加密指令或純文字指令而進行決策並且採取動作。在一種實施方式中，純文字指令不被允許執行專供指令解密模式設計的特定操作。

【0012】 在一種實施方式中，微處理器100爲一x86架構處理器，然而，微處理器100也可以其他架構之處理器實現。若一處理器可正確執行設計給x86處理器執行的大多數應用程式，則視之爲x86架構的處理器。若應用程式執行後可獲得預期結果，則可判斷該應用程式是被正確執行。特別是，微處理器100是執行x86指令集的指令，且具有x86用戶可用暫存器組(x86 user-visible register set)。

【0013】 在一種實施方式中，微處理器100乃設計成供應一複合安全架構(comprehensive security architecture)—稱爲安全執行模式(secure execution mode，簡稱SEM)—以於其中執行程式。根據一種實施方式，SEM程式的執行可由數種處理器事件(processor events)引發，且不受一般(非SEM)操作封鎖。以下舉例說明限定於SEM下執行的程式所實現的功能，其中包括關鍵安全任務(critical security tasks)如：憑證核對以及資料加密、系統軟件活動監控、系統軟件完整性驗證、資源使用追蹤、新軟件的安裝控制…等。關於SEM的實施方式請參考本公司於2008年10月31日申請的美國專利申請案，案號12/263,131，(美國專利公開號爲2009-0292893，於2009年11月26日公開)；該案的優先權主張溯及2008年5月24日的美國專利臨時申請案(案

號61/055,980)；本申請案相關技術部份可參照上述案件內容。在一種實施方式中，用於存儲SEM數據爲安全非揮發記憶體(未顯示在圖示)—如快取記憶體(flash memory)—可用於存儲解密密鑰，並藉由一隔離串行匯流排(private serial bus)耦接微處理器100，且其中所有資料乃AES加密(AES-encrypted)且經過簽署驗正(signature-verified)的。在一種實施方式中，微處理器100包括少量的單一次寫入性非揮發記憶體(non-volatile write-once memory，未顯示於圖示)，用於存儲解密密鑰；其中一種實施方式可參考美國專利案7,663,957所揭露的一熔絲型非揮發存儲器；可參照上述案件內容應用於本案發明。本案所揭露的指令解密特徵的其中一項優點爲：擴展安全執行模式(SEM)的應用範圍，使安全性程式(secure program)得以存儲在微處理器100外的記憶體，無須限定完整存儲於微處理器100內部。因此，安全性程式可利用記憶體階層架構所提供的完整空間以及功能。在一種實施方式中，部分或全部的結構性異常/中斷(architectural exceptions/interrupts，例如，頁面錯誤page faults、除錯中斷點debug breakpoints)…等，在SEM模式下是除能(disable)的。在一種實施方式中，部分或全部的結構性異常/中斷在解密模式(即E位元148爲設定)下是除能(disable)的。

【0014】 微處理器100更包括一密鑰暫存器檔案(key register file)124。密鑰暫存器檔案124包括複數個暫存器，其中儲存的密鑰可藉由密鑰切換指令(switch key instruction，後續討論之)載入提取單元104的主密鑰暫存器142，以解密所提取的加密指令數據106。

【0015】 微處理器100更包括一安全存儲區(secure memory area，簡寫爲SMA)122，用於存儲解密密鑰，該解密密鑰待經第5圖所示之密鑰載入指令(load key instruction)500進而載入密鑰暫存器檔案124。在一種實施方式中，安全存儲區122限定以SEM程式存取。也就是說，安全存儲區122不可藉一般執行模式(非SEM)下所執行的程式存取。此外，安全存儲區122也不可藉處理器匯流排存取，且不屬於微處理器100之快取記憶體階層的一部份。因此，舉例說明之，快取清空操作(cache flush operation)不會導致安全存儲區122的內容寫入記憶體。關於安全存儲區122的讀寫，微處理器100指令集架構中設計有特定指令。一種實施方式是在安全存儲區122中設計一隔離式隨機存取記憶體(private RAM)，相關技術內容可參考2008年2月20日申請的美國專利申請案12/034,503(該案於2008年10月16日公開，公開號爲2008/0256336)；可參照上述案件內容應用於本案發明。

【0016】 起先，作業系統或其他特權程序(privileged program)下載密鑰的初始化設定於該安全存儲區122、密鑰暫存器檔案124、以及主密鑰暫存器142。微處理器100起先會以該密鑰的初始化設定以解密一加密程式。此外，加密程式本身可接續寫入新的密鑰至安全存儲區122、並自安全存儲區122將密鑰載入密鑰暫存器檔案124(藉由密鑰載入指令)、且自密鑰暫存器檔案124將密鑰載入主密鑰暫存器142(藉由密鑰切換指令)。所述操作之優勢在於：所揭露的密鑰切換指令使得加密程式在執行當下得以切換解密密鑰組(on-the-fly switching)，

以下將詳述之。新的密鑰可由加密程式指令自身的即時數據組成。在一種實施方式中，程式檔案標頭的一欄位會指示程式指令是否爲加密型式。

【0017】 第1圖所描述的技術有多項優點。第一，自加密指令數據106所解密出來的純文字指令數據無法由微處理器100外部獲得。

【0018】 第二，提取單元104提取加密指令數據所需的時間與提取純文字指令數據所需的時間相同。此特色關係著安全與否。反之，若有時間差存在，駭客可藉此破解加密技術。

【0019】 第三，相較於傳統設計，本案所揭露之指令解密技術不會額外增加提取單元104所耗的時脈數量。如以下討論，密鑰擴展器152增加解密密鑰之有效長度，該解密密鑰用於解密一加密程式，且此方式不會使提取加密程式數據所需的時間長於提取純文字程式數據所需的時間。特別是，因爲密鑰擴展器152之運作限時於以提取位址134查表該指令快取記憶體102獲得指令數據106之內完成，密鑰擴展器152並不會增加一般的提取程序的時間。此外，因爲多工器154以及密鑰擴展器152一併限時於以提取位址134查表該指令快取記憶體102獲得指令數據106之內完成，故不會增加一般的提取程序的時間。互斥邏輯156是唯一添加於一般提取路徑的邏輯運算，所幸互斥操作156的傳播延遲相當小，不會增加工作週期。因此，本案所揭露的指令解密技術不會增加提取單元104時脈數量負擔。此外，相較於一般技術所應用於解密指令數據106的複雜解密機制，例如S盒(S-boxes)，一般技術會增加提取以及解碼

指令數據106時所需的工作週期且/或所消耗的時脈數量。

【0020】 接著，參考第2圖，一方塊圖詳細圖解第1圖之提取單元104。特別是，第1圖之密鑰擴展器152也詳細圖列其中。先前已討論採用互斥邏輯解密上述加密指令數據106的優點。然而，快且小的互斥邏輯有其缺點：若加密/解密密鑰被重複使用，則互斥邏輯屬於一種脆弱加密方法(weak encryption method)。不過，若密鑰的有效長度等同所欲加密/解密之程式的長度，互斥邏輯加密會是一種強度極高的加密技術。微處理器100之特徵在於可增長解密密鑰的有效長度，以降低密鑰重複使用的需求。第一，主密鑰暫存器檔案142所儲存的數值爲中大型尺寸：在一種實施方式中，其尺寸等同自指令快取記憶體102所取出的指令數據106之提取量、或區塊尺寸，爲128位元(16位元組)。第二，加密擴展器152用於增長解密密鑰的有效長度，例如，增至一實施方式所揭露的2048位元組，將於後續篇幅詳述。第三，加密程式可藉由密鑰切換指令(或其變形)在操作中改變主密鑰暫存器142內的數值，之後段落將詳述之。

【0021】 在第2圖所示實施方式中，142使用了五個主密鑰暫存器，編號0-4。然而，在其他實施方式中，也可以較少或較多量的主密鑰暫存器142數量增長解密密鑰長度。例如，一種實施方式採用12個主密鑰暫存器142。密鑰擴充器152包括一第一多工器A 212以及一第二多工器 B 214，用以接收主密鑰暫存器142所供應的密鑰。提取位址134的部分內容用於控制多工器212/214。在第2圖所示實施方式中，多工器B 214爲三轉一多工器，而多工器A 212爲四轉一多工器。表格1顯示多工器

212/214如何根據各自的選擇輸入選取該等主密鑰暫存器142(以上述編號識別)。表格2顯示上述選擇輸入的產生方式，以及基於提取位址134的位元[10:8]所呈的主密鑰暫存器142組合。

多工器B的選擇信號	選取的主密鑰暫存器之編號	多工器A的選擇信號	選取的主密鑰暫存器之編號
00	0	00	1
01	1	01	2
10	2	10	3
		11	4

表格1

提取位址之位元[10:8]	多工器B-多工器A的選取組合	多工器B的選取信號	多工器A的選取信號
000	0-1	00	00
001	0-2	00	01
010	0-3	00	10
011	0-4	00	11
100	1-2	01	01
101	1-3	01	10
110	1-4	01	11
111	2-3	10	10

表格2

【0022】 多工器B 214的輸出236是供應給加法/減法器218。多工器A 212的輸出234是供應給一旋轉器(rotator)216。旋轉器216接收提取位址134的位元[7:4]，據以旋轉多工器輸出234，決定旋轉的位元組數量。在一種實施方式中，提取位址134的位元[7:4]在供應給旋轉器216控制旋轉的位元組數量前增量，以表格3顯示之。旋轉器216的輸出238是供應給加法/減法器218。加法器/減法器218接收提取位址134的位元[7]。若該位元[7]爲清空，加法/減法器218將旋轉器216的輸出238自多工器B 214之輸出236減去。若該位元[7]爲設定，加法/減法器218將旋轉器216的輸出238加上多工器B 214的輸出236。加法/減法器218的輸出即第1圖所示之解密密鑰174，將供應給多工器154。以下以第3圖之流程圖詳述相關技術。

【0023】 接著，參閱第3圖，一流程圖基於本發明技術圖解第2圖提取單元104的操作。流程始於方塊302。

【0024】 在方塊302，提取單元104以提取位址134讀取指令快取記憶體102，以開始提取一16位元組之區塊的指令數據106。指令數據106可爲加密狀態或爲純文字狀態，視指令數據106是爲一加密程式或一純文字程式的一部分而定，由E位元148標示。流程接著進入方塊304。

【0025】 參考方塊304，根據提取位址134較高的數個位元，多工器A 212以及多工器B 214分別自主密鑰暫存器142所供應的密鑰172中選取出一第一密鑰234以及一第二密鑰236。在一種實施方式中，提取位址134所供應的該些位元施加於多工器212/214，以產生特定的密鑰對(234/236 key pair)組合。在

第2圖所示之實施方式中，所供應的主密鑰暫存器142數量爲5，因此，存在10組可能的密鑰對。爲了簡化硬體設計，僅使用了其中8組；此設計將供應2048位元組的有效密鑰，將於後續段落詳細討論之。然而，其他實施方式也可能使用其他數量的密鑰暫存器142。以供應12個主密鑰暫存器142的實施方式爲例，主密鑰暫存器142的可能組合有66組，若採用其中64組，所產生的有效密鑰將爲16384位元組。整體而言，假設上述複數個密鑰數值總量爲K(例如：5，且採用全部組合)，該解密密鑰、以及上述複數個密鑰數值各自的長度爲W位元組(例如：16位元組)，則產生的有效密鑰將爲W^2 * (K!/(2*(K-2)!)) 位元組。流程接著進入方塊306。

【0026】 在方塊306，基於提取位址134的位元[7:4]，旋轉器216使第一密鑰234旋轉相應數量的位元組。例如，若提取位址134的位元[7:4]爲數值9，旋轉器216將第一密鑰234朝右旋轉9個位元組。流程接著進入方塊308。

【0027】 在方塊308，加法/減法器218將旋轉後的第一密鑰238加至/減自該第二密鑰236，以產生第1圖之解密密鑰174。在一種實施方式中，若提取位址134的位元[7]爲1，則加法/減法器218將旋轉後的第一密鑰234加至該第二密鑰236；若提取位址134的位元[7]爲0，則加法/減法器218將旋轉後的第一密鑰234自該第二密鑰236減去。接著，流程進入方塊312。

【0028】 在決策方塊312，多工器154根據其控制信號判斷所提取的該區塊之指令數據106是來自一加密程式或一純文字程式，所述控制信號來自控制暫存器144所供應的位元E 148。

若指令數據106爲加密狀態，流程進入方塊314，反之，則流程進入方塊316。

【0029】 在方塊314，多工器154選擇輸出解密密鑰174，且互斥邏輯156令加密指令數據106以及解密密鑰174進行一布林互斥運算，以產生第1圖之純文字指令數據162。流程止於方塊314。

【0030】 在方塊316，多工器154選擇輸出16位元組的二進位零值176，且互斥邏輯156令指令數據106(爲純文字)以及該16位元組的二進位零值進行一布林互斥運算，以產生同樣的純文字指令數據162。流程止於此方塊316。

【0031】 參考第2圖以及第3圖所揭露內容，解密密鑰174供應給所提取的該區塊指令數據106進行互斥運算，且該解密密鑰174是所選取的主密鑰對234/236以及提取位址134之函數。相比於傳統解密程序—使解密密鑰爲先前密鑰值的一函數，其中持續修正密鑰以供應新的在下一次工作區間使用—本案所揭露之解密技術完全不同。以主密鑰對234/236以及提取位址134爲函式獲得解密密鑰174的方式有至少以下兩種優點。第一，如以上所討論，加密指令數據以及純文字指令數據106之提取耗時相當，不會增加微處理器100所需的工作時脈。第二，遇到程式中的分支指令(branch instruction)，提取指令數據106所需的時間不會增加。在一種實施方式中，一分支預測器(branch predictor)接收提取位址134，並預測該提取位址134所指之該區塊的指令數據106是否存在一分支指令，並預測其方向以及目標位址。以第2圖所示實施方式爲例，產出的解密密

鑰174是主密鑰對234/236以及提取位址134的一函式，將在目標位址所指之該區塊指令數據106送抵該互斥邏輯156的同一時間產出預測之目標位址的適當解密密鑰174。與傳統解密密鑰運算手法針對目標位址計算解密密鑰所必須的多個「倒帶(rewind)」步驟相較，本案所揭露技術在處理加密指令數據時不會產生額外的延遲。

【0032】 另外，如第2圖以及第3圖所示，密鑰擴展器152之旋轉器216以及加法/減法器218之聯合設計，使得解密密鑰長度有效擴展，超越主密鑰之長度。例如，主密鑰共貢獻32位元組(2*16位元組)；更甚者，以駭客企圖判斷解密密鑰174爲何的角度而言，旋轉器216以及加法/減法器218有效地將位於主密鑰暫存器142的32位元組的主密鑰擴展爲256位元組的密鑰序列。更具體地說，有效擴展後的密鑰序列之位元組n爲：

$$k_{0_n} \pm k_{1_{n+x}}$$

k_{0_n}爲第一主密鑰234的位元組n，且$k_{1_{n+x}}$爲第二主密鑰236的位元組n+x。如上所述，密鑰擴展器152所產生的前八套16位元組解密密鑰174是由減法方式產生，且後八套是由加法方式產生。具體來說，選定的主密鑰對234/236各自所提供的位元組內容用於爲16個連續的16位元組區塊之指令數據各個位元組產生解密密鑰174位元組，詳情請見表格3。舉例說明之，表格3第1列的符號“15-00”表示第二主密鑰236的位元組0的內容會經8位元算數運算(an eight-bit arithmetic operation)自第一主密鑰234的位元組15減去，以獲得一位元組的有效解密密鑰

174，用以與一16位元組區塊之指令數據106中的位元組15進行互斥運算。

15-00	14-15	13-14	12-13	11-12	10-11	09-10	08-09	07-08	06-07	05-06	04-05	03-04	02-03	01-02	00-01
15-01	14-00	13-15	12-14	11-13	10-12	09-11	08-10	07-09	06-08	05-07	04-06	03-05	02-04	01-03	00-02
15-02	14-01	13-00	12-15	11-14	10-13	09-12	08-11	07-10	06-09	05-08	04-07	03-06	02-05	01-04	00-03
15-03	14-02	13-01	12-00	11-15	10-14	09-13	08-12	07-11	06-10	05-09	04-08	03-07	02-06	01-05	00-04
15-04	14-03	13-02	12-01	11-00	10-15	09-14	08-13	07-12	06-11	05-10	04-09	03-08	02-07	01-06	00-05
15-05	14-04	13-03	12-02	11-01	10-00	09-15	08-14	07-13	06-12	05-11	04-10	03-09	02-08	01-07	00-06
15-06	14-05	13-04	12-03	11-02	10-01	09-00	08-15	07-14	06-13	05-12	04-11	03-10	02-09	01-08	00-07
15-07	14-06	13-05	12-04	11-03	10-02	09-01	08-00	07-15	06-14	05-13	04-12	03-11	02-10	01-09	00-08
15+08	14+07	13+06	12+05	11+04	10+03	09+02	08+01	07+00	06+15	05+14	04+13	03+12	02+11	01+10	00+09
15+09	14+08	13+07	12+06	11+05	10+04	09+03	08+02	07+01	06+00	05+15	04+14	03+13	02+12	01+11	00+10
15+10	14+09	13+08	12+07	11+06	10+05	09+04	08+03	07+02	06+01	05+00	04+15	03+14	02+13	01+12	00+11
15+11	14+10	13+09	12+08	11+07	10+06	09+05	08+04	07+03	06+02	05+01	04+00	03+15	02+14	01+13	00+12
15+12	14+11	13+10	12+09	11+08	10+07	09+06	08+05	07+04	06+03	05+02	04+01	03+00	02+15	01+14	00+13
15+13	14+12	13+11	12+10	11+09	10+08	09+07	08+06	07+05	06+04	05+03	04+02	03+01	02+00	01+15	00+14
15+14	14+13	13+12	12+11	11+10	10+09	09+08	08+07	07+06	06+05	05+04	04+03	03+02	02+01	01+00	00+15
15+15	14+14	13+13	12+12	11+11	10+10	09+09	08+08	07+07	06+06	05+05	04+04	03+03	02+02	01+01	00+00

表格3

【0033】 給定適當的主密鑰數值後，密鑰擴展器152所產生的擴展密鑰統計來說可有效預防互斥加密常見的攻擊，包括令文件之加密區塊以密鑰長度位移、並對加密區塊一併施行互斥運算，以下更詳細討論之。密鑰擴展器152對選定主密鑰對234/236之影響是：在所述實施方式中，程式中以完全相同的密鑰所加密的兩個指令數據106位元組之跨距可高達256位元組。在其他具有不同區塊尺寸的指令數據106、以及不同主密鑰長度的實施方式中，以同樣密鑰加密的兩個指令數據106位元組的最大跨距可有不同的量。

【0034】 用來選定主密鑰對234/236的主密鑰暫存器142以及密鑰擴展器152內的多工器212/214也會決定有效密鑰長度的擴展程度。如以上討論，第2圖所示實施方式供應有5個主密

鑰暫存器142，主密鑰暫存器142所供應的內容因此可以10種方式組合，而多工器212/214是用於自上述10種可能組合方式中選擇八種作用。表格3所示各密鑰對234/236所對應的256位元組有效密鑰長度搭配八種主密鑰對234/236組合後，所產生的有效密鑰長度爲2048位元組。也就是說，程式中以完全相同之密鑰加密的兩個指令數據106位元組之跨距可高達2048位元組。

【0035】 爲了更加說明密鑰擴展器152所帶來的優點，以下簡短敘述互斥加密程序所常見的的攻擊。若互斥加密運算所採用的密鑰長度短於所加密/解密之程式指令數據的長度，密鑰中的許多位元組必須被重複使用，且被重複使用的位元組數量視程式之長度而定。此弱點使互斥指令加密程序可被破解。第一，駭客嘗試判斷出重複密鑰之長度，以下展示的說明(1)至(3)令之爲n+1。第二，駭客假定指令數據內各個密鑰長度區塊(key-length block)是以同樣密鑰加密。以下列舉根據一傳統互斥加密運算加密得到的二密鑰長度區塊的數據：

(1) $b_{n_0} \wedge k_n$，…，$b_{1_0} \wedge k_1$，$b_{0_0} \wedge k_0$

(2) $b_{n_1} \wedge k_n$，…，$b_{1_1} \wedge k_1$，$b_{0_1} \wedge k_0$

其中，b_{n_0}爲第一密鑰長度區塊之數據的位元組n，將被加密；b_{n_1}爲第二密鑰長度區塊之數據的位元組n，將被加密；且k_n爲密鑰的位元組n。第三，駭客對所述兩區塊進行互斥運算，使其中密鑰成分彼此相銷，獨留以下內容：

(3) $b_{n_0} \wedge b_{n_1}$，…，$b_{1_0} \wedge b_{1_1}$，$b_{0_0} \wedge b_{0_1}$

【0036】 最後，由於計算出的位元組爲單純兩個純文字位元組的函式，駭客可以統計分析純文字內容之出現頻率，以嘗試求得純文字位元組的數值。

【0037】 然而，根據第2圖以及第3圖所揭露方式計算出的加密指令數據106位元組之圖樣如以下說明(4)與(5)所示：

$$(4)\quad b_{n_0} \wedge \left(k_{n_x} \pm k_{0_y}\right), \cdots, b_{1_0} \wedge \left(k_{1_x} \pm k_{2_y}\right), b_{0_0} \wedge \left(k_{0_x} \pm k_{1_y}\right)$$

$$(5)\quad b_{n_1} \wedge \left(k_{n_x} \pm k_{1_y}\right), \cdots, b_{1_1} \wedge \left(k_{1_x} \pm k_{3_y}\right), b_{0_1} \wedge \left(k_{0_x} \pm k_{2_y}\right)$$

其中 b_{n_0} 標示所加密之第一16位元組區塊之指令數據的位元組n，b_{n_1} 標示所加密之第二16位元組區塊之指令數據的位元組n，k_{n_x} 標示主密鑰x的位元組n，且 k_{n_y} 標示主密鑰y的位元組n。如前述，主密鑰x與y爲不同密鑰。假定一種實施方式以五個主密鑰暫存器142提供八種主密鑰對234/236組合，2048位元組序列中各位元組是與兩個獨立的主密鑰位元組的一組合進行互斥運算。因此，當加密數據以任何方式於256位元組的區塊中移位並且彼此作互斥運算，所求得的位元組都會存在兩個主密鑰的複雜成分，因此，不若說明(3)的內容，此處所得的運算結果不單純只是純文字位元組。例如，假設駭客選擇使同一256位元組區塊中的16位元組區塊對齊並彼此進行互斥操作使同樣的密鑰零位元組在各段中被使用，位元組0之運算結果如說明(6)所示，所獲得的位元組存在兩個主密鑰的複雜組合：

$$(6)\quad b_{0_0} \wedge \left(k_{0_x} \pm k_{1_y}\right) \wedge b_{0_1} \wedge \left(k_{0_x} \pm k_{n_y}\right),$$

其中n不爲1。

【0038】 再者，若駭客換成將選自不同256位元組區塊內的16位元組區塊對齊、且彼此作互斥運算，運算結果的位元組0如說明(7)所示：

(7) $b0_0 \wedge \left(k0_x \pm k1_y\right) \wedge b0_1 \wedge \left(k0_u \pm kn_v\right)$，

其中主密鑰u與v中至少一者不同於主密鑰x以及y。模擬隨機主密鑰數值所產生之有效密鑰位元組之互斥運算，可發現運算結果$\left(k0_x \pm k1_y\right) \wedge \left(k0_{ux} \pm kn_v\right)$呈現相當平滑的分布。

【0039】 當然，若駭客選擇將不同的2048位元組長度區塊內的16位元組區塊對齊、並且彼此進行互斥操作，駭客可能會獲得與說明(3)類似的結果。然而，請參照以下內容。第一，某些程式—例如，安全性相關程式—可能短於2048位元組。第二，相距2048位元組的指令位元組之統計相關性(statistical correlation)很可能非常小，導致很難破解。第三，如前述內容，所述技術之實施方式可以較多數量實現主密鑰暫存器142，使解密密鑰之有效長度擴展；例如，以12個主密鑰暫存器142供應16384位元組長度的解密密鑰，甚至其他更長的解密密鑰。第四，以下將討論的密鑰下載指令500以及密鑰切換指令600更使程式設計師得以載入新的數值至主密鑰暫存器142，以有效擴展密鑰長度超過2048位元組，或者，如果必要，也可擴展密鑰長度至程式的完整長度。

【0040】 現在，參考第4圖，一方塊圖根據本發明技術圖解第1圖的標誌暫存器128。根據第4圖所示之實施方式，標誌暫

存器128包括標準x86暫存器的複數個位元408；不過，爲了此處敘述的新功能，第4圖所示實施方式會動用x86架構中一般爲預留(RESERVED)的一位元。特別說明之，標誌暫存器128包括一E位元欄位402。E位元欄位402用於修復控制暫存器144的E位元148數值，用以於加密以及純文字程式間切換以及/或於不同加密程式間切換，以下將詳細討論之。E位元欄位402標示目前所執行的程式是否有加密。若目前所執行的程式有加密，E位元欄位402爲設定狀態，否則，爲清除狀態。當中斷事件發生，控制權切換給其他程式(例如，中斷interrupt、異常exception如頁錯誤page fault、或任務切換task switch)，儲存標誌暫存器128。反之，若控制權重回先前因中斷事件中斷的程式，則修復標誌暫存器128。微處理器100之設計會在標誌暫存器128修復時以標誌暫存器128之E位元402欄位數值更新控制暫存器144之E位元148數值，以下將詳細討論之。因此，若中斷事件發生時一加密程式正在執行(即提取單元104處於解密模式)，當控制權交還給該加密程式時，以修復的E位元欄位402令E位元148爲設定狀態，以修復提取單元104爲解密模式。在一種實施方式中，E位元148以及E位元欄位402爲同一個具體硬體位元，因此，儲存標誌暫存器128的E位元欄位402中數值即是儲存E位元148，且修復標誌暫存器128的E位元欄位402的數值即是修復E位元148。

【0041】 參閱第5圖，一方塊圖圖解根據本發明技術所實現的一密鑰載入指令500之格式。密鑰載入指令500包括一操作碼(opcode)502欄位，特地標示其爲微處理器100指令集內的密鑰

載入指令500。在一種實施方式中，操作碼欄位502數值爲0FA6/4(x86領域)。密鑰載入指令500包括兩個運算元：一密鑰暫存器檔案目標位址504以及一安全存儲區來源位址506。該安全存儲區來源位址506爲安全存儲區122中儲存一16位元組主密鑰的一位址。密鑰暫存器檔案位址504標示密鑰暫存器檔案124內的一個暫存器的位址，此暫存器將載入自安全存儲區122載出之16位元組主密鑰。在一種實施方式中，若一程式企圖在微處理器100不爲安全操作模式下執行密鑰載入指令500，則視之爲無效指令異常；此外，若安全存儲區來源位址506數值位於有效安全存儲區122之外，則視之爲一般保護異常。在一種實施方式中，若一程式試圖在微處理器100不爲最高權限級別時(例如，x86環0權限/x86 ring 0)執行密鑰載入指令500，則視之爲無效指令異常。在某些狀況下，16位元組主密鑰之構成可能包括在加密指令的即時數據字段內。所述即時數據可被一塊一塊移至安全存儲區122組成16位元組的密鑰。

【0042】 現在，參閱第6圖，一方塊圖圖解根據本發明技術所實現的一密鑰切換指令600之格式。密鑰切換指令600包括一操作碼602欄位，特指其爲微處理器100指令集內的密鑰切換指令600。密鑰切換指令600更包括一密鑰暫存器檔案索引欄位604，標示密鑰暫存器檔案124一序列暫存器中的開端，以自此將密鑰載入主密鑰暫存器142。在一種實施方式中，若一程式嘗試在微處理器100不爲安全操作模式時執行一密鑰切換指令600，則視之爲無效指令異常。在一種實施方式中，若一程式意圖在微處理器100不爲最高權限級別(例如，x86環0權限)時執

行一密鑰切換指令600，則視之爲無效指令異常。在一種實施方式中，密鑰切換指令600爲原子操作型式(atomic)，即不可中斷；此處所討論，用於載入密鑰至主密鑰暫存器142的其他指令也是如此一例如，以下將討論的分支與切換密鑰指令。

【0043】 現在，參閱第7圖，一流程圖圖解第1圖之微處理器100之操作，其中，根據本發明技術執行第6圖介紹的密鑰切換指令600。流程始於方塊702。

【0044】 在方塊702，解碼單元108將一密鑰切換指令600解碼，且將解碼結果代入微代碼單元132內實現密鑰切換指令600的微代碼程序。流程接著進入方塊704。

【0045】 在方塊704，微代碼會根據密鑰暫存器檔案索引欄位604自密鑰暫存器檔案124下載主密鑰暫存器142的內容。較佳實施方式是：微代碼以密鑰暫存器檔案索引欄位604所標示的密鑰暫存器爲起始，自密鑰暫存器檔案124下載連續的n個暫存器內容作爲n個密鑰存入主密鑰暫存器142，其中n爲主密鑰暫存器142的總數。在一種實施方式中，數值n可標示於密鑰切換指令600的一額外空間，設定爲少於主密鑰暫存器142的總數。流程接著進入方塊706。

【0046】 在方塊706，微代碼使微處理器100分支至接續的x86指令(即該密鑰切換指令600之後的指令)，將導致微處理器100中較密鑰切換指令600新的所有x86指令被清空，致使微處理器100內、較切換至接續x86指令的微操作新的所有微操作被清空。上述被清空的指令包括自指令快取記憶體102提取出、緩衝暫存於提取單元104以及解碼單元108內等待解密與解碼

的所有指令位元組106。流程接著進入方塊708。

【0047】 在方塊708，基於方塊706分支至接續指令的操作，提取單元104開始利用方塊704載入主密鑰暫存器142的新一組密鑰值自指令快取記憶體102提取並且解密指令數據106。流程結束於方塊708。

【0048】 如第7圖所示，密鑰切換指令600令正在執行中的加密程式在自指令快取記憶體102提取出來的同時得以改變主密鑰暫存器142內所儲存、供解密該加密程式使用的內容。所述主密鑰暫存器142動態調整技術使得加密該程式的有效密鑰長度超越提取單元104先天支援的長度(例如，第2圖實施方式所提供的2048位元組)；如第8圖所示程式，若將之以第1圖微處理器100操作，駭客會更不易攻破電腦系統的安全防護。

【0049】 現在，參閱第8圖，一方塊圖圖解根據本發明技術所實現的一加密程式的一記憶體用量(memory footprint)800，其中採用第6圖所示之密鑰切換指令600。第8圖所示之加密程式記憶體用量800包括連續數「塊chunk」指令數據位元組。每一「塊」的內容爲一序列多個指令數據位元組(其中爲預先加密的數據)，且屬於同一「塊」的指令數據位元組是由同樣的一套主密鑰暫存器142數值解密。因此，不同兩「塊」的界線是由密鑰切換指令600定義。也就是說，各「塊」的上、下界是由密鑰切換指令600之位置區分(或者，以一程式的第一「塊」爲例，其上界爲該程式的起始處；此外，以該程式的最後一「塊」爲例，其下界爲該程式的結束處)。因此，各「塊」指令數據位元組是由提取單元104基於不同套主密鑰暫存器142數值解

密，意即各「塊」指令數據位元組的解密是根據前一「塊」所供應的一密鑰切換指令600所載入主密鑰暫存器142數值。加密一程式的後處理器(post-processor)會知曉各密鑰切換指令600所在之記憶體位址，並且會利用此資訊—即提取位址的相關位址位元—配合密鑰切換指令600密鑰數值產生加密密鑰位元組，以加密該程式。一些目的檔格式(object file format)允許程式設計者標示程式載入記憶體何處，或至少載明特定大小的對齊形式(例如，頁面邊界page boundary)，以提供足夠的位址資訊加密該程式。此外，一些作業系統預設值是將程式載入頁面邊界上。

【0050】 密鑰切換指令600可安置於程式的任何地方。然而，若密鑰切換指令600載入特定值至主密鑰暫存器142供下一「塊」指令數據位元組解密使用、且密鑰切換指令600(或甚至密鑰載入指令500)之位置導致每一「塊」之長度短於、或等於提取單元104所能應付的有效密鑰長度(例如，第2圖實施方式所揭露的2048位元組)，則程式可被以有效長度等同整體程式長度的密鑰加密，此爲相當強健的加密方式。此外，即使密鑰切換指令600的使用使得有效密鑰長度仍短於加密程式的長度(即，同樣一套主密鑰暫存器142數值被用於加密一程式的多個「塊」)，改變「塊」尺寸(例如，不限定全爲2048位元組)可增加駭客破解系統的困難度，因爲，駭客必須先判斷以同一套主密鑰暫存器142數值加密的「塊」位於何處，並且必須判斷該些長度不一的「塊」各自的尺寸。

【0051】 值得注意的是，以密鑰切換指令600實現的動態密

鑰切換耗費相當大量的時脈數目，主要是因爲管線必須清空。此外，在一種實施方式中，密鑰切換指令600主要是以微代碼(microcode)實現，通常較非微代碼實現的指令慢。因此，程式碼開發者須考慮密鑰切換指令對效能的影響，在執行速度以及特定應用之安全性考量之間尋求平衡點。

【0052】 現在，參閱第9圖，一方塊圖圖解根據本發明技術實現的一分支與切換密鑰指令900的格式。首先敘述該分支與切換密鑰指令900的必要性。

【0053】 根據以上實施例所揭露內容，加密程式交由提取單元104提取的各個16位元組區塊的指令數據是有先經過加密運算(採互斥技術)，所採用的加密密鑰等同提取單元104用來解密(互斥運算)所提取之各區塊之指令數據106的各個16位元組長之解密密鑰174。如以上所述，解密密鑰174的位元組數值是由提取單元104基於以下兩種輸入計算而得：儲存於主密鑰暫存器142的主密鑰位元組數值、以及所提取之16位元組區塊之指令數據106的提取位址134的部分位元(以第2圖所揭露實施方式爲例，爲位元[10:4])。因此，加密一程式使之由微處理器100執行的一後處理器會知曉將儲存於主密鑰暫存器142的主密鑰位元組數值、以及一位址(或更限定爲該位址的數個相關位元)；該位址指示加密程式將被載入記憶體何處、且微處理器100將自此處一連串地提取出該加密程式數個區塊的指令數據。基於上述資訊，後處理器得以適切產生解密密鑰174數值，用於加密該程式的各個16位元組區塊之指令數據。

【0054】 如以上所討論，當一分支指令被預測到且/或被執

行，提取單元104會以分支目標位址更新提取位址134。只要加密程式從未改變(經由密鑰切換指令600)主密鑰暫存器142內儲存的主密鑰數值，分支指令是由提取單元104透明控制。也就是說，提取單元104會採用同樣的主密鑰暫存器142數值估算解密密鑰174，以供解密包括該分支指令的一區塊之指令數據106、以及解密該分支指令之目標位址所指的一區塊之指令數據106內的指令。然而，程式改變(經由密鑰切換指令600)主密鑰暫存器142數值的能力意味著提取單元104有可能以一套主密鑰暫存器142數值估算解密密鑰174解密包括該分支指令的一區塊之指令數據106，並以不同的另外一套主密鑰暫存器142數值估算解密密鑰174解密該分支指令之目標位址所指的一區塊之指令數據106內的指令。解決此問題的一種方法是限定分支目標位址於程式同一「塊」中。另外一種解決方式是採用第9圖所揭露的分支與切換密鑰指令900。

【0055】 再次參閱第9圖，一方塊圖圖解根據本發明技術實現的一分支與切換密鑰指令900的格式。分支與切換密鑰指令900包括一操作碼902欄位，標示其爲微處理器100指令集內的分支與切換密鑰指令900。分支與切換密鑰指令900更包括一密鑰暫存器檔案索引欄位904，標示密鑰暫存器檔案124中一連串暫存器裡的開端，以自此將密鑰載入主密鑰暫存器142。分支與切換密鑰指令900更包括一分支資訊欄位906，記載分支指令的典型資訊—如，計算目標位址的資訊、以及分支條件。在一種實施方式中，若一程式在微處理器100不爲安全執行模式時嘗試執行一分支與切換密鑰指令900，則視之爲無效指令異

常。在一種實施方式中，若一程式在微處理器100不爲最高權限層級(例如，x86的環0權限)時試圖執行分支與切換密鑰指令900，則視之爲無效指令異常。在一種實施方式中，分支與切換密鑰指令900爲原子操作型(atomic)。

【0056】 參閱第10圖，一流程圖圖解第1圖微處理器100之操作，其中，根據本發明技術執行第9圖所揭露之分支與切換密鑰指令900。流程始於方塊1002。

【0057】 在方塊1002，解碼單元108解碼一分支與切換密鑰指令900且將之代入微代碼單元132中實現該分支與切換密鑰指令900的微代碼程序。流程接著進入方塊1006。

【0058】 在方塊1006，微代碼解出分支方向(採用、或不採用)、以及目標位址。值得注意的是，對於無條件型分支指令(unconditional branch instruction)，所述方向衡爲採用。流程接著進入判斷方塊1008。

【0059】 在判斷方塊1008，微代碼判斷方塊1006所解出的方向是否爲採用。若爲採用，流程進入方塊1014。反之，流程進入方塊1012。

【0060】 在方塊1012，微代碼不切換密鑰、或跳至目標位址，因爲分支操作未被採用。流程結束於方塊1012。

【0061】 在方塊1014，微代碼根據密鑰暫存器檔案索引欄位904，將密鑰自密鑰暫存器檔案124載入主密鑰暫存器142。較佳實施例是，微代碼以密鑰暫存器檔案索引欄位904所標示的位置爲起始，將密鑰暫存器檔案124內n個鄰近暫存器所記載的n個密鑰載入主密鑰暫存器142，其中n爲主密鑰暫存器142的

總數。在一種實施方式中，n值可紀錄於分支與切換密鑰指令900的一額外空間，設定爲小於主密鑰暫存器142總數的值。流程接著進入方塊1016。

【0062】 在方塊1016，微代碼使得微處理器100跳至方塊1006所解出的目標位址，將導致微處理器100中較分支與切換密鑰指令900新的所有x86指令被清空，致使微處理器100內、較分支至目標位址的微操作新的所有微操作被清空。上述被清空的指令包括自指令快取記憶體102提取出、緩衝暫存於提取單元104以及解碼單元108內等待解密與解碼的所有指令位元組106。流程接著進入方塊1008。

【0063】 在方塊1018，隨著方塊1016分支至目標位址的操作，提取單元104採用方塊1014載入主密鑰暫存器142的新一組密鑰數值開始自指令快取記憶體102提取且解密指令數據106。流程結束於方塊1018。

【0064】 現在，參閱第11圖，一流程圖圖解根據本發明技術所實現的一後處理器的操作。所述後處理器爲軟件工具，可用於後處理一程式並加密之，以交由第1圖的微處理器100執行。流程始於方塊1102。

【0065】 在方塊1102，後處理器接收一程式的一目的檔。根據一種實施方式，該目的檔內的分支指令的目標位址可在程式執行前確定；例如，指向固定目標位址的分支指令。在程式運行前決定好目標位址的分支指令尚有另一形式，例如，一相對分支指令(relative branch instruction)，其中記載一偏移量，用來加上分支指令所在之記憶體位址，以求得分支目標位址。

反之，關於目標位址不會在程式執行前確定的分支指令，其中一種例子是基於暫存器或記憶體所儲存的運算元計算出目標位址，因此，其值在程式執行當中可能有變動。流程接著進入方塊1104。

【0066】 在方塊1104，後微處理器將跨塊分支指令(inter-chunk branch instruction)以分支與切換密鑰指令900取代，所述指令900在密鑰暫存器檔案索引空間904儲存有適當的數值，該數值乃基於分支指令之目標位址所坐落的「塊」而設定。如第8圖所揭露內容，一「塊」是由一序列多個指令數據位元組所組成，將由同一套主密鑰暫存器142數值解密。因此，跨塊分支指令之目標位址所坐落的「塊」不同於分支指令本身的「塊」。值得注意的是，塊內分支—即目標位址與本身位於同一「塊」的分支指令—無須被替代。值得注意的是，產生出原始檔(source file)以產出目的檔的程式設計及/或編譯器可視需求明確包括分支與切換密鑰指令900，以降低後處理器取代操作的負擔。流程接著進入方塊1106。

【0067】 在方塊1106，後處理器加密該程式。後處理器知道每一「塊」之記憶體位置以及主密鑰暫存器142數值，並將之用於加密該程式。流程結束於方塊1106。

【0068】 現在，參閱第12圖，一方塊圖圖解本發明技術另一種實施方式所實現的一分支與切換密鑰指令1200之格式。第12圖所示之分支與切換密鑰指令1200適用於目標位址在程式執行前爲未知的分支操作，以下將詳細討論之。分支與切換密鑰指令1200包括一操作碼1202欄位，用以標示其爲微處理器

100指令集內的分支與切換密鑰指令1200。分支與切換密鑰指令1200同樣包括一分支資訊欄位906，功用與第9圖之分支與切換密鑰指令900的該欄位類似。在一種實施方式中，若一程式在微處理器100不爲安全執行模式時試圖執行分支與切換密鑰指令1200，則視之爲無效指令異常。在一種實施方式中，若一程式在微處理器100不爲最高權限級別(例如，x86環0權限)時試圖執行一分支與切換密鑰指令1200，則視之爲無效指令異常。在一種實施方式中，分支與切換密鑰指令1200爲原子型式。

【0069】 現在，參閱第13圖，一方塊圖圖解根據本發明技術實現的「塊」位址範圍表1300。表格1300包括多個單元。每一單元與加密程式的一「塊」相關。每一單元包括一位址範圍欄位1302以及一密鑰暫存器檔案索引欄位1304。位址範圍欄位1302標示所對應「塊」的記憶體位址範圍。密鑰暫存器檔案索引欄位1304標示密鑰暫存器檔案124內的暫存器，由分支與切換密鑰指令1200將索引所指的暫存器所儲存的密鑰數值載入主密鑰暫存器142，供提取單元104解密該「塊」使用。以下參考第18圖進行討論，表格1300於需要存取表格1300內容的分支與切換密鑰指令1200執行前載入微處理器100。

【0070】 現在，參閱第14圖，一流程圖圖解第1圖微處理器100的操作，其中，根據本發明技術執行第12圖的分支與切換密鑰指令1200。流程始於方塊1402。

【0071】 在方塊1402，解碼單元108解碼一分支與切換密鑰指令1200且將之代入微代碼單元132中實現分支與切換密鑰指令1200的微代碼程序。流程接著進入方塊1406。

【0072】 在方塊1406，微代碼解出分支方向(採用、或不採用)、且找出目標位址。流程接著進入判斷方塊1408。

【0073】 在判斷方塊1408，微代碼判斷方塊1406所解出的分支方向是否爲採用。若爲採用，流程進入方塊1414。反之，流程進入方塊1412。

【0074】 在方塊1412，微代碼不切換密鑰、或跳至目標位址，因爲該分支未被採用。流程結束於方塊1412。

【0075】 在方塊1414，微代碼基於方塊1406所解出的目標位址查詢第13圖所示之表格1300，得到該目標位址所坐落之「塊」所對應之密鑰暫存器檔案索引欄位1304的內容。微代碼接著基於密鑰暫存器檔案索引欄位1304內所記載的索引，自密鑰暫存器檔案124將密鑰數值載入主密鑰暫存器142。較佳實施方式是，微代碼根據密鑰暫存器檔案索引欄位1304所儲存的索引，自密鑰暫存器檔案124將n個相鄰暫存器儲存的n個密鑰值載入主密鑰暫存器142的，其中，n爲主密鑰暫存器142的總數。在一種實施方式中，數值n可紀錄於分支與切換密鑰指令1200的一額外欄位中，設定爲少於主密鑰暫存器142總數。流程接著進入方塊1416。

【0076】 在方塊1416，微代碼致使微處理器100分支至方塊1406所解出的目標位址，將導致微處理器100中較分支與切換密鑰指令1200新的所有x86指令被清空，致使微處理器100內、較分支至目標位址的微操作新的所有微操作被清空。上述被清空的指令包括自指令快取記憶體102提取出、緩衝暫存於提取單元104以及解碼單元108內等待解密與解碼的所有指令位元

組106。流程接著進入方塊1418。

【0077】 在方塊1418，隨著方塊1416分支至目標位址的操作，提取單元104採用方塊1414載入主密鑰暫存器142的新一套密鑰值，開始自指令快取記憶體102提取並且解密指令數據106。流程結束於方塊1418。

【0078】 現在，參考第15圖，一方塊圖圖解根據本發明技術另外一種實施方式所實現的一分支與切換密鑰指令1500的格式。第15圖所示之分支與切換密鑰指令1500以及其操作類似第12圖所示之分支與切換密鑰指令1200。然而，取代自密鑰暫存器檔案124載入密鑰至主密鑰暫存器142，分支與切換密鑰指令1500是自安全存儲區122載入密鑰至主密鑰暫存器142，以下討論之。

【0079】 現在，參考第16圖，一方塊圖圖解根據本發明技術所實現的「塊」位址範圍表1600。第16圖所示表格1600類似第13圖所示之表格1300。然而，取代包括一密鑰暫存器檔案索引欄位1304，表格1600包括一安全存儲區位址欄位1604。安全存儲區位址欄位1604記載安全存儲區122內的一位址，該位址儲存的密鑰值須由分支與切換密鑰指令1500載入主密鑰暫存器142，以供該提取單元104解密該「塊」時使用。以下討論參考第18圖內容，表格1600是在需要查詢該表格1600的分支與切換密鑰指令1500被執行前載入微處理器100。在一種實施方式中，安全存儲區122位址之較低數個位元無須儲存在安全存儲區位址欄位1604，特別是因爲安全存儲區122中儲存一組密鑰的位置之總量相當大(例如，16位元組x 5)、且該組密鑰可沿著

一設定尺寸範圍對齊。

【0080】 現在，參閱第17圖，一流程圖圖解第1圖微處理器100的操作，其中根據本發明技術執行第15圖的分支與切換密鑰指令1500。流程始於方塊1702。第17圖之流程圖的許多方塊與第14圖的許多方塊類似，因此採同樣的編號。然而，方塊1414是由方塊1714取代，微代碼基於方塊1406所求得的目標位址查表第16圖之表格1600，以獲得目標位址所坐落的「塊」之安全存儲區位址欄位1604數值。微代碼接著根據安全存儲區位址欄位1604數值自安全存儲區122將密鑰數值載入主密鑰暫存器142。較佳實施方式是，微代碼由安全存儲區位址欄位1604數值自安全存儲區122將n個鄰近16位元組空間位置內所儲存的n個密鑰數值載入主密鑰暫存器142，其中n爲主密鑰暫存器142的總數。在一種實施方式中，數值n可記載於分支與切換密鑰指令1500中一額外欄位，設定爲少於主密鑰暫存器142總數。

【0081】 現在，參閱第18圖，一流程圖圖解根據本發明另外一種實施方式所實現的一後處理器的操作。所述後處理器可用於後處理一程式並加密之，以交由第1圖的微處理器100執行。流程始於方塊1802。

【0082】 在方塊1802，後處理器接收一程式的目的檔。根據一種實施方式，該目的檔內的分支指令，可爲目標位址在程式執行前判定、可爲目標位址不可在程式執行前判定。流程接著進入方塊1803。

【0083】 在方塊1803，後處理器建立第13圖或第16圖之「塊」位址範圍表1300或1600，以列入該目標檔。在一種實施

方式中，作業系統在載入且執行一加密程式前將表格1300/1600載入微處理器100，使分支與切換密鑰指令1200/1500得以存取之。在一種實施方式中，後處理器在程式中插入指令，以在任何分支與切換密鑰指令1200/1500執行前載入表格1300/1600至微處理器100。流程接著進入方塊1804。

【0084】 在方塊1804，類似先前所討論、關於第11圖之方塊1104的操作，後處理器將每個執行前目標位址可決定的跨塊分支指令以第9圖的分支與切換密鑰指令900取代，指令900基於分支指令目標位址所在「塊」記載有合適的密鑰暫存器檔案索引欄位904數值。流程接著進入方塊1805。

【0085】 在方塊1805，後處理器根據方塊1803所產生的表格型態(1300/1600)將每個限於執行過程中決定目標位址的分支指令以第12圖或第15圖所示之分支與切換密鑰指令1200或1500取代。流程接著進入方塊1806。

【0086】 在方塊1806，後處理器加密該程式。該後處理器知道關於各「塊」的記憶體位置與主密鑰暫存器142數值，將用於加密該程式。流程結束於方塊1806。

【0087】 現在，參閱第19圖，一流程圖圖解第1圖微處理器100的操作，其中，根據本發明技術處理加密程式以及純文字程式之間的任務切換。流程始於方塊1902。

【0088】 在方塊1902，標誌暫存器128的E位元欄位402的E位元以及第1圖控制暫存器144之E位元148由微處理器100的一重置操作清空。流程接著進入方塊1904。

【0089】 在方塊1904，微處理器100在執行其重置微代碼進

行初始化後，開始提取並且執行使用者程式指令(例如，系統韌體)，其爲純文字程式指令。特別是，由於E位元128爲清空，如前所述，提取單元104視提取出來的指令數據106爲純文字指令。流程接著進入方塊1906。

【0090】 在方塊1906，系統韌體(例如，作業系統、韌體、基本輸入輸出系統BIOS…等)接收一要求(request)，要執行一加密程式。在一種實施方式中，執行一加密程式的上述要求伴隨、或由一切換操作指示，以切換至微處理器100的一安全執行模式，如以上討論內容。在一種實施方式中，微處理器100僅在安全執行模式時，方允許操作於一解密模式(即，E位元148爲設定狀態)。在一種實施方式中，微處理器100僅在系統管理模式(system management mode，例如，x86架構中常見的SSM)，方允許以解密模式操作。流程接著進入方塊1908。

【0091】 在方塊1908，系統軟體於主密鑰暫存器142中載入其初始值，與程式中將被執行的第一「塊」相關。在一種實施方式中，系統軟體執行一密鑰切換指令600下載密鑰至主密鑰暫存器142。在載入密鑰至主密鑰暫存器142之前，密鑰暫存器檔案124的內容可由一或多個密鑰載入指令500載入。在一種實施方式中，載入密鑰至主密鑰暫存器142以及密鑰暫存器檔案124之前，安全存儲區122可先被寫入密鑰數值，其中，所述寫入乃經由常見的安全通道技術，例如，AES或RSA加密通道，以防止駭客窺探其值。如以上所討論，以上密鑰數值可儲存在一安全非揮發性記憶體(例如快閃記憶體)經由一隔離串行總線(private serial bus)耦接微處理器100，或者，可儲存在微處理

器100的一非揮發性單次寫入記憶體。如以上討論，所述程式可包含在單一「塊」中。也就是說，所述程式可不包括密鑰切換指令600，整個程式可由單一套主密鑰暫存器142數值解密。流程接著進入方塊1916。

【0092】 在方塊1916，隨著控制權轉移至加密程式，微處理器100設定標誌暫存器128的E位元欄位402標示目前所執行的程式爲加密型式，且設定控制暫存器144的E位元148，使提取單元104處於解密模式。微處理器100更致使管線內的指令被刷新，其動作類似第7圖方塊706所實行的刷新操作。流程接著進入方塊1918。

【0093】 在方塊1918，提取單元104提取加密程式內的指令106，並且參考第1圖至第3圖所揭露的技術將之以解密模式解密並且執行之。流程接著進入方塊1922。

【0094】 在方塊1922，微處理器100提取並且執行加密程式時，微處理器100接收到中斷事件。舉例說明之，所述中斷事件可爲一中斷interrupt、一異常exception(如頁面錯誤page fault)、或任務切換task switch。當一中斷事件發生，微處理器100管線所有待處理的指令會被清空。所以，若管線中有任何先前提取的加密指令，將之清空。此外，自指令快取記憶體102所提取出、可能在緩衝儲存在提取單元104以及解碼單元108中等待被解密、解碼的所有指令位元組會被清空。在一種實施方式中，微代碼被喚起回應中斷事件。流程接著進入方塊1924。

【0095】 在方塊1924，微處理器100儲存標誌暫存器128(以及微處理器100其他結構狀態，包括受中斷的加密程式的目前

指令指標數值)至一堆疊式記憶體(stack memory)。儲存加密程式之E位元欄位402數值將使其得以在後續操作中修復(在方塊1934)。流程接著進入方塊1926。

【0096】 在方塊1926，當控制權轉移到新的程式(例如，中斷處理程序interrupt handler、異常處理程序exception handler、或新任務)，微處理器100清空標誌暫存器128的E位元欄位402、以及控制暫存器144的E位元148，以應付純文字的新程式。也就是說，第19圖所示實施例假設微處理器100同一時間只有允許運作一個加密程式，且已有一個加密程式在執行(但被中斷)。第22圖至第26圖另外揭露有其他種的實施方式。流程接著進入方塊1928。

【0097】 在方塊1928，提取單元104參考第1圖至第3圖所揭露內容以純文字模式提取新程式的指令106。特別是，控制暫存器144內E位元148的清空狀態使得多工器154將指令數據106與多位元的二進位零值176進行互斥運算，使得指令數據106不被解密操作。流程接著進入方塊1932。

【0098】 在方塊1932，新程式執行一返回操作自中斷指令(例如，x86 IRET)或類似指令返回，使得控制權回歸加密程式。在一種實施方式中，自中斷指令返回的操作由微代碼實現。流程接著進入方塊1934。

【0099】 在方塊1934，回應前述自中斷指令返回的操作，由於控制權移轉回加密程式，微處理器100修復標誌暫存器128，令標誌暫存器128之E位元欄位402重回先前方塊1924所儲存的設定狀態。流程接著進入方塊1938。

【0100】 在方塊1938，由於控制權移轉回加密程式，微處理器100以標誌暫存器128的E位元欄位402數值更新控制暫存器144的E位元148，使得提取單元104重新提取並且解密該加密程式之指令數據106。流程接著進入方塊1942。

【0101】 在方塊1942，微代碼令微處理器100分支至先前方塊1924儲存於堆疊式記憶體中的指令指標數值，使得微處理器100中所有x86指令清空、且使得微處理器100中所有微操作清空。所清空內容包括提取自指令快取記憶體102、緩衝暫存在提取單元104以及解碼單元108中等待被解密、解碼的所有指令位元組106。流程接著進入方塊1944。

【0102】 在方塊1944，提取單元104重新開始提取該加密程式內的指令106，並且參考第1圖至第3圖所揭露技術以解密模式解密並且執行之。流程結束於方塊1944。

【0103】 現在，參考第20圖，一流程圖圖解根據本發明技術實現的一系統軟體之操作，由第1圖之微處理器100執行。第20圖流程可配合第19圖內容執行。流程始於方塊2002。

【0104】 在方塊2002，系統軟體收到一要求，欲執行一個新的加密程式。流程接著進入決策方塊2004。

【0105】 在決策方塊2004，系統軟體判斷此一加密程式是否爲系統已在執行的程式之一。在一種實施方式中，系統軟體以一旗標標示一加密程式是否爲系統中已在執行的程式之一。若此加密程式是系統已在執行的程式之一，流程進入方塊2006，反之，則流程進入方塊2008。

【0106】 在方塊2006，系統軟體等待該加密程式執行完畢

且不再是系統執行中的程式之一。流程接著進入方塊2008。

【0107】 在方塊2008，微處理器100允許新的加密程式開始執行。流程結束於方塊2008。

【0108】 現在，參考第21圖，一方塊圖根據本發明技術另外一種實施方式，圖解第1圖標誌暫存器128的欄位。第21圖的標誌暫存器128類似第4圖所示實施方式，相比之，更包括索引欄位(index bits)2104。根據一種實施方式，索引欄位2104(類似E位元402)通常是x86架構所預留的位元。索引欄位2104用於應付多個加密程式的切換，以下詳細討論之。較佳實施方式是，密鑰切換指令600以及分支與切換密鑰指令900/1200以本身的密鑰暫存器索引欄位604/904/1304更新標誌暫存器128的索引欄位2104。

【0109】 現在，參考第22圖，一流程圖圖解第1圖微處理器100的操作，其中，根據本發明技術採用第21圖所示之標誌暫存器128實行多個加密程式之間的任務切換。流程接著進入方塊2202。

【0110】 在方塊2202，一要求發向該系統軟體，要執行一個新的加密程式。流程接著進入決策方塊2204。

【0111】 在決策方塊2204，系統軟體判斷密鑰暫存器檔案124中是否有空間應付一個新的加密程式。在一種實施方式中，方塊2202所產生的該要求會指出需要密鑰暫存器檔案124內多少空間。若密鑰暫存器檔案124中有空間應付新的加密程式，流程進入方塊2208，反之，流程進入方塊2206。

【0112】 在方塊2206，系統軟體等待一或多個加密程式完

成、使密鑰暫存器檔案124騰出空間應付新的加密程式。流程接著進入方塊2208。

【0113】 在方塊2208，系統軟體將密鑰暫存器檔案124內的空間配置給新的加密程式，並且隨之填寫標誌暫存器128中的索引欄位2104，以標示密鑰暫存器檔案124中新配置的空間。流程接著進入方塊2212。

【0114】 在方塊2212，系統軟體在方塊2208所配置的密鑰暫存器檔案124位置載入供新程式使用的密鑰數值。如以上討論，所載入的密鑰數值可採用密鑰載入指令500自安全存儲區122載入，或者，在必要情況下，可以安全管道由微處理器100外部位置取得。流程接著進入方塊2214。

【0115】 在方塊2214，系統軟體基於密鑰暫存器檔案索引欄位604/904/1304將密鑰自密鑰暫存器檔案124載入主密鑰暫存器142。在一種實施方式中，系統軟體執行一密鑰切換指令600載入密鑰至主密鑰暫存器142。流程接著進入方塊2216。

【0116】 在方塊2216，由於控制權移轉至加密程式，微處理器100設定標誌暫存器128之E位元欄位402以標示目前執行的程式爲加密型式，並且設定控制暫存器144的E位元148以設定提取單元104爲解密模式。流程結束於方塊2216。

【0117】 現在，參考第23圖，一流程圖圖解第1圖微處理器100的操作，其中，根據本發明技術採用第21圖所示之標誌暫存器128應付多個加密程式之間的任務切換。流程始於方塊2302。

【0118】 在方塊2302，目前執行的程式執行一返回操作，

自一中斷指令返回，引發一任務切換至新程式；所述新程式先前曾被執行過但被跳開，且其結構狀態(例如，標誌暫存器128、指令指標暫存器、以及通用暫存器)曾被儲存在堆疊式記憶體中。如先前所提過，在一種實施方式中，自中斷指令返回的操作是由微代碼實現。現在執行中的程式以及新的程式可爲加密程式或純文字程式。流程進入方塊2304。

【0119】 在方塊2304，微處理器100根據堆疊式記憶體修復標誌暫存器128，以應付接續返回的程式。也就是說，微處理器100將接續程式(即目前跳換回的程式)先前跳換出去時儲存於堆疊式記憶體的標誌暫存器128數值重新載入標誌暫存器128。流程接著進入決策方塊2306。

【0120】 在決策方塊2306，微處理器100判斷修復後的標誌暫存器128之E位元402是否爲設定狀態。若是，則流程進入方塊2308；反之，則流程進入方塊2312。

【0121】 在方塊2308，微處理器100根據方塊2304所修復的EFLAGS暫存器128索引欄位2104數值將密鑰載入密鑰暫存器檔案124。流程接著進入方塊2312。

【0122】 在方塊2312，微處理器100將控制暫存器144之E位元148的內容以方塊2304所修復的標誌暫存器128之E位元欄位402數值更新。因此，若接續的程式是一個加密程式，提取單元104會被設定爲解密模式，反之，則設定爲純文字模式。流程接著進入方塊2314。

【0123】 在方塊2314，微處理器100以堆疊式記憶體的內容修復指令指標暫存器、並且分支跳躍至指令指標所指的位置，

所述動作將清除微處理器100所有x86指令，並且清除微處理器所有微操作。所清除的包括自指令快取記憶體102所提取出、緩衝暫存於提取單元104、解碼單元108中等待解密、解碼的所有指令位元組106。流程接著進入方塊2316。

【0124】 在方塊2316，提取單元104參考第1圖至第3圖技術重新開始自接續程式中提取指令106，並視方塊2312所修復的控制暫存器144之E位元148數值以解密模式或純文字模式操作。流程結束於方塊2316。

【0125】 現在，參考第24圖，一方塊圖根據本發明、圖解第1圖密鑰暫存器檔案124之單一個暫存器的另外一種實施方式。根據第24圖所示之實施方式，每個密鑰暫存器檔案124更包括一位元—爲淘汰位元2402(kill bit，以下簡稱K位元)。K位元2402用於應付微處理器100對多個加密程式的多任務(multitasking)操作，所述多個加密程式總計需要多於密鑰暫存器檔案124空間尺寸的密鑰儲存空間，以下將詳述之。

【0126】 現在，參考第25圖，一流程圖圖解第1圖微處理器100的操作，其中根據本發明技術以第21圖之標誌暫存器128以及第24圖之密鑰暫存器檔案124實現多個加密程式之間之任務切換的另外一種實施方式。第25圖所示流程類似第22圖所示流程。不同處在於決策方塊2204判定密鑰暫存器檔案124中沒有足夠可用空間時，第25圖流程會進入方塊2506而非不存在於第25圖的方塊2204。另外，若決策方塊2204判定密鑰暫存器檔案124中尚有足夠可用空間，則第25圖流程同樣進入第22圖之方塊2208至方塊2216。

【0127】 在方塊2506，系統軟體將密鑰暫存器檔案124中已經被其他加密程式使用(即已經被配置)的空間(即暫存器)配置出來，並且設定所配置暫存器的K位元2402爲設定狀態，並且隨之設定標誌暫存器128的索引欄位2104以標示新配置空間在密鑰暫存器檔案124中的位置。K位元2402之設定狀態，是標示該暫存器中關於其他加密程式的密鑰值將被方塊2212的操作覆寫爲新的加密程式的密鑰值。然而，如以下第26圖所敘述，其他加密程式的密鑰值將在其返回程序中由方塊2609重新載入。第25圖流程進入方塊2506，會接著導向第22圖所示之方塊2212，結束於方塊2216。

【0128】 現在，參閱第26圖，一流程圖圖解第1圖微處理器100的操作，其中根據本發明技術以第21圖之標誌暫存器128以及第24圖之密鑰暫存器檔案124實現多個加密程式之間之任務切換的另外一種實施方式。第26圖所示流程類似第23圖所示流程。不同處在於，若決策方塊2306判定標誌暫存器128的E位元402爲設定，第26圖令流程進入決策方塊2607而非方塊2308。

【0129】 在決策方塊2607，微處理器100判斷密鑰暫存器檔案124中，由標誌暫存器128索引欄位2104數值(於方塊2304中修復)所標示的任何暫存器之K位元2402是否爲設定狀態。若是，則流程進入方塊2609；若否，則流程進入方塊2308。

【0130】 在方塊2609，微處理器100產生一異常警示(exception)交由一異常處理程序處理。在一種實施方式中，異常處理程序設計於系統軟體中。在一種實施方式中，異常處理程序是由安全執行模式架構提供。根據方塊2304所修復的標誌

暫存器128索引欄位2104數值，異常處理程序將目前修復的加密程式(即現在所返回執行的加密程式)之密鑰重新載入密鑰暫存器檔案124。異常處理程序可類似先前第19圖所提及的方塊1908作動，將修復之加密程式的密鑰載入密鑰暫存器檔案124，或者，在必要情況下，自微處理器100外部將密鑰載入安全存儲區122。同樣地，若密鑰暫存器檔案124中被重新載入的暫存器有被其他加密程式使用，系統軟體會令其暫存器的K位元2402爲設定狀態。流程接著自方塊2609進入2308，且方塊2308至2316是參考第23圖內容。

【0131】 如第24圖至第26圖所教示，此處所敘述的實施方式令微處理器100得以實行多個加密程式的多任務操作，即便上述加密程式需要密鑰暫存空間總合多於密鑰暫存器124空間尺寸。

【0132】 現在，參考第27圖，一方塊圖圖解修改自第1圖微處理器100的本發明另外一種實施方式。與第1圖類似的元件是採用同樣標號；例如，指令快取記憶體102、提取單元104以及密鑰暫存器檔案124。然而，此處提取單元104被修正成更包括密鑰切換邏輯2712，耦接第1圖所介紹之主密鑰暫存器檔案142以及密鑰暫存器檔案124。第27圖之微處理器100更包括一分支目標位址快取記憶體(branch target address cache，BTAC)2702。BTAC 2702接收第1圖所揭露之提取位址134，且與指令快取記憶體102的存取平行，皆是基於該提取位址134。根據提取位址134，BTAC 2702供應分支目標位址2706給第1圖所揭露的提取位址產生器164，供應一採用/不採用指標(T/NT

indicator)2708以及一型式指標(type indicator)2714給密鑰切換邏輯2712，並且供應一密鑰暫存器檔案(KRF)索引2716給密鑰暫存器檔案124。

【0133】 現在，參閱第28圖，一方塊圖根據本發明技術更詳細圖解第27圖的BTAC 2702。BTAC 2702包括一BTAC矩陣2802，其中具有複數個BTAC單元2808，第29圖圖解BTAC單元2808的內容。BTAC 2802儲存的資訊包括先前執行過的分支指令的歷史資訊，以預測接續執行之分支指令的方向以及目標位址。特別是，BTAC 2802會採用儲存的歷史資訊，基於提取的位址134預測先前執行過的分支指令後續發生的提取操作。分支目標位址快取之操作可參考常見的分支預測技術。然而，本發明所揭露的BTAC 2802是更修正成記錄先前執行過的分支與切換密鑰指令900/1200的歷史資訊，以進行相關的預測操作。特別是，儲存的歷史紀錄使得BTAC 2802得以在提取時間內預測所提取的分支與切換密鑰指令900/1200將載入主密鑰暫存器142的該組數值。此操作致能密鑰切換邏輯2712在分支與切換密鑰指令900/1200實際執行前將密鑰數值載入，避免受限於需根據分支與切換密鑰指令900/1200之執行清空微處理器100的管線內容，以下將詳細討論之。此外，根據一種實施方式，BTAC 2802更被修正成儲存包括先前執行過的密鑰切換指令600的歷史資訊，以達到相同的效果。

【0134】 現在，參閱第29圖，一方塊圖根據本發明技術更詳細圖解第28圖BTAC單元2808的內容。每個單元2808包括一有效位元2902指示所屬單元2808是否爲有效。每個單元2808更

包括一標記欄位2904，用以與提取位址134的部分內容比較。若提取位址134的索引部分選擇的單元2808使得提取位址134之標記部分吻合其中有效標記2904，則提取位址134正中BTAC 2802。每個陣列單元2808更包括一目標位址欄位2906，用於儲存先前執行過之分支指令—包括分支與切換密鑰指令900/1200—的目標位址。每個陣列單元2808更包括一採用/不採用欄位2908，用以儲存先前執行過的分支指令—包括分支與切換密鑰指令900/1200—的方向(採用/不採用)記錄。每個陣列單元2808更包括一密鑰暫存器索引欄位2912，用於儲存先前執行過的分支與切換密鑰指令900/1200的密鑰暫存器檔案索引904/1304記錄，以下將詳細討論之。根據一種實施方式，BTAC 2802是在其密鑰暫存器檔案索引欄位2912儲存先前執行過的密鑰切換指令600的密鑰暫存器檔案索引604記錄。每個陣列單元2808更包括一型式欄位2914，指示所紀錄的指令的型式。例如，型式欄位2914可標示所紀錄的歷史指令爲一呼叫(call)、返回(return)、條件跳躍(conditional jump)、無條件跳躍(unconditional jump)、分支與切換密鑰指令900/1200、或密鑰切換指令600。

【0135】 現在，參閱第30圖，一流程圖圖解第27圖微處理器100的操作，其中，根據本發明技術，所述微處理器100包括第28圖揭露的BTAC 2802。流程始於方塊3002。

【0136】 在方塊3002，微處理器100執行一分支與切換密鑰指令900/1200，以下將以第32圖詳述之。流程接著進入方塊3004。

【0137】 在方塊3004，微處理器100在BTAC 2802中配置一陣列單元2808給執行過的分支與切換密鑰指令900/1200，將該分支與切換密鑰指令900/1200解出的方向、目標位址、密鑰暫存器檔案索引904/1304、以及指令型式分別紀錄於所配置的該陣列單元2808之採用/不採用欄位2908、目標位址欄位2906、密鑰暫存器檔案索引欄位2912、以及型式欄位2914中，以作爲該分支與切換密鑰指令900/1200的歷史資訊。流程結束於方塊3004。

【0138】 現在，參閱第31圖，一流程圖圖解第27圖微處理器100的操作，其中，根據本發明技術，所述微處理器100包括第28圖揭露的BTAC 2802。流程始於方塊3102。

【0139】 在方塊3102，提取位址134供應給指令快取記憶體102以及BTCA 2802。流程接著進入方塊3104。

【0140】 在方塊3104，提取位址134正中BTAC 2802，且BTAC 2802將對應的陣列單元2808之目標位址2906、採用/不採用2908、密鑰暫存器檔案索引2912以及型式2914欄位的內容分別以目標位址2706、採用/不採用指標2708、密鑰暫存器檔案索引2716、以及型式指標2714輸出。特別是，型式欄位2914用於指示所儲存指令爲一分支與切換密鑰指令900/1200。流程接著進入決策方塊3106。

【0141】 在決策方塊3106，密鑰切換邏輯2712藉由檢驗採用/不採用輸出2708判斷分支與切換密鑰指令900/1200被BTAC 2802預測爲會採用。若採用/不採用輸出2708顯示分支與切換密鑰指令900/1200被預測爲採用，流程接著進入方塊3112；反

之，流程接著進入方塊3108。

【0142】 在方塊3108，微處理器100隨著分支與切換密鑰指令900/1200順著輸送一指示，顯示BTAC 2802預測其不被採用。(此外，若採用/不採用輸出2708顯示該分支與切換密鑰指令被預測爲採用，微處理器100在方塊3112隨著該分支與切換密鑰指令900/1200順著輸送一指示，顯示BTAC 2802預測其會被採用)。流程結束於3108。

【0143】 在方塊3112，提取位址產生器164以BTAC 2802於方塊3104所預測的目標位址2706更新提取位址134。流程接著進入方塊3114。

【0144】 在方塊3114，根據BTAC 2802於方塊3104所預測的密鑰暫存器檔案索引2712，密鑰切換邏輯2712以其所指示之密鑰暫存器檔案124位置更新主密鑰暫存器142內的密鑰數值。在一種實施方式中，必要狀況下，密鑰切換邏輯2712會拖延提取單元104提取指令數據106內的區塊，直至主密鑰暫存器142被更新。流程接著進入方塊3116。

【0145】 在方塊3116，提取單元104利用方塊3114所載入的新主密鑰暫存器142內容持續提取並且解密指令數據106。流程結束於方塊3116。

【0146】 現在，參閱第32圖，一流程圖圖解第27圖微處理器100的操作，其中，根據本發明技術，執行一分支與切換密鑰指令900/1200。第32圖流程在某一方面類似第10圖流程，且類似的方塊是採以同樣標號。雖然第32圖的討論是參照第10圖內容，其應用可更考慮第14圖所介紹的分支與切換密鑰指令

1200操作。第32圖流程始於方塊1002。

【0147】 在方塊1002，解碼單元108解碼一分支與切換密鑰指令900/1200，且將之代入微代碼單元132實現分支與切換密鑰指令900/1200的微代碼程序。流程接著進入方塊1006。

【0148】 在方塊1006，微代碼解出分支方向(即採用/不採用)以及目標位址。流程接著進入方塊3208。

【0149】 在方塊3208，微代碼判斷BTAC 2802是否爲該分支與切換密鑰指令900/1200提供一預測。若有提供，流程接著進入決策方塊3214；若無提供，流程接著進入第10圖的方塊1008。

【0150】 在決策方塊3214，微代碼藉由將BTAC 2802輸送出的採用/不採用指標2708以及目標位址2706與方塊1006所解出的方向以及目標位址判斷BTAC 2802所做的預測是否正確。若BTAC 2802的預測正確，則流程結束；反之，則流程來到決策方塊3216。

【0151】 在決策方塊3216，微代碼判斷此不正確的BTAC 2802預測有沒有被採用。若已被採用，流程進入方塊3222；若無，流程進入第10圖的方塊1014。

【0152】 在方塊3222，微代碼修復主密鑰暫存器142的內容，因爲BTAC 2802對分支與切換密鑰指令900/1200所做的錯誤預測被採用，導致第31圖方塊3114將錯誤的密鑰數值載入其中。在一種實施方式中，密鑰切換邏輯2712包括修復主密鑰暫存器142所需的儲存元件與邏輯。在一種實施方式中，微代碼產生一異常警示交由一異常處理器修復主密鑰暫存器142。此外，微代碼使得微處理器100分支跳躍到該分支與切換密鑰指

令900/1200之後接續的x86指令，使得微處理器100中新於該分支與切換密鑰指令900/1200的所有x86指令清空，並且使微處理器100中較分支至目標位址之微代碼新的所有微代碼清空。被清空的內容包括讀取自指令快取記憶體102、且緩衝暫存於提取單元104、解碼單元108中等待被解碼的所有指令位元組106。隨著分支至接續的指令，提取單元104開始使用主密鑰暫存器142內的該組修復後的密鑰數值自指令快取記憶體102提取並且解密指令數據106。流程結束於方塊3222。

【0153】 除了以上所述、由微處理器100實現的指令解密實施方式所帶來的安全優勢，發明人更發展出建議編碼指南，其使用可配合以上實施方式，削弱藉由分析x86指令實際使用量、對加密x86碼以統計技巧發展出的駭客攻擊。

【0154】 第一，由於駭客通常假設所提取的16位元組的指令數據106全數爲x86指令，因此，相對於程式執行流程，編碼時應當在16位元組區塊之間加入「洞(holes)」。也就是說，其編碼應當以多個指令跳躍一些指令位元組，以未加密的位元組產生多個「洞」，其中可填入適當的數值，以增加純文字位元組的熵值(entropy)。此外，倘若能更提升純文字位元組的熵值，其編碼可盡可能採用即時數據值。此外，所述即時數據值可作爲假線索，指向錯誤的指令操作碼位址。

【0155】 第二，所述編碼可包括特別的NOP指令，其中包括“不理會”欄位，填有適當數值以增加上述熵值。例如，x86指令0x0F0D05xxxxxxxx屬於7位元組的NOP，其中最後四個位元組可爲任意值。此外，NOP指令的操作碼型式以及其「不理

會」位元組的數量更可有其他變化。

【0156】 第三，許多x86指令具有與其他x86指令相同的基本功能。關於等效功能的指令，其編碼可捨棄重複使用同樣的指令，改採用多重型式並且/或採用使純文字熵值提升的型式。例如，指令0xC10107以及指令0xC10025作的是同樣的事情。甚至，某些等效指令是以不同長度的版本呈現，例如，0xEB22以及0xE90022；因此，編碼時可採用多種長度但相同效果的指令。

【0157】 第四，x86架構允許使用冗餘且無意義的操作碼字首(opcode prefixes)，因此，編碼時可小心應用之，以更增加上述熵值。例如，指令0x40 以及0x2627646567F2F340作的是完全一樣的事情。因爲其中僅有8個安全的x86字首，他們需被小心地安插在編碼中，以避免過度頻繁地出現。

【0158】 雖然已經列舉多種實施例以密鑰擴展器對主密鑰暫存器數值中的一對數值進行旋轉以及加/減運算，尚有其他實施方式可考慮使用，其中，密鑰擴展器可對多於兩個的主密鑰暫存器數值進行運算，此外，所進行的運算可不同於旋轉以及加/減運算。此外，第6圖揭露的密鑰切換指令600以及第9圖揭露的分支與切換密鑰指令900更可有其他實施方式，例如，將新的密鑰數值由安全存儲區122載入主密鑰暫存器142而非由密鑰暫存器檔案124載入，並且，第15圖所介紹的分支與切換密鑰指令1500的其他實施方式是以索引欄位2104儲存安全存儲區122的位址。此外，雖然已列舉多種實施例調整BTAC 2702儲存KRF索引配合分支與切換密鑰指令900/1200使用，尚

有其他實施方式是調整BTAC 2702儲存安全存儲區位址，以配合分支與切換密鑰指令1500使用。

【0159】 以上列舉的本發明諸多實施方式僅是作爲說明例使用，並非意圖限制發明範圍。相關電腦技術領域人員可在不偏離本發明範圍的前提下作出形式以及細節的諸多變形。例如，可以軟體方式實現所述如函式、製作、模組化、模擬、說明、以及/或測試此篇所討論之設備與方法的方式。實現方式包括一般程式語言(例如，C、C++)、硬體描述語言包括Verilog HDL、VHDL…等、或其他可用的程式工具。所述軟體可載於任何已知的計算機可讀媒體，例如，磁帶、半導體、磁碟、或光碟(例如，CD-ROM、DVD-ROM等)、網路、有線傳輸、無線或其他通訊媒體。所述設備與方法的實施方式可包含於半導體知識產權核心，例如一微處理器核心(例如以HDL實現)，並可轉成硬體以積體電路實現。此外，所述之設備與方法可由軟、硬體結合方式實現。因此，本發明範圍不應限定於所述任何實施方式，應當是以下列請求項以及其等效技術界定之。特別是，本發明技術可以一般用途計算機所採用的微處理器實現。值得注意的是，本技術領域人員可能不偏離請求項所定義之發明範圍、以所揭露之概念以及特殊實施例爲基礎、設計或修正提出其他架構產生與本發明相同的效果。

【符號說明】

【0160】

100~微處理器；

102~指令快取記憶體；

104~提取單元；

106~指令數據(可爲加密)；

108~解碼單元；

112~執行單元；

114~引出單元；

118~通用暫存器；

122~安全存儲區；

124~密鑰暫存器檔案；

128~標誌暫存器；

132~微代碼單元；

134~提取位址；

142~主密鑰暫存器；

144~控制暫存器；

148~E位元；

152~密鑰擴展器；

154~多工器；

156~互斥邏輯；

162~純文字指令數據；

164~提取位址產生器；

172~兩組密鑰；

174~解密密鑰；

176~多位元的二進位零值；

178~多工器154的輸出；

212~多工器A；

214~多工器B；

216~旋轉器；

218~加法/減法器；

234~第一密鑰；

236~第二密鑰；

238~旋轉器的輸出；

302-316~步驟方塊；

402~E位元欄位；

408~多個位元的標準x86標誌；

500~密鑰載入指令；

502~操作碼；

504~密鑰暫存器檔案目標位址；

506~安全存儲區來源位址；

600~密鑰切換指令；

602~操作碼；

604~密鑰暫存器檔案索引；

702-708~方塊步驟；

800~記憶體用量；

900~分支與切換密鑰指令；

902~操作碼；

904~密鑰暫存器檔案索引；

906~分支資訊；

1002-1018~步驟方塊；

1102-1106~步驟方塊；

1200~分支與切換密鑰指令；

1202~操作碼；

1300~塊位址範圍表：

1302~位址範圍；

1304~密鑰暫存器檔案索引；

1402-1418~步驟方塊；

1500~分支與切換密鑰指令；

1502~操作碼；

1600~塊位址範圍表：

1604~安全存儲區位址；

1714~步驟方塊；

1802-1806~步驟方塊；

1902-1944~步驟方塊；

2002-2008~步驟方塊；

2104~索引；

2202-2216~步驟方塊；

2302-2316~步驟方塊；

2402~淘汰位元；

2506~步驟方塊；

2607、2609~步驟方塊；

2702~分支目標位址快取記憶體(BTAC)；

2706~目標位址；

2708~採用/不採用指標；

2712~密鑰切換邏輯；

2714~型式指標；

2716~密鑰暫存器檔案索引；

2802~BTAC陣列；

2808~BTAC單元；

2902~有效位元；

2904~標記欄位；

2906~目標位址；

2908~採用/不採用欄位；

2912~密鑰暫存器檔案索引；

2914~型式欄位；

3002-3004~步驟方塊；

3102-3116~步驟方塊；

3208-3222~步驟方塊；

ZEROES~多位元的二進位零值。

申請專利範圍

1. 一種微處理器，包括：

一提取單元，採用第一解密密鑰數據提取並且解密一分支與切換密鑰指令；以及

微代碼，用於：

在該分支與切換密鑰指令的方向不被採用的狀況下，使該提取單元以上述第一解密密鑰數據提取並且解密該分支與切換密鑰指令之後的接續指令；以及

在該分支與切換密鑰指令的方向被採用的狀況下，使該提取單元以不同於該第一解密密鑰數據的第二解密密鑰數據提取並且解密該分支與切換密鑰指令的一目標指令。

2. 如申請專利範圍第1項所述之微處理器，其中，若該分支與切換密鑰指令的方向被採用，上述微代碼更在致使該提取單元以上述第二解密密鑰數據解密該目標指令之前，更新該提取單元，而使該提取單元使用上述第二解密密鑰數據而非上述第一解密密鑰數據。

3. 如申請專利範圍第1項所述之微處理器，其中，若該分支與切換密鑰指令的方向被採用，該分支與切換密鑰指令標示該微處理器中上述第二解密密鑰數據的儲存位置，以供該提取單元載入。

4. 如申請專利範圍第1項所述之微處理器，其中，若該分支與切換密鑰指令的方向被採用，上述微代碼更基於該目標指令的記憶體位址判斷上述第二解密密鑰數據於該微處理器內的一儲存位置，以供該提取單元載入。

5. 如申請專利範圍第4項所述之微處理器，其中，爲了根據該目標指令的上述記憶體位址判斷該微處理器中上述第二解密密鑰數據的該儲存位置，上述微代碼以上述記憶體位址查詢該微處理器內一表格，該表格記載有記憶體位址範圍至該微處理器內儲存位置的一映射。

6. 如申請專利範圍第5項所述之微處理器，其中，該微處理器的儲存位置是設計在該微處理器的一暫存器中或一隨機存取記憶體中。

7. 如申請專利範圍第1項所述之微處理器，其中該微處理器以原子型式執行該分支與切換密鑰指令。

8. 如申請專利範圍第1項所述之微處理器，其中，該提取單元更用於：

於解密該分支與切換密鑰指令前，基於上述第一解密密鑰數據以及用於提取該分支與切換密鑰指令的一提取位址之部分內容，生成一解密密鑰；

其中，爲了採用上述第一解密密鑰數據解密該分支與切換密鑰指令，該提取單元以所生成的該解密密鑰對該分支與切換密鑰指令實行布林互斥運算。

9. 一種操作方法，以一微處理器處理一加密程式，該操作方法包括：

採用第一解密密鑰數據提取並且解密一分支與切換密鑰指令；

若該分支與切換密鑰指令的方向不被採用，基於上述第一解密密鑰數據提取並且解密該分支與切換密鑰指令之後的

接續指令；以及

若該分支與切換密鑰指令的方向被採用，基於不同於上述第一解密密鑰數據的第二解密密鑰數據提取並且解密該分支與切換密鑰指令的一目標指令。

10.如申請專利範圍第9項所述之操作方法，其中，上述基於上述第一解密密鑰數據解密該分支與切換密鑰指令與接續指令的操作是由該微處理器中使用上述第一解密密鑰數據的一提取單元執行，其中，若該分支與切換密鑰指令的方向被採用，則該操作方法更包括：

在基於上述第二解密密鑰數據解密該目標指令之前，更新該提取單元使用上述第二解密密鑰數據而非上述第一解密密鑰數據。

11.如申請專利範圍第9項所述之操作方法，其中，若該分支與切換密鑰指令的方向被採用，該分支與切換密鑰指令標示該微處理器中上述第二解密密鑰數據的一儲存位置，以供一提取單元載入。

12.如申請專利範圍第9項所述之操作方法，若該分支與切換密鑰指令的方向被採用，更包括：

根據該目標指令的記憶體位址判斷該微處理器中上述第二解密密鑰數據的一儲存位置，以供一提取單元載入。

13.如申請專利範圍第12項所述之操作方法，其中，上述根據該目標指令的上述記憶體位址判斷該微處理器內上述第二解密密鑰數據的該儲存位置之步驟包括基於上述記憶體位址查詢該微處理器內一表格，該表格記錄記憶體位址範圍

至該微處理器儲存位置的一映射。

14.如申請專利範圍第13項所述之操作方法，其中，該微處理器的儲存位置位於該微處理器的一暫存器中或一隨機存取記憶體中。

15.如申請專利範圍第9項所述之操作方法，其中，該分支與切換密鑰指令的執行是由該微處理器以原子型式操作。

16.如申請專利範圍第9項所述之操作方法，更包括：

在上述解密該分支與切換密鑰指令的步驟之前，根據上述第一解密密鑰數據以及用於提取該分支與切換密鑰指令的一提取位址的部分內容生成一解密密鑰；

其中，上述基於上述第一解密密鑰數據解密該分支與切換密鑰指令的步驟包括以所生成的該解密密鑰對該分支與切換密鑰指令實行布林互斥運算。

17.一種加密方法，用於加密一程式，以供用於解密與執行加密程式的一微處理器日後執行，該加密方法包括：

接收關於一非加密程式的一目的檔，該非加密程式包括傳統分支指令，上述傳統分支指令的目標位址在該微處理器執行該程式之前被判定；

分析該程式以獲得塊資訊，上述塊資訊將該程式劃分成一序列多個塊，其中，各個上述塊包括一序列多個指令，其中，上述塊資訊更包括上述各個塊的加密密鑰數據，其中，各塊的加密密鑰數據並不相同；

將上述傳統分支指令中目標位址不與本身位於同一塊者各自以一分支與切換密鑰指令取代；以及

基於上述塊資訊加密該程式。

18.如申請專利範圍第17項所述之加密方法，其中，上述分支與切換密鑰指令各自標示該微處理器內一儲存空間，以儲存上述分支與切換密鑰指令之目標位址所在該塊的加密密鑰數據。

19.如申請專利範圍第17項所述之加密方法，其中，上述根據塊資訊加密該程式的步驟包括：

針對各塊內各個區塊的指令數據，基於所屬塊之加密密鑰數據以及所屬區塊一記憶體位址的部份內容生成一加秘密鑰；以及

以所生成的該加秘密鑰對所對應該區塊進行布林互斥運算。

20.一種加密方法，用於加密一程式，以供用於解密與執行加密程式的一微處理器日後執行，該加密方法包括：

接收一非加密程式的一目的檔，該非加密程式包括傳統分支指令，上述傳統分支指令的目標位址僅能在該微處理器執行該程式時判定；

分析該程式以獲得塊資訊，上述塊資訊將該程式劃分成一序列多個塊，其中，各個上述塊包括一序列多個指令，其中，上述塊資訊更包括上述各個塊的加密密鑰數據，其中，各塊的加密密鑰數據並不相同；

將上述傳統分支指令各自以一分支與切換密鑰指令取代；以及

基於上述塊資訊，加密該程式。

21.如申請專利範圍第20項所述之加密方法，更包括：

將上述塊資訊納入該目的檔中，以於該微處理器執行該程式前載入該微處理器。

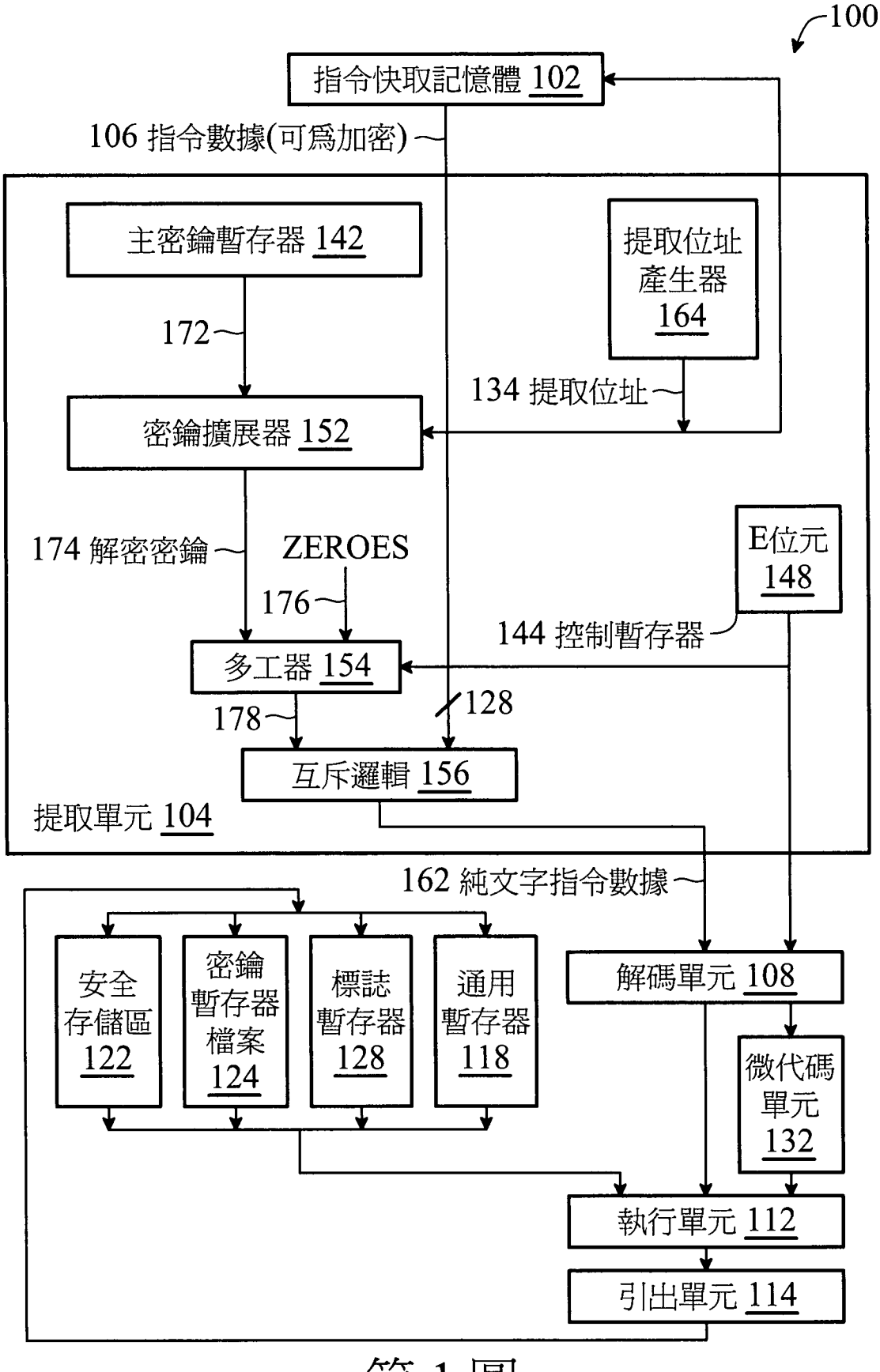
22.如申請專利範圍第21項所述之加密方法，其中，載於該目的檔中、於執行該程式前載入該微處理器的上述塊資訊爲各個上述塊在該微處理器中標示一儲存空間，以儲存關於各個上述塊的上述加密密鑰數據。

23.如申請專利範圍第20項所述之加密方法，其中，上述基於塊資訊加密該程式的步驟包括：

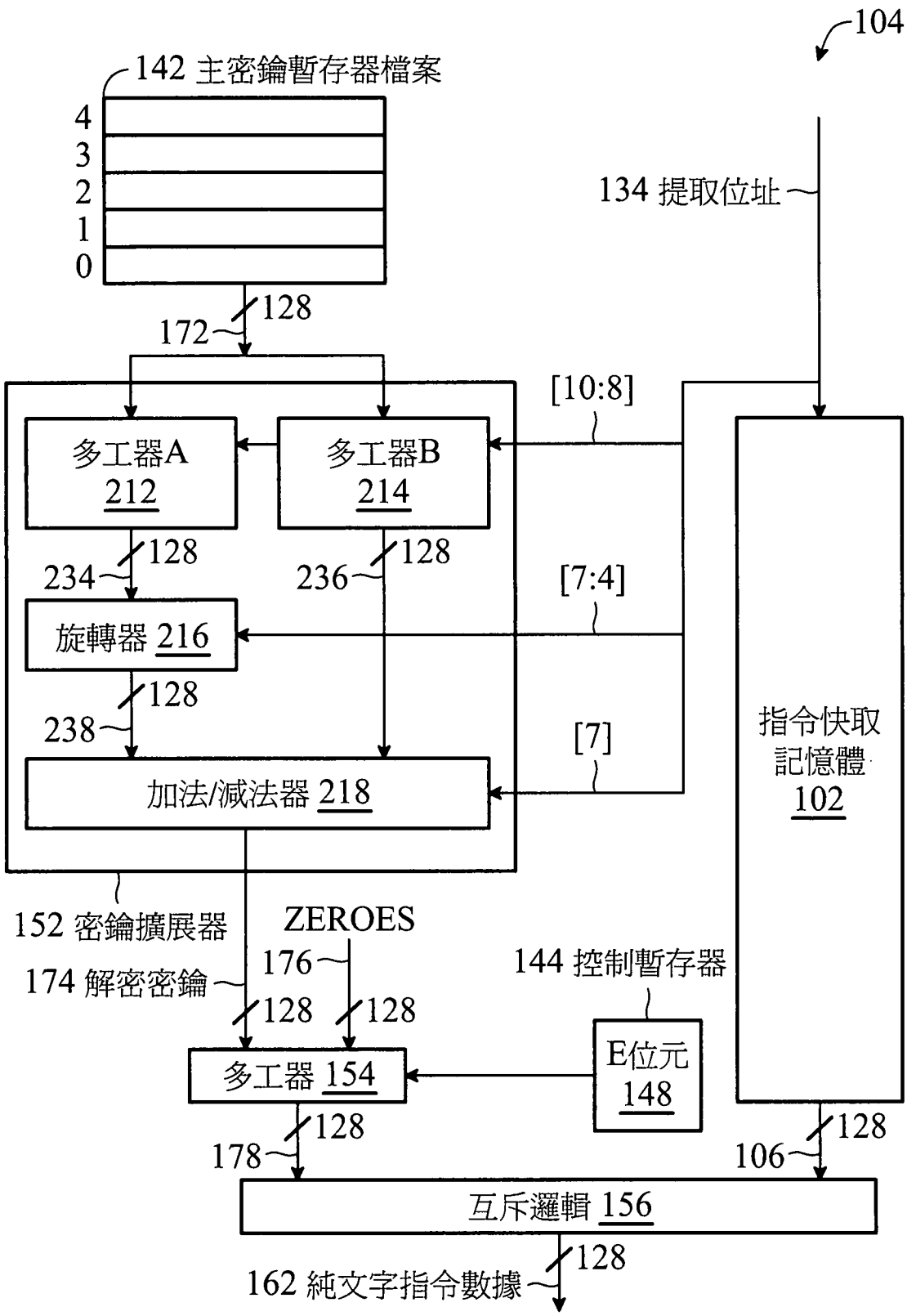
針對各個上述塊之各個區塊的指令數據，基於所屬塊之加密密鑰數據以及所屬區塊的一記憶體位址的部分內容生成一加密密鑰；以及

以所生成的上述加密密鑰，對所對應的區塊進行布林互斥運算。

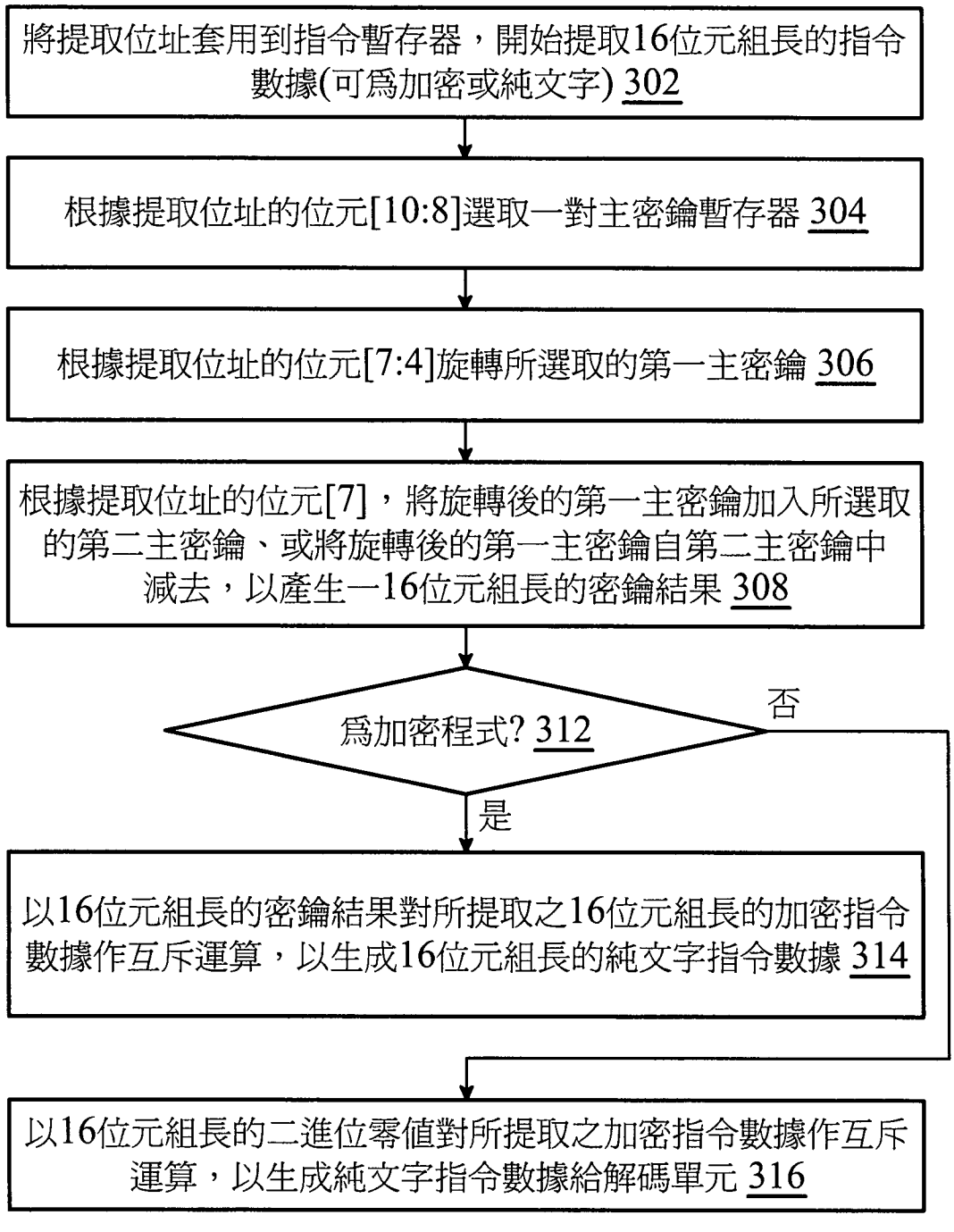
圖式


100
指令快取記憶體 102
106 指令數據(可爲加密)
主密鑰暫存器 142
提取位址產生器 164
172
134 提取位址
密鑰擴展器 152
174 解密密鑰
ZEROES
176
E位元 148
144 控制暫存器
多工器 154
178
128
互斥邏輯 156
提取單元 104
162 純文字指令數據
安全存儲區 122
密鑰暫存器檔案 124
標誌暫存器 128
通用暫存器 118
解碼單元 108
微代碼單元 132
執行單元 112
引出單元 114

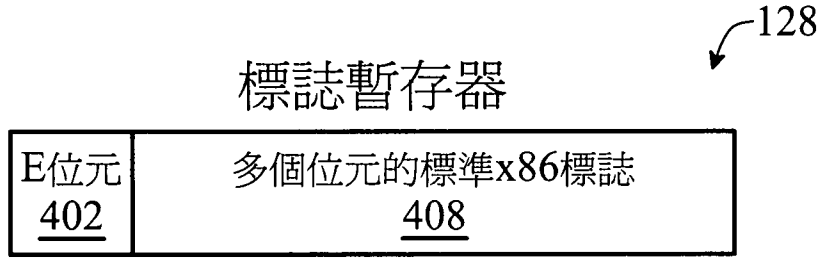
第 1 圖


104
142 主密鑰暫存器檔案
4
3
2
1
0
134 提取位址
128
172
[10:8]
多工器A
212
多工器B
214
128
128
234
236
[7:4]
旋轉器 216
128
238
[7]
加法/減法器 218
指令快取
記憶體
102
152 密鑰擴展器
ZEROES
176
174 解密密鑰
144 控制暫存器
128
128
多工器 154
E位元
148
128
128
178
106
互斥邏輯 156
128
162 純文字指令數據

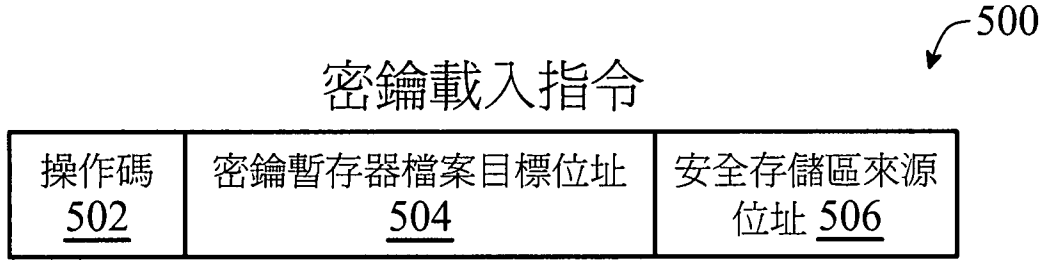
第 2 圖


將提取位址套用到指令暫存器，開始提取16位元組長的指令數據(可爲加密或純文字) 302
根據提取位址的位元[10:8]選取一對主密鑰暫存器 304
根據提取位址的位元[7:4]旋轉所選取的第一主密鑰 306
根據提取位址的位元[7]，將旋轉後的第一主密鑰加入所選取的第二主密鑰、或將旋轉後的第一主密鑰自第二主密鑰中減去，以產生一16位元組長的密鑰結果 308
爲加密程式? 312
否
是
以16位元組長的密鑰結果對所提取之16位元組長的加密指令數據作互斥運算，以生成16位元組長的純文字指令數據 314
以16位元組長的二進位零值對所提取之加密指令數據作互斥運算，以生成純文字指令數據給解碼單元 316

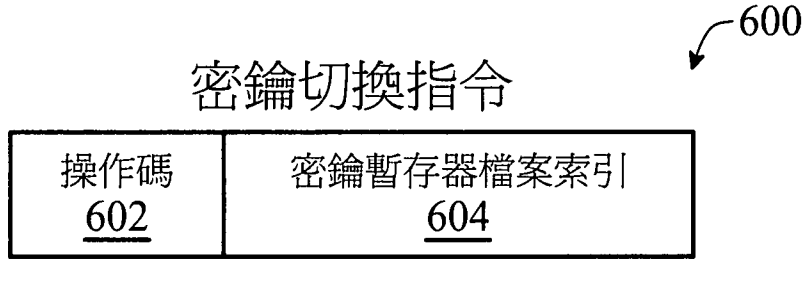
第 3 圖


128
標誌暫存器
E位元
402
多個位元的標準x86標誌
408

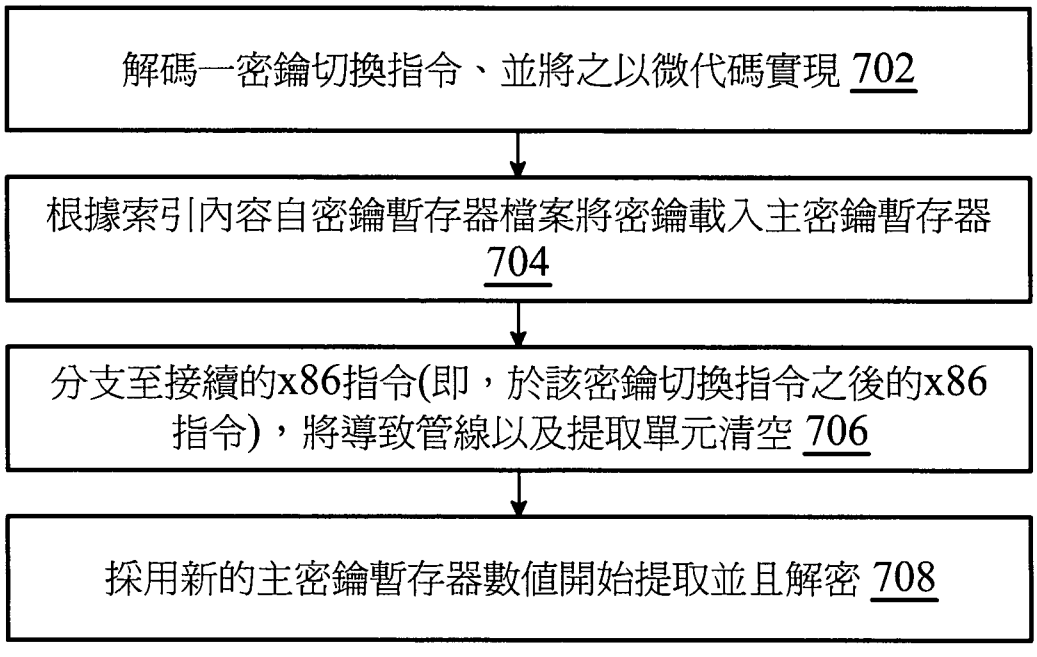
第 4 圖


500
密鑰載入指令
操作碼
502
密鑰暫存器檔案目標位址
504
安全存儲區來源
位址 506

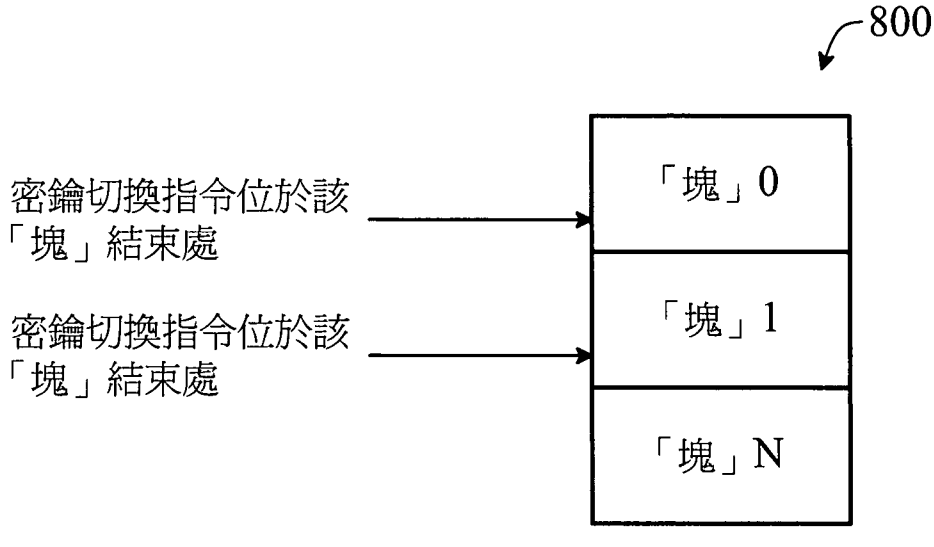
第 5 圖


600
密鑰切換指令
操作碼
602
密鑰暫存器檔案索引
604

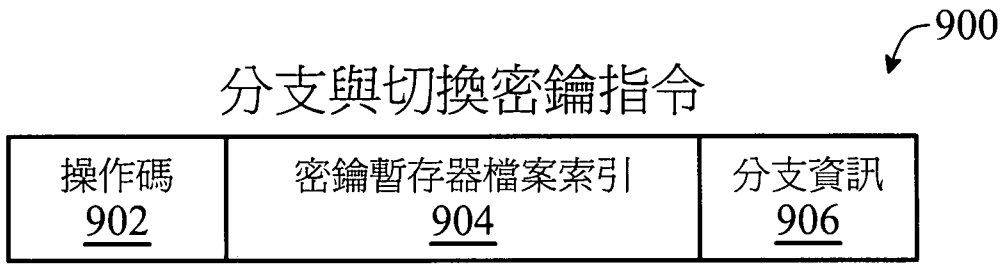
第 6 圖


解碼一密鑰切換指令、並將之以微代碼實現 702
根據索引內容自密鑰暫存器檔案將密鑰載入主密鑰暫存器 704
分支至接續的x86指令(即，於該密鑰切換指令之後的x86指令)，將導致管線以及提取單元清空 706
採用新的主密鑰暫存器數值開始提取並且解密 708

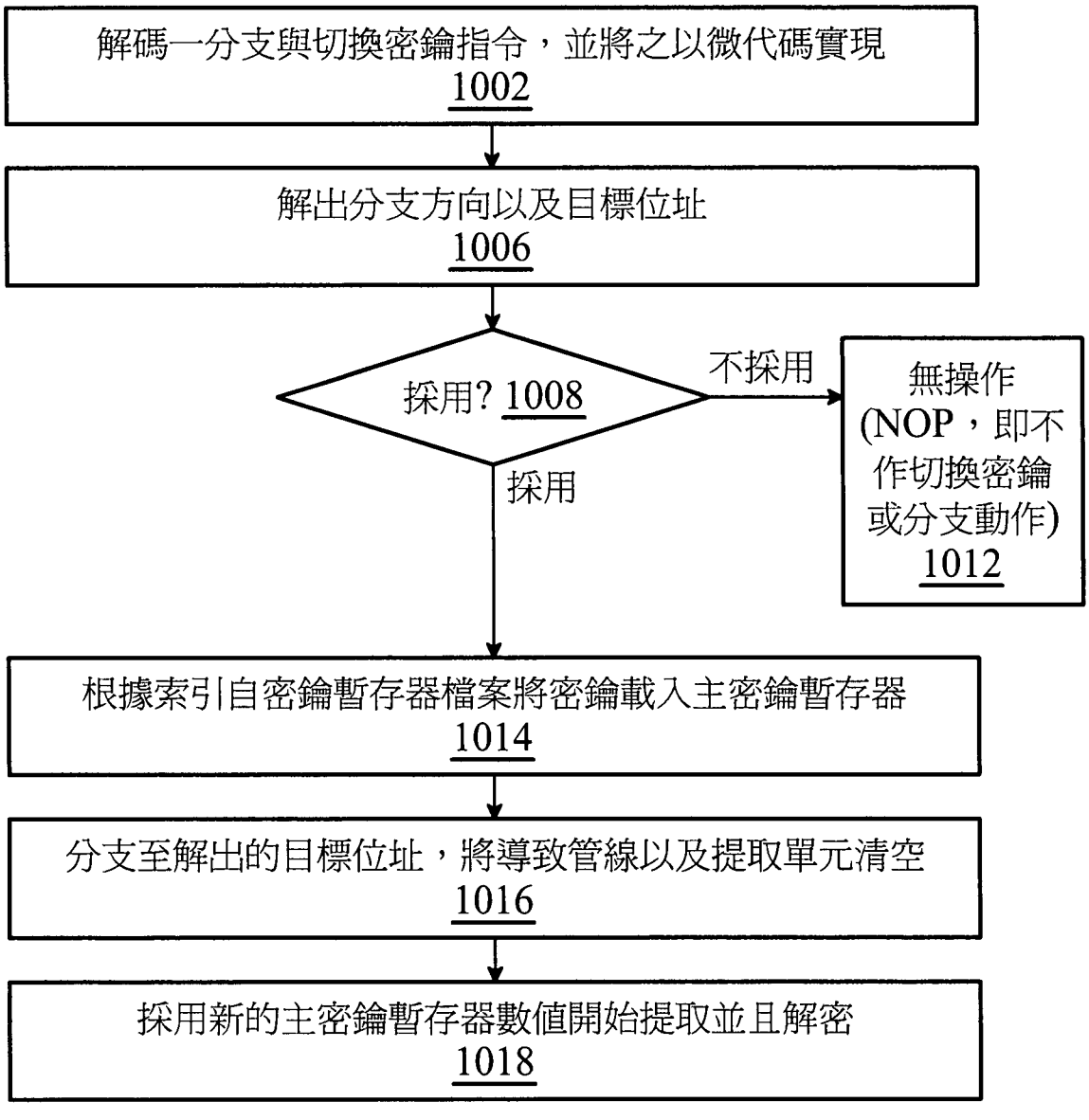
第 7 圖


800
密鑰切換指令位於該「塊」結束處
密鑰切換指令位於該「塊」結束處
「塊」0
「塊」1
「塊」N

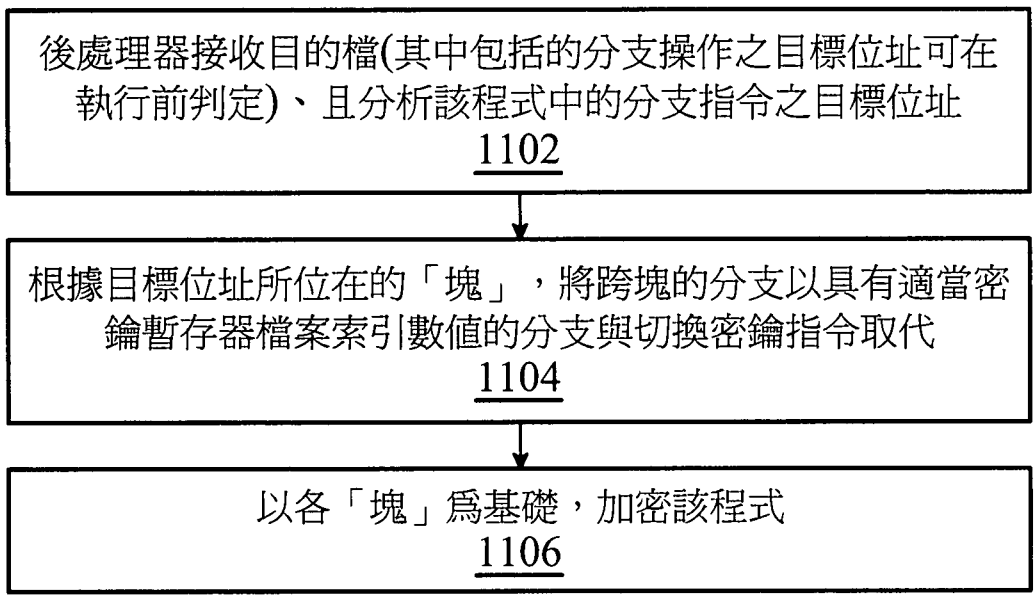
第 8 圖



第 9 圖


解碼一分支與切換密鑰指令，並將之以微代碼實現
1002
解出分支方向以及目標位址
1006
採用? 1008
不採用
無操作
(NOP，即不
作切換密鑰
或分支動作)
1012
採用
根據索引自密鑰暫存器檔案將密鑰載入主密鑰暫存器
1014
分支至解出的目標位址，將導致管線以及提取單元清空
1016
採用新的主密鑰暫存器數值開始提取並且解密
1018

第 10 圖


後處理器接收目的檔(其中包括的分支操作之目標位址可在執行前判定)、且分析該程式中的分支指令之目標位址
1102
根據目標位址所位在的「塊」，將跨塊的分支以具有適當密鑰暫存器檔案索引數值的分支與切換密鑰指令取代
1104
以各「塊」爲基礎，加密該程式
1106

第 11 圖

1200

分支與切換密鑰指令(另外一種實施方式)

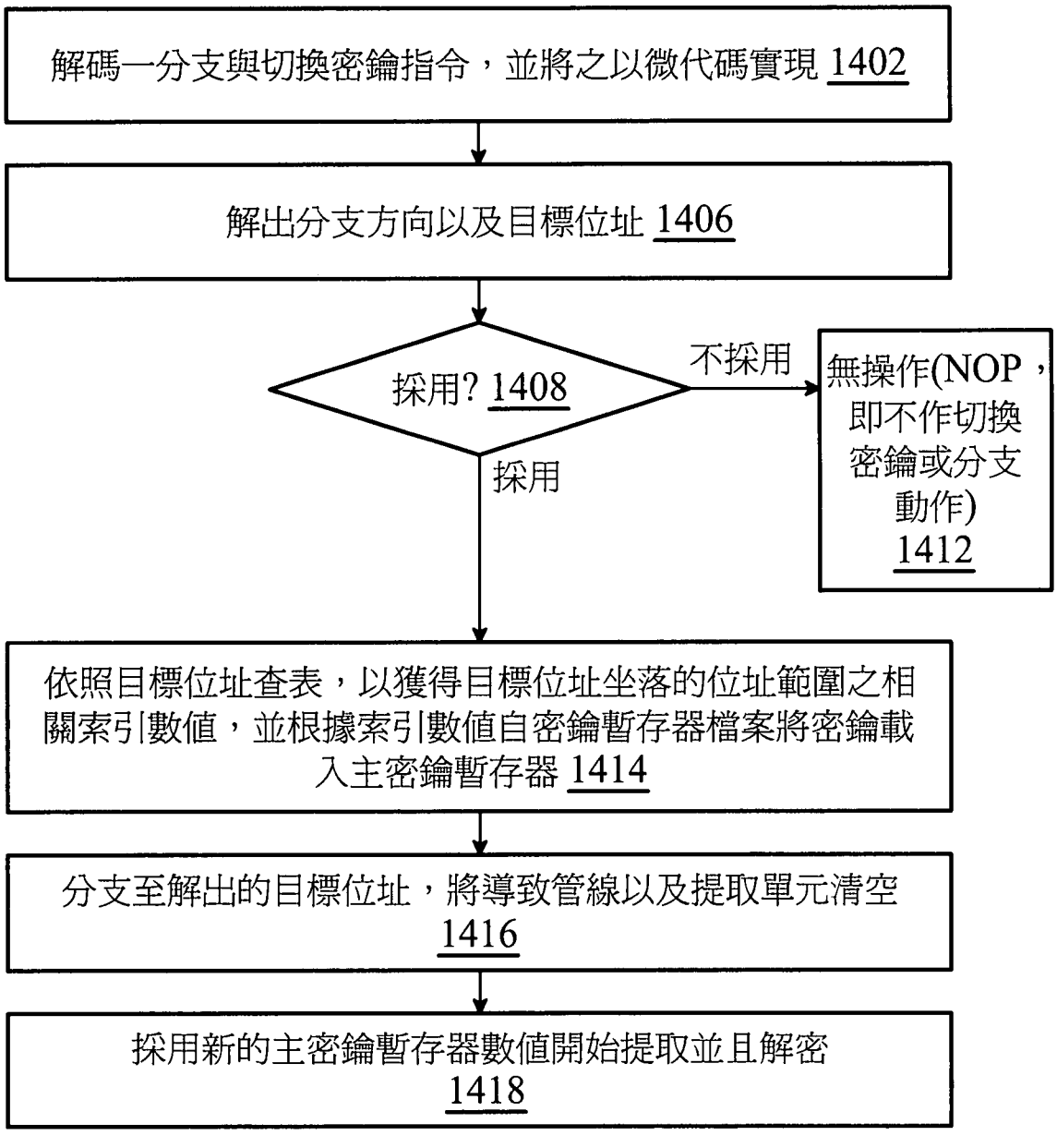
操作碼 1202	分支資訊 906

第 12 圖

1300

0	位址範圍 1302	密鑰暫存器檔案索引 1304
1	位址範圍 1302	密鑰暫存器檔案索引 1304
2	位址範圍 1302	密鑰暫存器檔案索引 1304
...	位址範圍 1302	密鑰暫存器檔案索引 1304
N-1	位址範圍 1302	密鑰暫存器檔案索引 1304

第 13 圖


解碼一分支與切換密鑰指令，並將之以微代碼實現 1402
解出分支方向以及目標位址 1406
採用? 1408
不採用
無操作(NOP，即不作切換密鑰或分支動作) 1412
採用
依照目標位址查表，以獲得目標位址坐落的位址範圍之相關索引數值，並根據索引數值自密鑰暫存器檔案將密鑰載入主密鑰暫存器 1414
分支至解出的目標位址，將導致管線以及提取單元清空 1416
採用新的主密鑰暫存器數值開始提取並且解密 1418

第 14 圖

1500

分支與切換密鑰指令(另外一種實施方式)

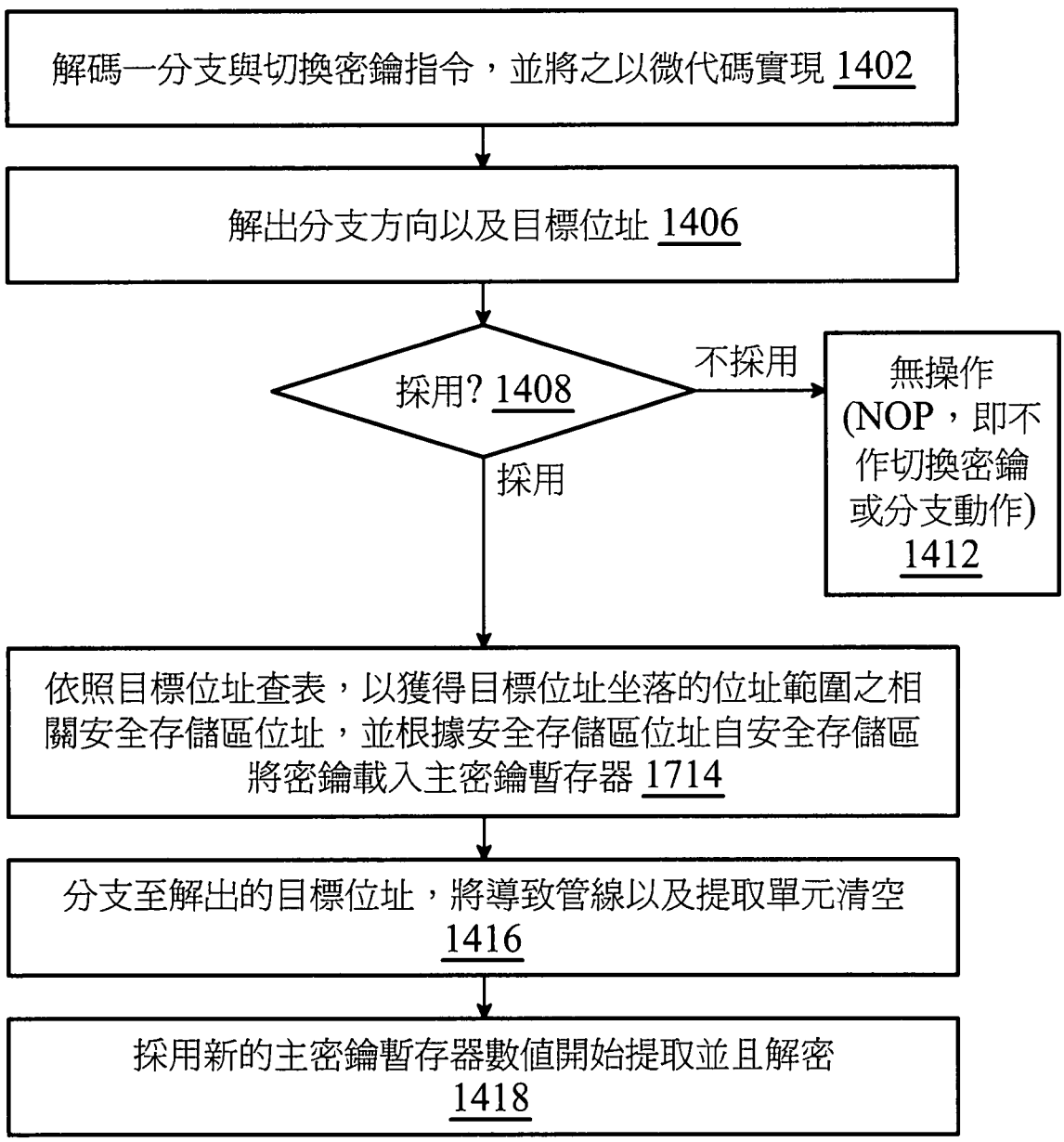
操作碼 1502	分支資訊 906

第 15 圖

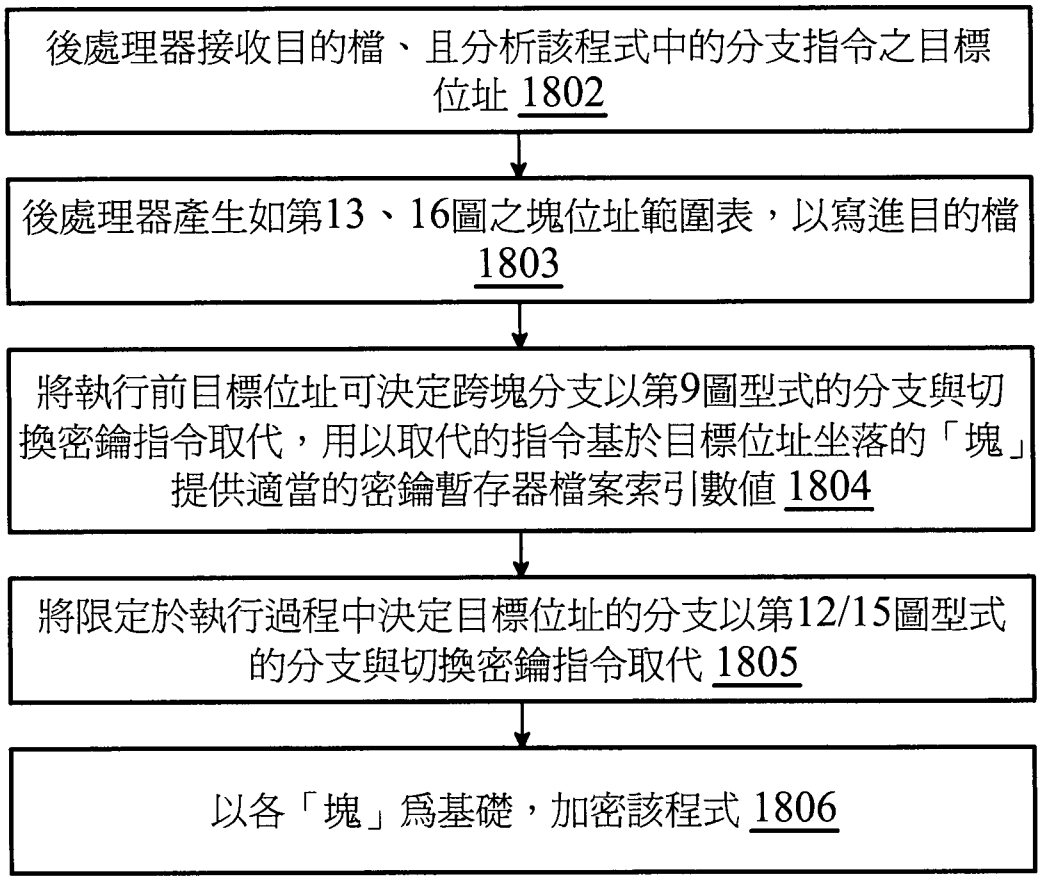
1600

0	位址範圍 1302	安全存儲區位址 1604
1	位址範圍 1302	安全存儲區位址 1604
2	位址範圍 1302	安全存儲區位址 1604
...	位址範圍 1302	安全存儲區位址 1604
N-1	位址範圍 1302	安全存儲區位址 1604

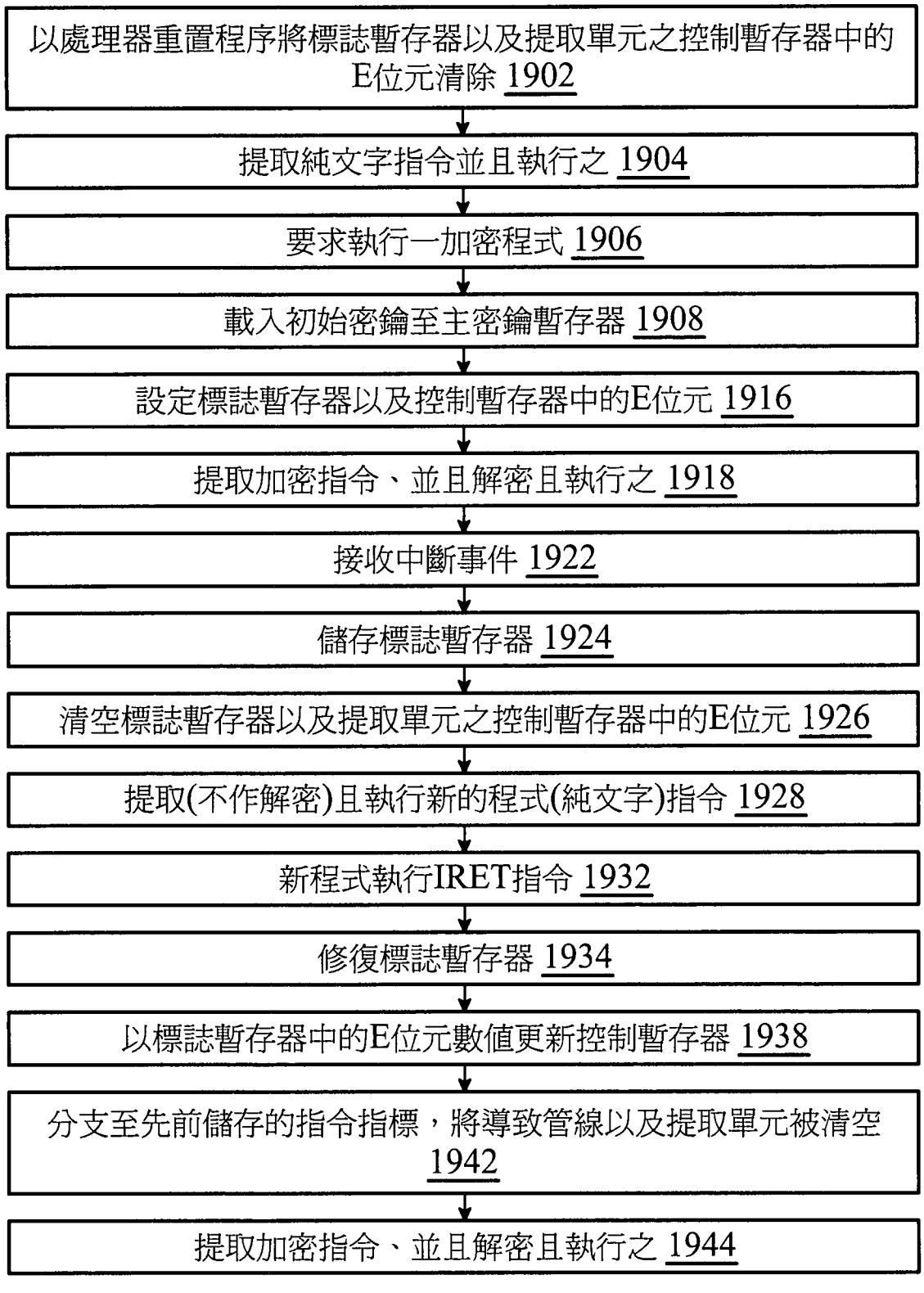
第 16 圖


解碼一分支與切換密鑰指令，並將之以微代碼實現 1402
解出分支方向以及目標位址 1406
採用? 1408
不採用
無操作 (NOP，即不作切換密鑰或分支動作) 1412
採用
依照目標位址查表，以獲得目標位址坐落的位址範圍之相關安全存儲區位址，並根據安全存儲區位址自安全存儲區將密鑰載入主密鑰暫存器 1714
分支至解出的目標位址，將導致管線以及提取單元清空 1416
採用新的主密鑰暫存器數值開始提取並且解密 1418

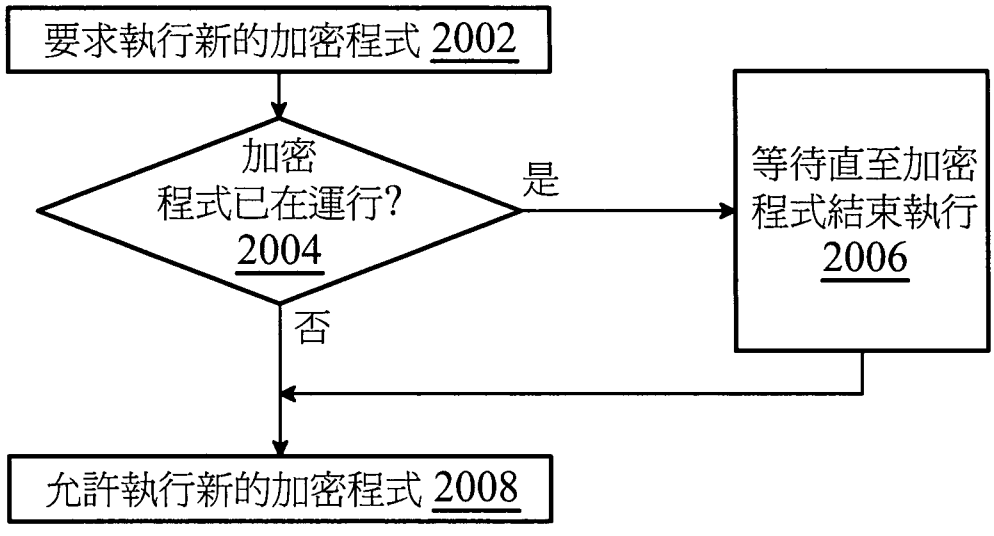
第 17 圖


後處理器接收目的檔、且分析該程式中的分支指令之目標位址 1802
後處理器產生如第13、16圖之塊位址範圍表，以寫進目的檔 1803
將執行前目標位址可決定跨塊分支以第9圖型式的分支與切換密鑰指令取代，用以取代的指令基於目標位址坐落的「塊」提供適當的密鑰暫存器檔案索引數值 1804
將限定於執行過程中決定目標位址的分支以第12/15圖型式的分支與切換密鑰指令取代 1805
以各「塊」爲基礎，加密該程式 1806

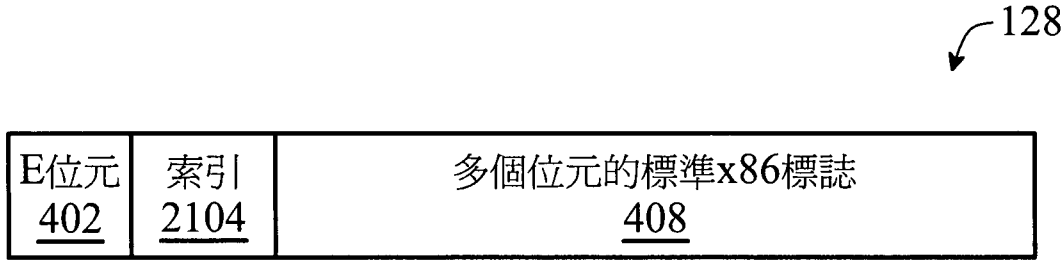
第 18 圖


以處理器重置程序將標誌暫存器以及提取單元之控制暫存器中的E位元清除 1902
提取純文字指令並且執行之 1904
要求執行一加密程式 1906
載入初始密鑰至主密鑰暫存器 1908
設定標誌暫存器以及控制暫存器中的E位元 1916
提取加密指令、並且解密且執行之 1918
接收中斷事件 1922
儲存標誌暫存器 1924
清空標誌暫存器以及提取單元之控制暫存器中的E位元 1926
提取(不作解密)且執行新的程式(純文字)指令 1928
新程式執行IRET指令 1932
修復標誌暫存器 1934
以標誌暫存器中的E位元數值更新控制暫存器 1938
分支至先前儲存的指令指標，將導致管線以及提取單元被清空 1942
提取加密指令、並且解密且執行之 1944

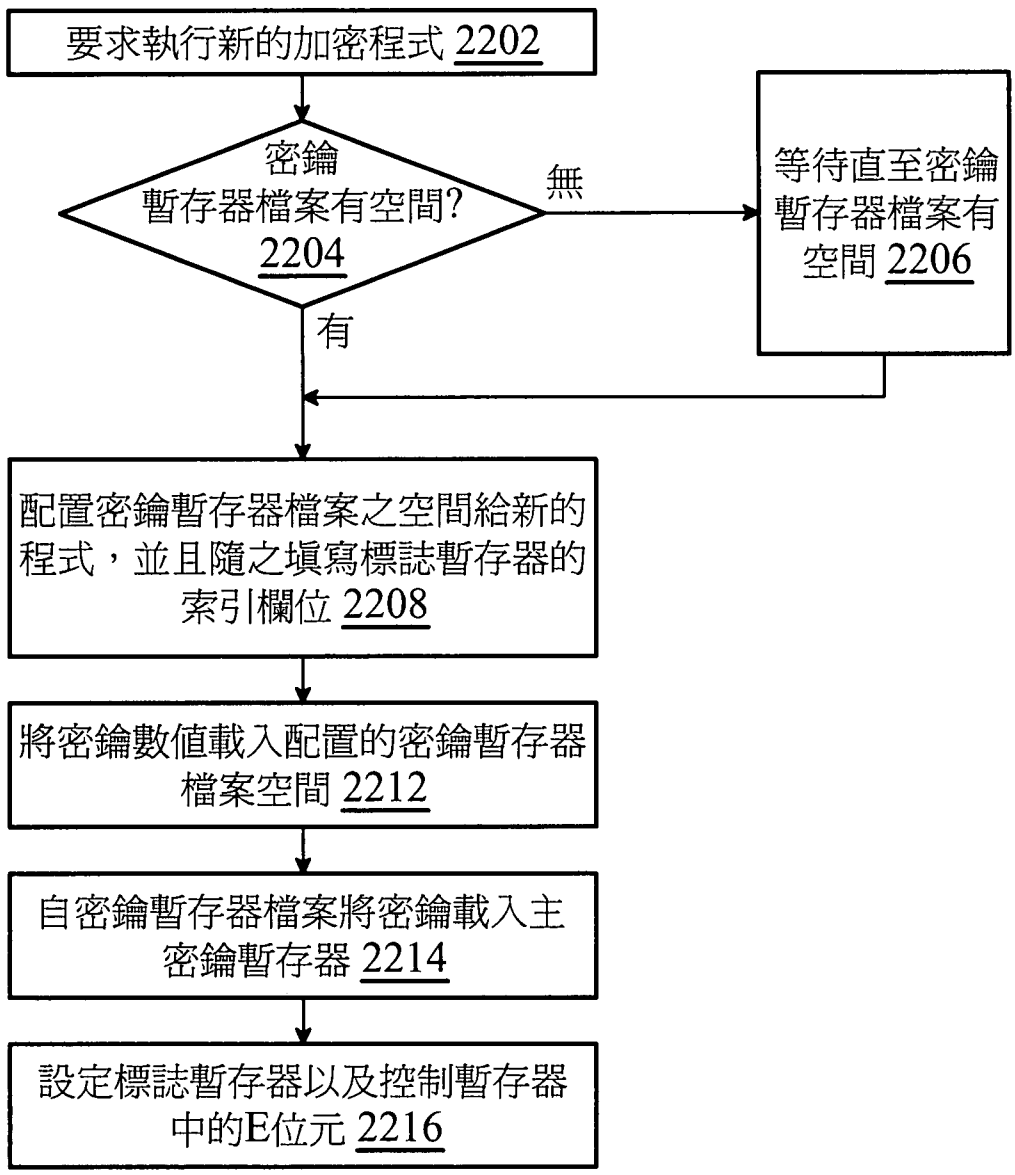
第 19 圖


要求執行新的加密程式 2002
加密
程式已在運行?
2004
是
等待直至加密
程式結束執行
2006
否
允許執行新的加密程式 2008

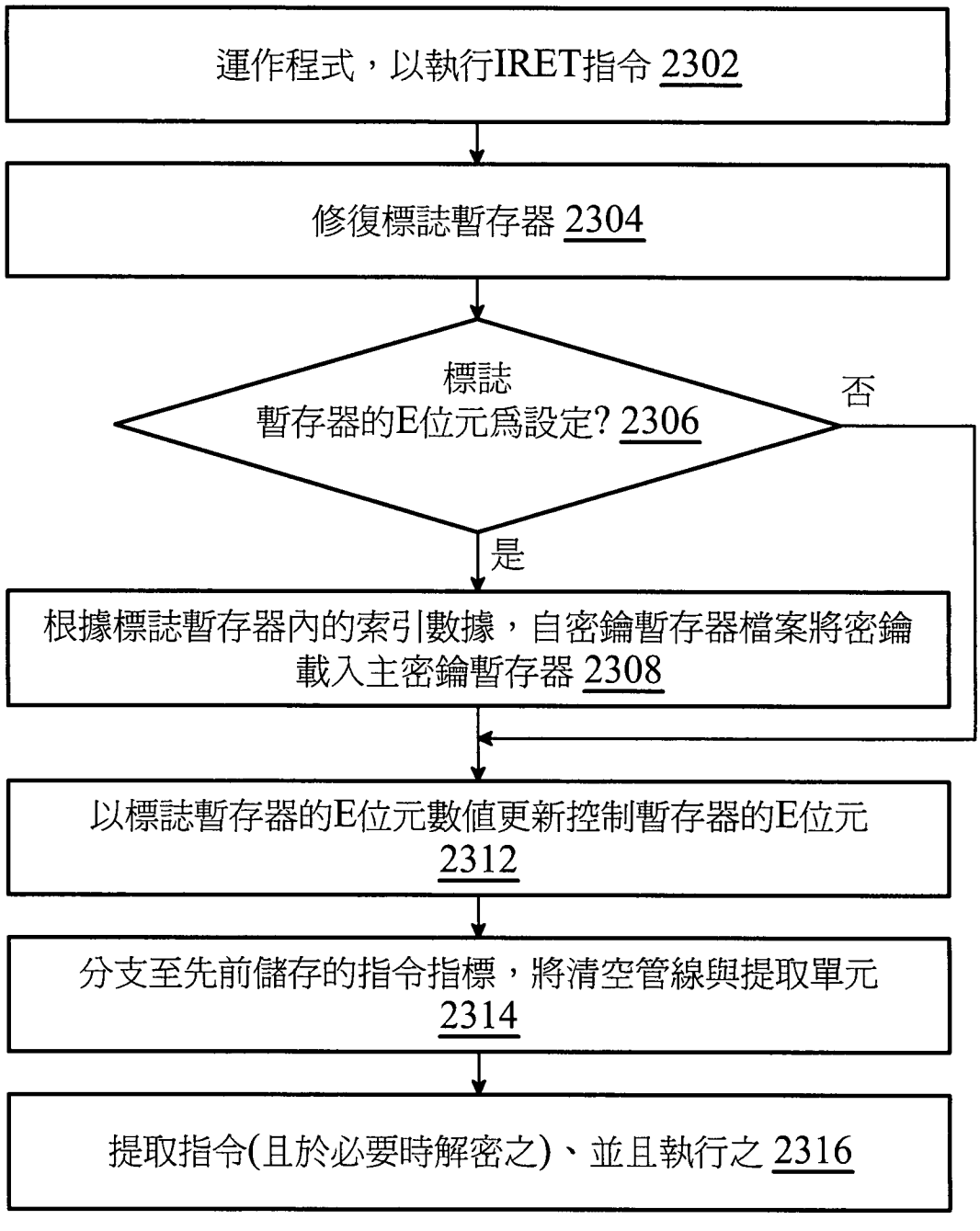
第 20 圖


128
E位元
402
索引
2104
多個位元的標準x86標誌
408

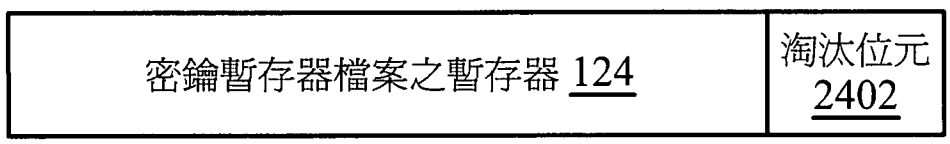
第 21 圖


要求執行新的加密程式 2202
密鑰暫存器檔案有空間? 2204
無
等待直至密鑰暫存器檔案有空間 2206
有
配置密鑰暫存器檔案之空間給新的程式，並且隨之填寫標誌暫存器的索引欄位 2208
將密鑰數值載入配置的密鑰暫存器檔案空間 2212
自密鑰暫存器檔案將密鑰載入主密鑰暫存器 2214
設定標誌暫存器以及控制暫存器中的E位元 2216

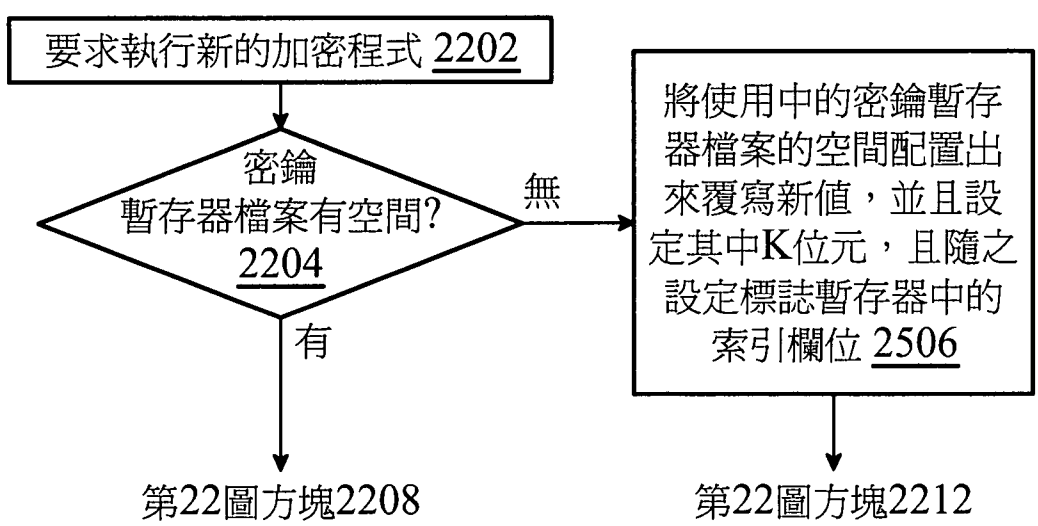
第 22 圖


運作程式，以執行IRET指令 2302
修復標誌暫存器 2304
標誌
暫存器的E位元爲設定? 2306
否
是
根據標誌暫存器內的索引數據，自密鑰暫存器檔案將密鑰載入主密鑰暫存器 2308
以標誌暫存器的E位元數值更新控制暫存器的E位元 2312
分支至先前儲存的指令指標，將清空管線與提取單元 2314
提取指令(且於必要時解密之)、並且執行之 2316

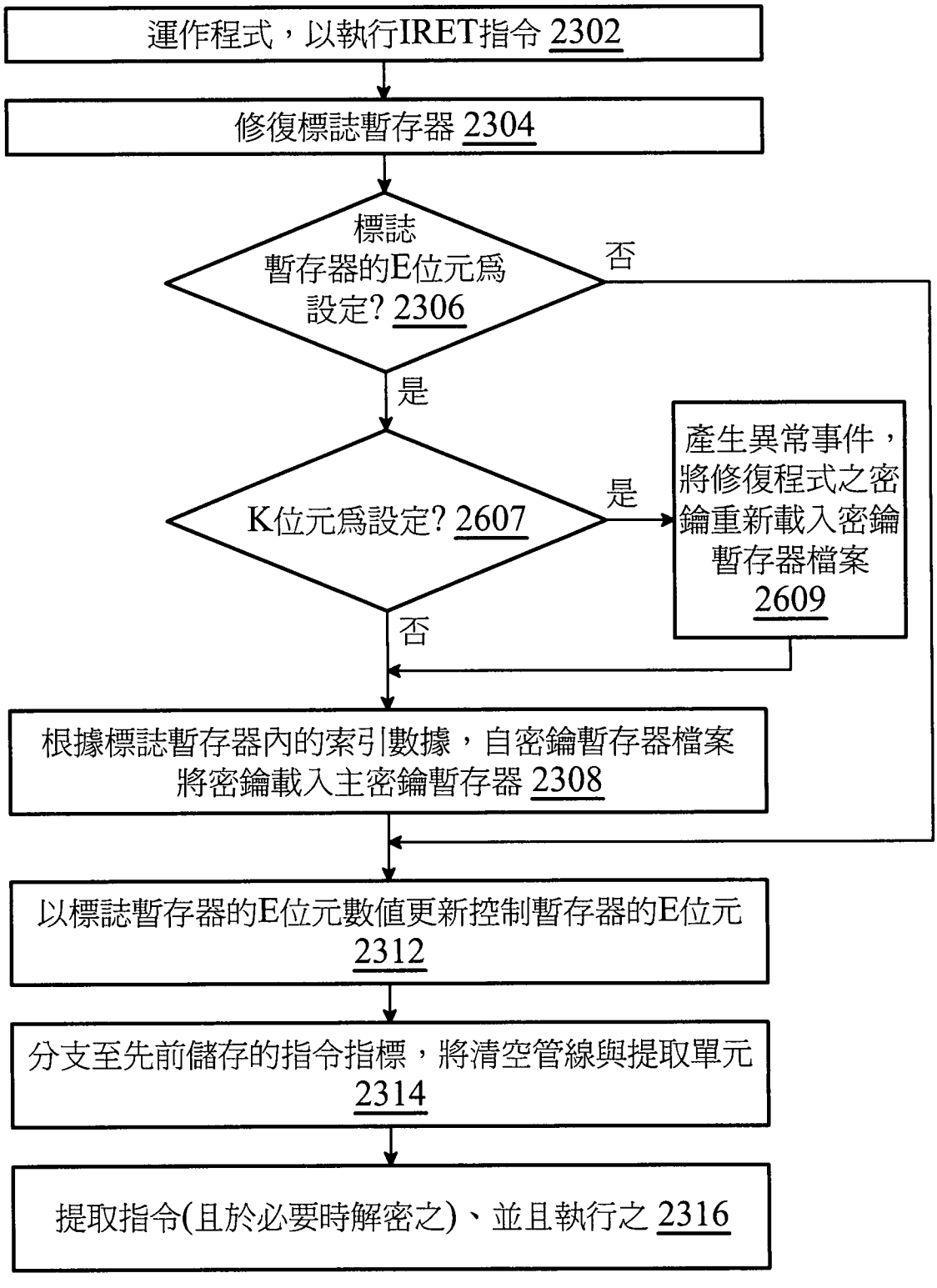
第 23 圖


密鑰暫存器檔案之暫存器 124
淘汰位元
2402

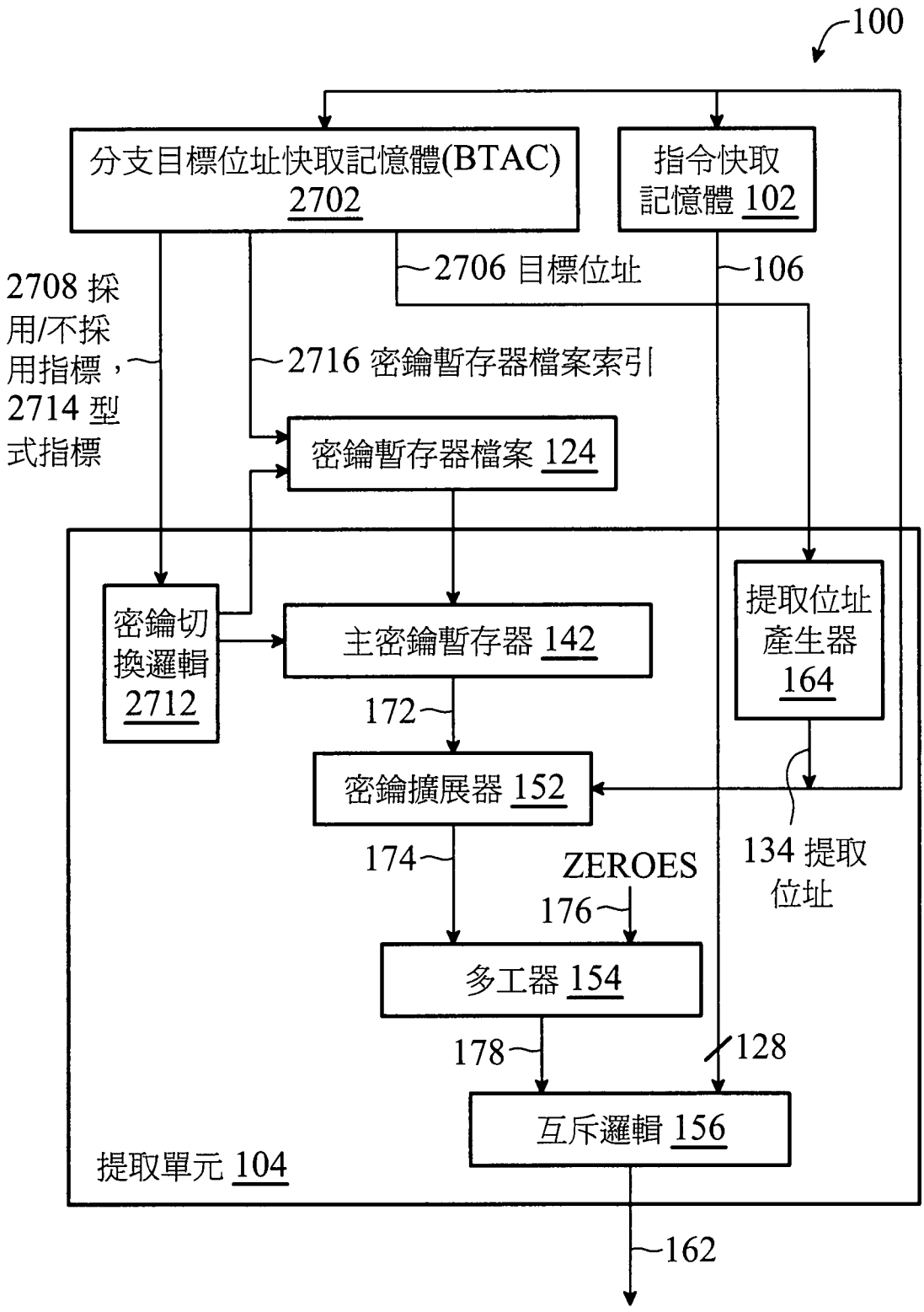
第 24 圖


要求執行新的加密程式 2202
密鑰
暫存器檔案有空間?
2204
無
有
將使用中的密鑰暫存器檔案的空間配置出來覆寫新值，並且設定其中K位元，且隨之設定標誌暫存器中的索引欄位 2506
第22圖方塊2208
第22圖方塊2212

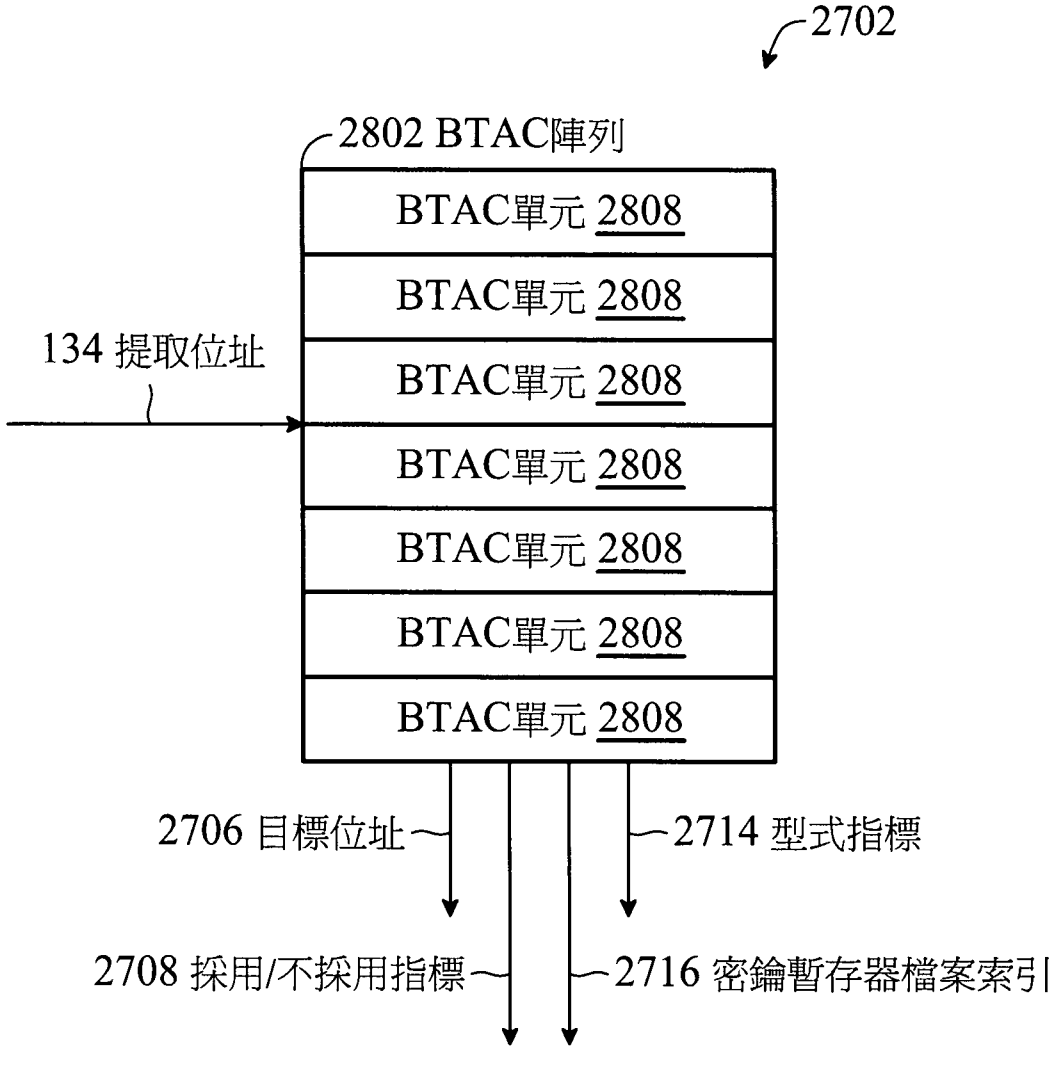
第 25 圖


運作程式，以執行IRET指令 2302
修復標誌暫存器 2304
標誌暫存器的E位元爲設定? 2306
否
是
K位元爲設定? 2607
是
產生異常事件，將修復程式之密鑰重新載入密鑰暫存器檔案 2609
否
根據標誌暫存器內的索引數據，自密鑰暫存器檔案將密鑰載入主密鑰暫存器 2308
以標誌暫存器的E位元數值更新控制暫存器的E位元 2312
分支至先前儲存的指令指標，將清空管線與提取單元 2314
提取指令(且於必要時解密之)、並且執行之 2316

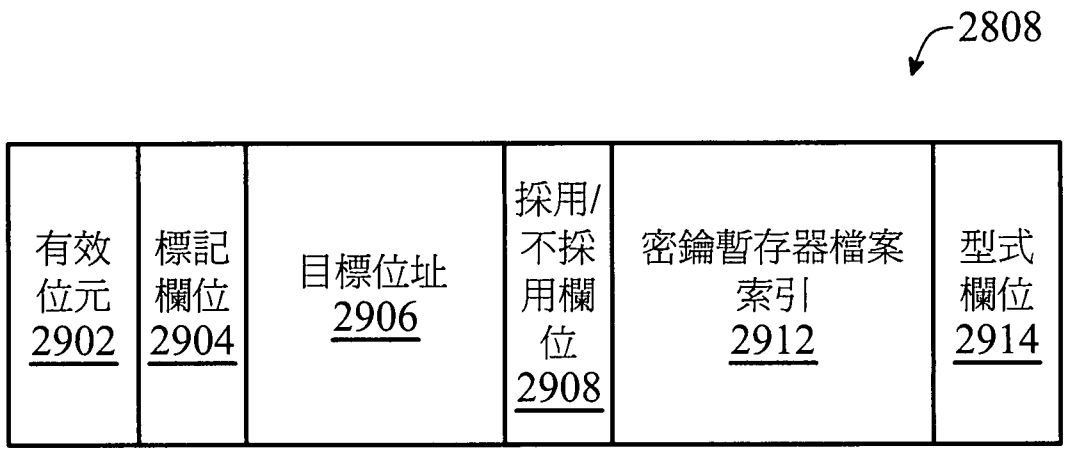
第 26 圖


100
分支目標位址快取記憶體(BTAC) 2702
指令快取記憶體 102
2706 目標位址
106
2708 採用/不採用指標，2714 型式指標
2716 密鑰暫存器檔案索引
密鑰暫存器檔案 124
密鑰切換邏輯 2712
主密鑰暫存器 142
提取位址產生器 164
172
密鑰擴展器 152
174
ZEROES
176
134 提取位址
多工器 154
178
128
互斥邏輯 156
提取單元 104
162

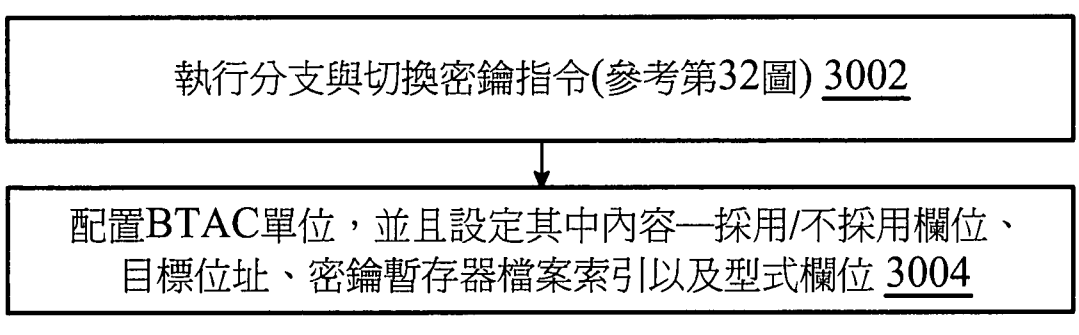
第 27 圖


2702
2802 BTAC陣列
BTAC單元 2808
BTAC單元 2808
BTAC單元 2808
BTAC單元 2808
BTAC單元 2808
BTAC單元 2808
BTAC單元 2808
134 提取位址
2706 目標位址
2714 型式指標
2708 採用/不採用指標
2716 密鑰暫存器檔案索引

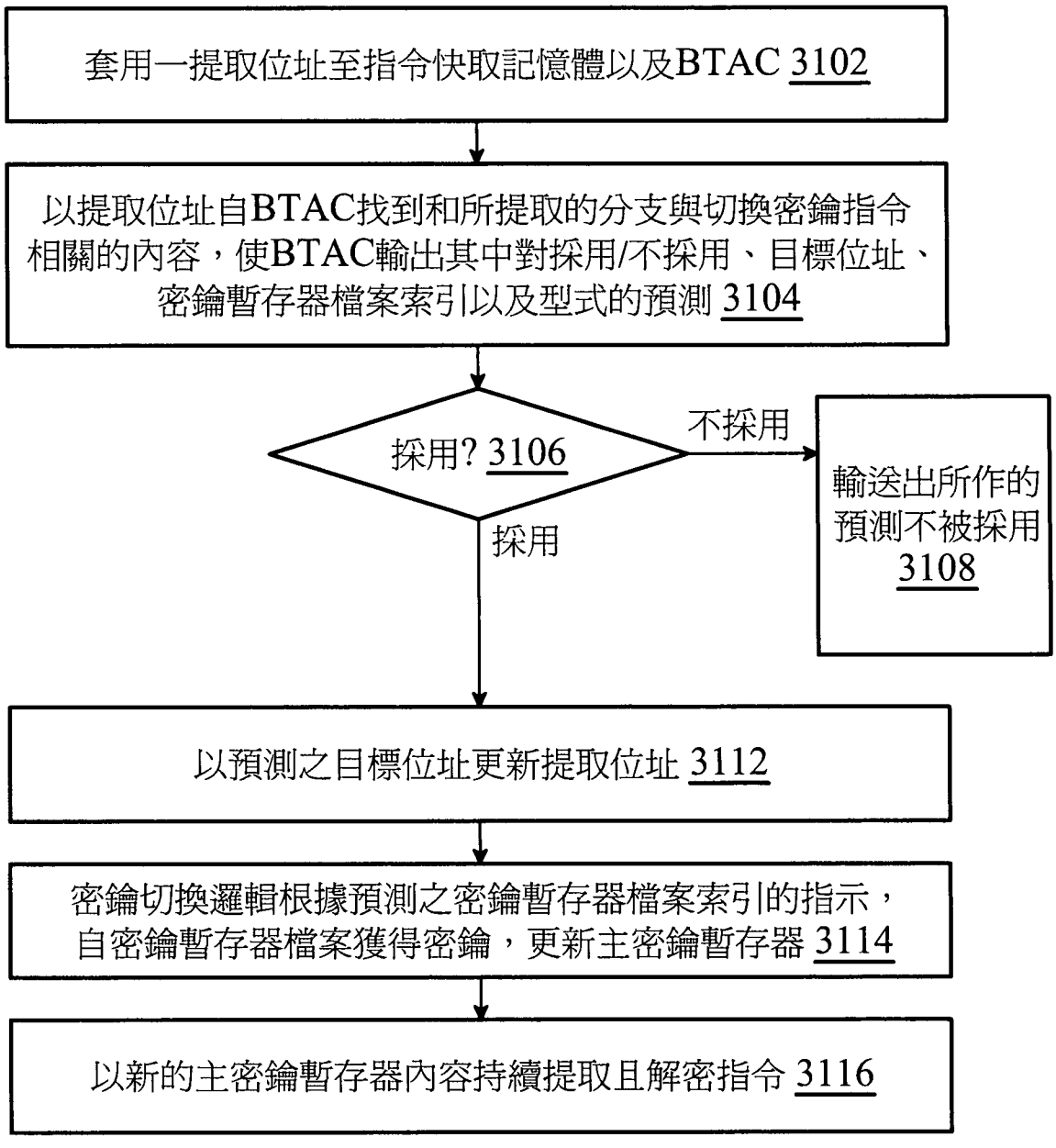
第 28 圖


2808
有效位元 2902
標記欄位 2904
目標位址 2906
採用/不採用欄位 2908
密鑰暫存器檔案索引 2912
型式欄位 2914

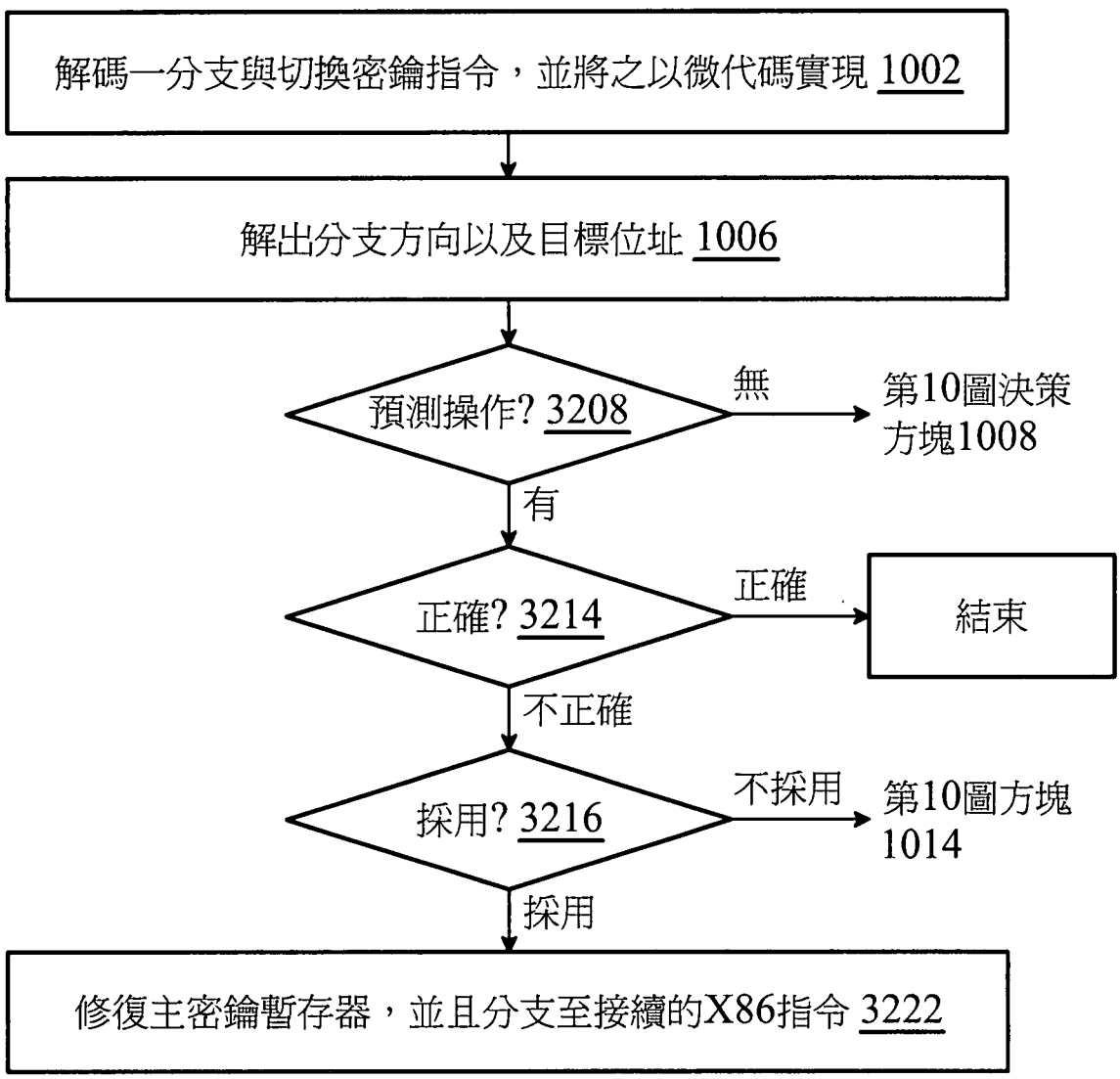
第 29 圖


執行分支與切換密鑰指令(參考第32圖) 3002
配置BTAC單位，並且設定其中內容—採用/不採用欄位、目標位址、密鑰暫存器檔案索引以及型式欄位 3004

第 30 圖


套用一提取位址至指令快取記憶體以及BTAC 3102
以提取位址自BTAC找到和所提取的分支與切換密鑰指令相關的內容，使BTAC輸出其中對採用/不採用、目標位址、密鑰暫存器檔案索引以及型式的預測 3104
採用? 3106
不採用
輸送出所作的預測不被採用 3108
採用
以預測之目標位址更新提取位址 3112
密鑰切換邏輯根據預測之密鑰暫存器檔案索引的指示，自密鑰暫存器檔案獲得密鑰，更新主密鑰暫存器 3114
以新的主密鑰暫存器內容持續提取且解密指令 3116

第 31 圖


解碼一分支與切換密鑰指令，並將之以微代碼實現 1002
解出分支方向以及目標位址 1006
預測操作? 3208
無
第10圖決策方塊1008
有
正確? 3214
正確
結束
不正確
採用? 3216
不採用
第10圖方塊1014
採用
修復主密鑰暫存器，並且分支至接續的X86指令 3222

第 32 圖