(19)日本国特許庁(JP)　(12)公表特許公報(A)　(11)特許出願公表番号
特表2017-518661
(P2017-518661A)
(43)公表日　平成29年7月6日(2017.7.6)

(51)Int.Cl.		F I			テーマコード（参考）
H04L 9/32	(2006.01)	H04L	9/00	675A	5B017
G09C 1/00	(2006.01)	G09C	1/00	640D	5J104
G06F 12/14	(2006.01)	G06F	12/14	510D	
G06F 21/52	(2013.01)	G06F	21/52		

審査請求 未請求　予備審査請求 有　（全 29 頁）

(21)出願番号　特願2016-560996 (P2016-560996)
(86)(22)出願日　平成27年4月14日 (2015.4.14)
(85)翻訳文提出日　平成28年10月5日 (2016.10.5)
(86)国際出願番号　PCT/US2015/025685
(87)国際公開番号　WO2015/160759
(87)国際公開日　平成27年10月22日 (2015.10.22)
(31)優先権主張番号　14/256,681
(32)優先日　平成26年4月18日 (2014.4.18)
(33)優先権主張国　米国 (US)

(71)出願人　507364838
クアルコム，インコーポレイテッド
アメリカ合衆国　カリフォルニア　９２１２１　サン　ディエゴ　モアハウス　ドライブ　５７７５
(74)代理人　100108453
弁理士　村山　靖彦
(74)代理人　100163522
弁理士　黒田　晋平
(72)発明者　キャン・エルキン・アカール
アメリカ合衆国・カリフォルニア・９２１２１－１７１４・サン・ディエゴ・モアハウス・ドライヴ・５７７５

最終頁に続く

(54)【発明の名称】ハードウェアベースのスタック制御情報の保護

(57)【要約】
プロセッサと関連付けられるスタックのコンテンツを保護するための技法が提供される。本技法は、プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信するステップであって、記憶命令がサブルーチンと関連付けられる制御情報を含む、ステップと、ソフトウェアプログラムから記憶命令を受信したことに応答して、安全な制御情報を生成するように制御情報を変更するステップと、安全な制御情報をスタックに記憶するステップと、ソフトウェアプログラムからロード命令を受信するステップと、ソフトウェアプログラムからロード命令を受信したことに応答して、スタックから安全な制御情報をロードするステップと、制御情報を復元するように安全な制御情報を変更するステップと、制御情報をソフトウェアプログラムに返すステップとを含む、方法を含む。

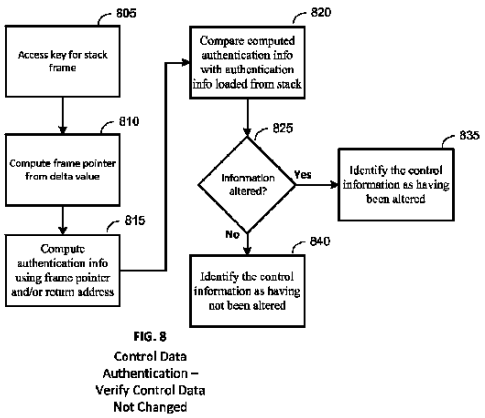


FIG. 8
Control Data Authentication – Verify Control Data Not Changed

【特許請求の範囲】
【請求項１】
プロセッサと関連付けられるスタックのコンテンツを保護するための方法であって、
前記プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信するステップであって、前記記憶命令がサブルーチンと関連付けられる制御情報を含む、ステップと、
前記ソフトウェアプログラムから前記記憶命令を受信したことに応答して、安全な制御情報を生成するように前記制御情報を変更するステップと、
前記安全な制御情報を前記スタックに記憶するステップと、
前記ソフトウェアプログラムからロード命令を受信するステップと、
前記ソフトウェアプログラムから前記ロード命令を受信したことに応答して、
前記安全な制御情報を前記スタックからロードするステップと、
前記制御情報を復元するように前記安全な制御情報を変更するステップと、
前記制御情報を前記ソフトウェアプログラムに返すステップと
を備える、方法。
【請求項２】
前記安全な制御情報を生成するように前記制御情報を変更するステップが、
前記安全な制御情報を生成するように前記制御情報を暗号化するステップを備える、請求項1に記載の方法。
【請求項３】
前記安全な制御情報を生成するように前記制御情報を暗号化するステップが、ブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して前記制御情報を暗号化するステップを備える、請求項2に記載の方法。
【請求項４】
前記安全な制御情報を生成するように前記制御情報を暗号化するステップが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して前記制御情報を暗号化するステップを備える、請求項2に記載の方法。
【請求項５】
前記制御情報を復元するように前記安全な制御情報を変更するステップが、
前記制御情報を復元するように前記安全な制御情報を復号するステップを備える、請求項2に記載の方法。
【請求項６】
前記安全な制御情報を生成するように前記制御情報を変更するステップが、
前記安全な制御情報を生成するように認証タグを前記制御情報に追加するステップを備える、請求項1に記載の方法。
【請求項７】
前記認証タグが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる、請求項6に記載の方法。
【請求項８】
前記制御情報を復元するように前記安全な制御情報を変更するステップが、前記安全な制御情報からフレームポインタおよびリターンアドレスを導出するステップを備える、請求項6に記載の方法。
【請求項９】
前記制御情報を復元するように前記安全な制御情報を変更するステップが、
タグ値を生成するためにメッセージ認証コードを前記フレームポインタおよび前記リターンアドレスに適用するステップと、
前記制御情報が前記スタック上にある間に変更されなかったことを確認するために、前記タグ値を前記安全な制御情報に含まれる前記認証タグと比較するステップと
を備える、請求項8に記載の方法。
【請求項１０】

プロセッサと関連付けられるスタックのコンテンツを保護するための前記プロセッサであって、
前記プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信するための手段であって、前記記憶命令がサブルーチンと関連付けられる制御情報を含む、手段と、
前記ソフトウェアプログラムから前記記憶命令を受信したことに応答して、安全な制御情報を生成するように前記制御情報を変更するための手段と、
前記安全な制御情報を前記スタックに記憶するための手段と、
前記ソフトウェアプログラムからロード命令を受信するための手段と、
前記ソフトウェアプログラムから前記ロード命令を受信したことに応答して、
前記安全な制御情報を前記スタックからロードするための手段と、
前記制御情報を復元するように前記安全な制御情報を変更するための手段と、
前記制御情報を前記ソフトウェアプログラムに返すための手段と
を備える、プロセッサ。
【請求項１１】
前記安全な制御情報を生成するように前記制御情報を変更するための前記手段が、
前記安全な制御情報を生成するように前記制御情報を暗号化するための手段を備える、請求項10に記載のプロセッサ。
【請求項１２】
前記安全な制御情報を生成するように前記制御情報を暗号化するための前記手段が、ブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して前記制御情報を暗号化するための手段を備える、請求項11に記載のプロセッサ。
【請求項１３】
前記安全な制御情報を生成するように前記制御情報を暗号化するための前記手段が、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して前記制御情報を暗号化するための手段を備える、請求項11に記載のプロセッサ。
【請求項１４】
前記制御情報を復元するように前記安全な制御情報を変更するための前記手段が、
前記制御情報を復元するように前記安全な制御情報を復号するための手段を備える、請求項11に記載のプロセッサ。
【請求項１５】
前記安全な制御情報を生成するように前記制御情報を変更するための前記手段が、
前記安全な制御情報を生成するように認証タグを前記制御情報に追加するための手段を備える、請求項10に記載のプロセッサ。
【請求項１６】
前記認証タグが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる、請求項15に記載のプロセッサ。
【請求項１７】
前記制御情報を復元するように前記安全な制御情報を変更するための前記手段が、前記安全な制御情報からフレームポインタおよびリターンアドレスを導出するための手段を備える、請求項15に記載のプロセッサ。
【請求項１８】
前記制御情報を復元するように前記安全な制御情報を変更するための前記手段が、
タグ値を生成するためにメッセージ認証コードを前記フレームポインタおよび前記リターンアドレスに適用するための手段と、
前記タグ値を前記安全な制御情報に含まれる前記認証タグと比較して、前記制御情報が前記スタック上にある間に変更されなかったことを確認するための手段と
を備える、請求項17に記載のプロセッサ。
【請求項１９】
メモリと、

1つまたは複数のサブルーチンと関連付けられるデータを記憶するための前記メモリ中のスタックと、
前記メモリに結合されるプロセッサと
を備え、前記プロセッサが、
前記プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信することであって、前記記憶命令がサブルーチンと関連付けられる制御情報を含む、受信することと、
前記ソフトウェアプログラムから前記記憶命令を受信したことに応答して、安全な制御情報を生成するように前記制御情報を変更することと、
前記安全な制御情報を前記スタックへ記憶することと、
前記ソフトウェアプログラムからロード命令を受信することと、 前記ソフトウェアプログラムから前記ロード命令を受信したことに応答して、前記プロセッサが、
前記安全な制御情報を前記スタックからロードすることと、
前記制御情報を復元するように前記安全な制御情報を変更することと、
前記制御情報を前記ソフトウェアプログラムに返すことと
を前記プロセッサに行わせるように構成されるプロセッサ実行可能命令を実行するように構成される、システム。

【請求項２０】
前記安全な制御情報を生成するように前記制御情報を変更するように構成されている前記プロセッサがさらに、
前記安全な制御情報を生成するように前記制御情報を暗号化するように構成される、請求項19に記載のシステム。

【請求項２１】
前記プロセッサが、ブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して前記制御情報を暗号化するように構成される、請求項20に記載のシステム。

【請求項２２】
前記プロセッサが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して前記制御情報を暗号化するように構成される、請求項20に記載のシステム。

【請求項２３】
前記プロセッサが、前記制御情報を復元するように前記安全な制御情報を復号するように構成される、請求項20に記載のシステム。

【請求項２４】
前記安全な制御情報を生成するように前記制御情報を変更するように構成されている前記プロセッサがさらに、
前記安全な制御情報を生成するように認証タグを前記制御情報に追加するように構成される、請求項19に記載のシステム。

【請求項２５】
前記認証タグが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる、請求項24に記載のシステム。

【請求項２６】
前記制御情報を復元するように前記安全な制御情報を変更するように構成されている前記プロセッサがさらに、前記安全な制御情報からフレームポインタおよびリターンアドレスを導出するように構成される、請求項24に記載のシステム。

【請求項２７】
前記制御情報を復元するように前記安全な制御情報を変更するように構成されている前記プロセッサがさらに、
メッセージ認証コードを前記フレームポインタおよび前記リターンアドレスに適用してタグ値を生成し、
前記制御情報が前記スタック上にある間に変更されなかったことを確認するために前記

タグ値を前記安全な制御情報に含まれる前記認証タグと比較する
ように構成される、請求項26に記載のシステム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ハードウェアベースのスタック制御情報の保護に関する。
【背景技術】
【０００２】
入力検証の欠損または誤りのような、ソフトウェア実装のエラーは、過剰なバッファアクセスおよびメモリの破損につながり得る。この実装のエラーは、ソフトウェアが不安定になること、または最終的に普通の条件のもとでクラッシュすることにつながり得る。しかしながら、コンピュータシステムを攻撃することに専心している悪意のある者により入力が操作されるとき、これらのメモリの破損およびオーバーフローのバグが、ソフトウェアコードの予想される挙動を変更するために、かつ攻撃者によって決定されるコードまたは機能を実行するために、攻撃者により悪用され得る。
【０００３】
攻撃者は通常、リターンアドレス、関数ポインタ、または仮想テーブルポインタのような、メモリ中のデータ構造を上書きすることによって、実行の制御権を得る。巨大なレガシーのコードベースにおけるすべての利用可能なメモリ破損のバグを見つけて修正することは、常に可能であるとは限らない。したがって、多くのコンピュータシステムが、「悪用軽減機構」と呼ばれる包括的な防御機能を含んでおり、これは、これらのバグを悪用して標的のコンピュータシステムの制御権を得るために、かつ/または標的のコンピュータシステムに損害を与えるために攻撃者が使用する、よく知られている技法に対しては効果的である。
【０００４】
コンピュータシステムに一般的に含まれる悪用軽減機構のいくつかの例には、データ実行防止(DEP)、スタック保護(SP)、およびアドレス空間レイアウトランダム化(ASLR)がある。DEP技法では、すべてのコードセクションが読取り専用としてマークされ、すべての書込み可能な領域が実行不可能である。通常、コードセクションの読取り専用という性質および書込み可能セクションの実行不可能という性質は、プロセッサのメモリ管理ユニット(MMU)によって課される。この技法は、攻撃者が自身のコードをデータエリアに置き、そのコードに対する実行フローを指示するのを防ぐことができる。SPの手法では、コンパイラが、スタック上でのバッファのオーバーフローを検出するための関数を実装する。スタックベースのバッファのオーバーフローは悪用されやすいことがあり、それは、ローカルアレイの直後のスタック上にリターンアドレスまたは保存されたリンクレジスタがあることが多いからである。攻撃者は、ローカルアレイをオーバーフローさせ、ローカルアレイにリターンアドレス/リンクレジスタを上書きさせ、ソフトウェアの実行フローをリダイレクトさせることが可能である。多くの従来のスタック保護の実装形態は、関数の最初においてスタック上にガード値を挿入し、その関数から戻る前にその値の変更について確認する。バッファのオーバーフローが発生した場合、ガード値はリターンアドレスとともに書き込まれており、それは、ガード値がスタックにおいてバッファとリターンアドレスの間に位置しているからである。ガード値は、SPの手法が安全であるために、ランダムな値および/または予測不可能な値でなければならない。ASLRの手法では、アプリケーションのメモリレイアウトが、各実行においてランダム化され得る。たとえば、コードのためのベースアドレス、静的データ、スタック、およびヒープがランダムに決定され得る。この手法は、悪用を難しくする。それは、攻撃の成功のために攻撃者が標的にする/操作する必要のあるコード/データ/ポインタの位置を、攻撃者が予測する必要があるからである。
【発明の概要】
【課題を解決するための手段】

【０００５】
本開示による、プロセッサと関連付けられるスタックのコンテンツを保護するための例示的な方法は、プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信するステップであって、記憶命令がサブルーチンと関連付けられる制御情報を含む、ステップと、ソフトウェアプログラムから記憶命令を受信したことに応答して、安全な制御情報を生成するように制御情報を変更するステップと、安全な制御情報をスタックに記憶するステップと、ソフトウェアプログラムからロード命令を受信するステップと、ソフトウェアプログラムからロード命令を受信したことに応答して、スタックから安全な制御情報をロードするステップと、制御情報を復元するように安全な制御情報を変更するステップと、制御情報をソフトウェアプログラムに返すステップとを含む。

【０００６】
そのような方法の実装形態は、次の特徴の1つまたは複数を含み得る。安全な制御情報を生成するように制御情報を変更するステップは、安全な制御情報を生成するように制御情報を暗号化するステップを含む。安全な制御情報を生成するように制御情報を暗号化するステップは、ブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して制御情報を暗号化するステップを含む。安全な制御情報を生成するように制御情報を暗号化するステップは、サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して制御情報を暗号化するステップを含む。制御情報を復元するように安全な制御情報を変更するステップは、制御情報を復元するように安全な制御情報を復号するステップを含む。安全な制御情報を生成するように制御情報を変更するステップは、安全な制御情報を生成するように認証タグを制御情報に追加するステップを含む。認証タグは、サブルーチンと関連付けられる現在のスタックフレームと関連付けられる。制御情報を復元するように安全な制御情報を変更するステップは、安全な制御情報からフレームポインタおよびリターンアドレスを導出するステップを含む。制御情報を復元するように安全な制御情報を変更するステップは、メッセージ認証コードをフレームポインタおよびリターンアドレスに適用してタグ値を生成するステップと、タグ値を安全な制御情報に含まれる認証タグと比較して、制御情報がスタック上にある間に変更されなかったことを確認するステップとを含む。

【０００７】
本開示による、プロセッサと関連付けられるスタックのコンテンツを保護するための例示的なプロセッサは、プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信するための手段であって、記憶命令がサブルーチンと関連付けられる制御情報を含む、手段と、ソフトウェアプログラムから記憶命令を受信したことに応答して、安全な制御情報を生成するように制御情報を変更するための手段と、安全な制御情報をスタックに記憶するための手段と、ソフトウェアプログラムからロード命令を受信するための手段と、ソフトウェアプログラムからロード命令を受信したことに応答して、スタックから安全な制御情報をロードするための手段と、制御情報を復元するように安全な制御情報を変更するための手段と、制御情報をソフトウェアプログラムに返すための手段とを含む。

【０００８】
そのようなプロセッサの実装形態は、次の特徴の1つまたは複数を含み得る。安全な制御情報を生成するように制御情報を変更するための手段は、安全な制御情報を生成するように制御情報を暗号化するための手段を含む。安全な制御情報を生成するように制御情報を暗号化するための手段は、ブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して制御情報を暗号化するための手段を含む。安全な制御情報を生成するように制御情報を暗号化するための手段は、サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して制御情報を暗号化するための手段を含む。制御情報を復元するように安全な制御情報を変更するための手段は、制御情報を復元するように安全な制御情報を復号するための手段を含む。安全な制御情報を生成するように制御情報を変更するための手段は、安全な制御情報を生成するように認証タグを制御情報に追加するための

手段を含む。認証タグは、サブルーチンと関連付けられる現在のスタックフレームと関連付けられる。制御情報を復元するように安全な制御情報を変更するための手段は、安全な制御情報からフレームポインタおよびリターンアドレスを導出するための手段を含む。制御情報を復元するように安全な制御情報を変更するための手段は、メッセージ認証コードをフレームポインタおよびリターンアドレスに適用してタグ値を生成するための手段と、タグ値を安全な制御情報に含まれる認証タグと比較して、制御情報がスタック上にある間に変更されなかったことを確認するための手段とを含む。

【０００９】

本開示による例示的なシステムは、メモリと、1つまたは複数のサブルーチンと関連付けられるデータを記憶するためのメモリ中のスタックと、メモリに結合されるプロセッサとを含む。プロセッサは、プロセッサに、プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信させ、記憶命令がサブルーチンと関連付けられる制御情報を含み、ソフトウェアプログラムから記憶命令を受信したことに応答して、安全な制御情報を生成するように制御情報を変更させ、安全な制御情報をスタックへ記憶させ、ソフトウェアプログラムからロード命令を受信させるように構成される、プロセッサ実行可能命令を実行するように構成され、ソフトウェアプログラムからロード命令を受信したことに応答して、プロセッサは、スタックから安全な制御情報をロードし、制御情報を復元するように安全な制御情報を変更し、制御情報をソフトウェアプログラムに返すように構成される。

【００１０】

そのようなシステムの実装形態は、次の特徴の1つまたは複数を含み得る。安全な制御情報を生成するように制御情報を変更するように構成されるプロセッサはさらに、安全な制御情報を生成するように制御情報を暗号化するように構成される。プロセッサは、ブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して制御情報を暗号化するように構成される。プロセッサは、サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して制御情報を暗号化するように構成される。プロセッサは、制御情報を復元するように安全な制御情報を復号するように構成される。安全な制御情報を生成するように制御情報を変更するように構成されるプロセッサはさらに、安全な制御情報を生成するように認証タグを制御情報に追加するように構成される。認証タグは、サブルーチンと関連付けられる現在のスタックフレームと関連付けられる。制御情報を復元するように安全な制御情報を変更するように構成されるプロセッサはさらに、安全な制御情報からフレームポインタおよびリターンアドレスを導出するように構成される。制御情報を復元するように安全な制御情報を変更するように構成されるプロセッサはさらに、メッセージ認証コードをフレームポインタおよびリターンアドレスに適用してタグ値を生成し、タグ値を安全な制御情報に含まれる認証タグと比較して、制御情報がスタック上にある間に変更されなかったことを確認するように構成される。

【図面の簡単な説明】

【００１１】

【図１】本明細書で論じられるハードウェアベースのスタック制御情報の保護技法が実施され得る、例示的なコンピュータシステムの機能ブロック図である。

【図２】本明細書で開示されるハードウェアベースのスタック制御情報の技法を実施するために使用され得る制御データ認証技法の実装形態を示す、図1からの例示的なコンピュータシステムの機能ブロック図である。

【図３】本明細書で開示されるハードウェアベースのスタック制御情報の技法を実施するために使用され得る制御データ暗号化技法の実装形態を示す、図1からの例示的なコンピュータシステムの機能ブロック図である。

【図４】本明細書で開示されるハードウェアベースのスタック制御の技法を実施するために使用され得る、プロセッサと関連付けられるスタックのコンテンツを保護するためのプロセスの流れ図である。

【図５】制御データ認証技法を使用して、安全な制御情報を生成するように制御情報を変

更するためのプロセスの流れ図である。
【図６】スタックに安全な制御情報を記憶するためのプロセスの流れ図である。
【図７】スタックから安全な制御情報をロードするためのプロセスの流れ図である。
【図８】制御データ認証技法を使用して生成される安全な制御情報がスタック上にある間に変更されたかどうかを決定するためのプロセスの流れ図である。
【図９】制御データ暗号化技法を使用して、安全な制御情報を生成するように制御情報を変更するためのプロセスの流れ図である。
【図１０】スタックに安全な制御情報を記憶するためのプロセスの流れ図である。
【図１１】スタックから安全な制御情報をロードするためのプロセスの流れ図である。
【図１２】制御データ暗号化技法を使用して生成される安全な制御情報がスタック上にある間に変更されたかどうかを決定するためのプロセスの流れ図である。
【発明を実施するための形態】
【００１２】
本明細書で開示される技法は、ハードウェアベースのスタック制御情報の保護を提供するように構成される。本技法は、悪意のある攻撃者が、コンピュータシステムの制御権を得るために、または不安定になることによるコンピュータシステム上で実行されるソフトウェアの挙動を引き起こすために、リターンアドレス(またはリンクレジスタ)、フレームポインタ、および/またはスタック上の他の制御情報の内容を変えるのを防ぐことができる。本明細書で開示される技法は、スタック上の制御情報の変更に成功することを制御情報がいつより難しくするかを特定するために、かつ、制御情報がいつ変更されたかを特定するのをより簡単にするために使用され得る、制御データ認証および制御データ暗号化の技法を含む。
【００１３】
例示的なハードウェア
図1は、本明細書で論じられるハードウェアベースのスタック制御情報の保護技法が実施され得る、例示的なコンピュータシステム100の機能ブロック図である。本明細書で開示されるハードウェアベースのスタック制御情報の保護技法は、スタック115のスタックフレーム145と関連付けられるリターンアドレス125およびフレームポインタ120のような制御情報が、悪意のある攻撃者または低品質のソフトウェアコードにより変更されることを防ぐために使用され得る。図1に示されるコンピュータシステムは、図4～図12に示されたプロセスを実施するために使用され得る。
【００１４】
コンピュータシステム100は、プロセッサ105とメモリ110とを含む。プロセッサ105は、たとえば、Intel(登録商標)CorporationまたはAMD(登録商標)によって作られたもののようなパーソナルコンピュータの中央処理装置(CPU)、マイクロコントローラ、特定用途向け集積回路(ASIC)などの、インテリジェントデバイスであり得る。メモリ110は、プロセッサ105がそこからデータを読み取ることができプロセッサ105がそこへデータを書き込むことができる、ランダムアクセスメモリ(RAM)および/または他のタイプのメモリを含み得る、非一時的記憶デバイスである。いくつかの実装形態では、メモリ110の一部分は、プロセッサ105によって読み取られ得るが更新され得ないソフトウェア命令および/またはデータを含み得る、読取り専用メモリ(ROM)であってよい。
【００１５】
メモリ110は、本明細書で説明されるような様々な関数を実行するようにプロセッサ105を制御するための命令を含むプロセッサ可読のプロセッサ実行可能ソフトウェアコードを記憶することができる(ただし、説明ではソフトウェアが関数を実行すると書かれていることがある)。ソフトウェアは、ネットワーク接続を介してダウンロードされることによってメモリ260にロードされること、ディスクからアップロードされることなどが可能である。さらに、ソフトウェアは、直接的に実行可能ではない、たとえば実行前にコンパイルを必要とすることがある。
【００１６】

メモリ110は、スタック115を含み得る。スタック115は、プロセッサ105によって実行されている1つまたは複数のソフトウェアプログラムの1つまたは複数のアクティブサブルーチンと関連付けられるデータを記憶するために使用され得る、メモリ110中のデータ構造である。いくつかの実装形態では、プロセッサ105によって実行されている各ソフトウェアプログラムおよび/またはスレッドは、異なるスタック115と関連付けられ得る。スタック115は、1つまたは複数のスタックフレーム145から構成され得る。スタックフレーム145は、プロセッサ105によるリターンでまだ終了されていないサブルーチンの状態情報を含む。スタックの典型的な実装形態では、スタックの頂部に位置するスタックフレームは、プロセッサ105によって現在実行されているサブルーチンと関連付けられるスタックフレームである。スタックフレーム145は、サブルーチンと関連付けられる情報、たとえばそのサブルーチンを呼び出すルーチンによってサブルーチンに渡された可能性のある任意のパラメータ値と、ソフトウェアプログラムのフローおよびプログラムがデータにアクセスする場所を制御するために使用され得る制御情報とを含み得る。たとえば、制御情報は、スタックフレーム145と関連付けられるサブルーチンのコーラ(caller)を指し返すリターンアドレス(RA)125と、フレームポインタ(FP)120とを含み得る。スタックフレーム145はまた、サブルーチンと関連付けられる他の情報を記憶するために使用され得る。

【００１７】

リターンアドレス125は、スタックフレーム145と関連付けられるサブルーチンを呼び出した命令のアドレスを指し、プロセッサ105がサブルーチンの実行を完了するとサブルーチンを呼び出した呼出し命令に戻るためにプロセッサ105によって後で使用され得る。

【００１８】

フレームポインタ120は、サブルーチンと関連付けられるスタックフレームの開始部分のアドレスを指し、サブルーチンの持続時間の間は変化しない。プロセッサ105はまた、スタック115の頂部を参照するスタックポインタ(SP)(図示されず)を維持するように構成されることがある。コンテンツがスタックに追加されるにつれてスタックポインタの値は変化するが、フレームポインタ120はスタックフレームと関連付けられるデータを指し続ける。フレームポインタ120はスタックフレーム内の固定された位置を指すために使用されてよく、これにより、フレームポインタ120は、スタックフレームと関連付けられるサブルーチンと関連付けられる引数および/またはローカル変数へのアクセスを容易にするために使用され得る固定された参照点として使用されることが可能になる。引数および/またはローカル変数の位置は、フレームポインタ120の値からの固定されたオフセットとして決定され得る。

【００１９】

本明細書で開示される技法は、リターンアドレス125および/またはフレームポインタ120が、低品質のコードにより、またはコンピュータシステム100を攻撃しようとする悪意のある試みを通じて変更されることを防ぐのを、助けるために使用され得る。悪意のある者が、リターンアドレスを異なる命令のリターンアドレスに上書きすることによって、プロセッサ105によるプログラムコードの実行を、サブルーチンを呼び出した命令とは異なる命令にリダイレクトすることを試みることがある。悪意のあるコードは、攻撃者によってスタック上に挿入されるデータを指すために使用され得る、フレームポインタを変更することもできる。攻撃者は、データを記憶するために実際に割り振られたスタックデータ構造の外部でデータをスタック115に書き込むために、スタックバッファのオーバーフローを悪用することがある。プロセッサ105によって実行されているプログラムコードが、データを記憶することが意図されるバッファに対してスタックに書き込まれているデータが実際に適切なサイズであることを確実にするために確認を行わない場合、スタック115上のデータが破損した状態になることがあり、かつ/または、攻撃者がそのような技法を利用して、コンピュータシステム100の制御権を奪い取るためにフレームポインタ120および/またはリターンアドレス125を変えることがある。本明細書で開示される技法は、フレームポインタ120およびリターンアドレス125のような制御情報を変更するのをより難しくするために、コンピュータシステム100のハードウェアにおいて実施され得る。いくつかの

実装形態はまた、プロセッサ105によって実行されているソフトウェアプログラムに制御情報を返す前に、スタック115からロードされた安全な制御情報を検証するステップを含み得る。

【００２０】

プロセッサ105は、スタック115にデータを記憶するための手段を提供する(一態様では、スタック115にデータを記憶するための手段は、スタック115にデータをプッシュするための手段である)。プロセッサ105はまた、スタック115からデータをロードするための手段を提供する(一態様では、スタック115からデータをロードするための手段は、スタック115からデータをポップするための手段である)。プロセッサ105はまた、プロセッサ105によって実行されているソフトウェアプログラムから記憶命令とロード命令とを受信するための手段として機能し得る。記憶命令は制御情報も含み得る。プロセッサ105は、安全な制御情報を生成するように制御情報を変更するための、かつスタック115に安全な制御情報を記憶するための手段を提供する。プロセッサ105はまた、スタック115から安全な制御情報をロードするための、かつ制御情報を復元するように安全な制御情報を変更するための手段を提供する。プロセッサ105はまた、安全な制御情報を生成するように制御情報を暗号化するための、かつ制御情報を復元するように安全な制御情報を復号するための手段を提供し得る。プロセッサ105はまた、認証タグを制御情報に追加して安全な制御情報を生成するための手段を提供することができる。プロセッサ105はまた、安全な制御情報からフレームポインタおよびリターンアドレスを導出することによって、制御情報を復元するように安全な制御情報を変更するための手段を提供する。プロセッサ105はまた、メッセージ認証コードをフレームポインタおよびリターンアドレスに適用してタグ値を生成するための手段と、タグ値を安全な制御情報に含まれる認証タグと比較して、制御情報がスタック上にある間に変更されなかったことを確認するための手段とを提供することができる。プロセッサ105は、別段規定されない限り、図4～図12に示されるプロセスの段階を実施するための手段を提供することができる。図4～図12が以下で詳細に論じられる。

【００２１】

図1は、わかりやすくするために、プロセッサ105およびメモリ110だけを示す。コンピュータシステム100は、ネットワークインターフェース、入力/出力ポートおよび/もしくは入出力デバイス、ワイヤレスインターフェース、ならびに/または本明細書で示されていない他のコンポーネントのような、追加の要素を含み得る。

【００２２】

図2は、本明細書で開示されるハードウェアベースのスタック制御情報の技法を実施するために使用され得る制御データ認証技法の実装形態を示す、図1からの例示的なコンピュータシステムの機能ブロック図である。図2に示される制御データ認証技法では、認証情報140の形態の安全な制御情報が生成され、スタック115に記憶される。図2に示される実装形態では、認証情報140がフレームポインタ120の代わりにスタックにプッシュされている。認証情報140は、安全な制御情報がスタック上にある間に変更されていないことを確認するために使用され得る。たとえば、安全な制御情報と関連付けられるサブルーチンが完了すると、プロセッサ105は、スタック115からロードされたリターンアドレス125に基づいて新たな認証情報を生成することができる。プロセッサ105は、スタック115にプッシュされた認証情報140を生成するために使用されたのと同じ技法を使用して、新たな認証情報を生成することができる。プロセッサ105は、スタック115からポップされたリターンアドレス125から生成される新たな認証情報を、認証情報140に含まれる認証情報140と比較することができる。新たな認証情報がスタックからの認証情報140と一致しない場合、プロセッサ105は、リターンアドレス125が変更されたので、または破損したので、例外処理ルーチンを実施して、実行されているアプリケーションの実行を中止することができる。新たな認証情報が認証情報140と一致する場合、リターンアドレス125は変更されておらず、または破損しておらず、プロセッサ105は、リターンアドレス125によって参照される命令を実行することに進むことができる。図2に示される制御データ認証技法の追加の詳細が、図4～図12に示されるプロセスに関して以下で論じられる。

【００２３】
図3は、本明細書で開示されるハードウェアベースのスタック制御情報の技法を実施するために使用され得る制御データ暗号化技法の実装形態を示す、図1からの例示的なコンピュータシステム100の機能ブロック図である。図3に示される制御データ暗号化技法では、暗号化されたフレームデータ135の形態の安全な制御情報が生成され、スタックにプッシュされる。図3に示される実装形態では、暗号化されたフレームデータ135が、フレームポインタ120およびリターンアドレス125の代わりにスタックにプッシュされている。暗号化されたフレームデータ135は、安全な制御情報がスタック上にある間に変更されていないことを確認するために使用され得る。たとえば、安全な制御情報と関連付けられるサブルーチンが完了すると、プロセッサ105は、暗号化されたフレームデータ135を復号して、暗号化されたフレームデータ135がスタック115上にある間に変更されたかどうかを決定することができる。新たな暗号化されたフレームデータ135がスタック115上にある間に変更された場合、プロセッサ105は、制御情報が変更されているので、または破損しているので、例外処理ルーチンを実施して実行されているアプリケーションの実行を中止するように構成され得る。暗号化されたフレームデータ135がスタック上にある間に変更されなかった場合、プロセッサは、復号された暗号化されたフレームデータ135からフレームポインタ120およびリターンアドレス125を取得することができ、リターンアドレス125によって参照される命令を実行することに進むことができる。図3に示される制御データ認証技法の追加の詳細が、図4～図12に示される処理に関して以下で論じられる。

【００２４】
例示的な実装形態
図4は、本明細書で開示されるハードウェアベースのスタック制御の技法を実施するために使用され得る、プロセッサと関連付けられるスタックのコンテンツを保護するためのプロセスの流れ図である。図4に示されるプロセスは、図1に示されるコンピュータシステム100を使用して実施され得る。図1に示される例示的なコンピュータシステム100は、図4に示される処理が実施され得る1つのコンピュータシステムの一例を提供し、図4に示されるプロセスは、図1に示される例とは異なるアーキテクチャを有するコンピュータシステムのプロセッサによって実施され得る。

【００２５】
プロセスは、プロセッサ105によって実行されているソフトウェアプログラムから記憶命令を受信することで開始し得る(段階401)。記憶命令は、ソフトウェアプログラムがスタック115へ記憶させることを望む、サブルーチンと関連付けられる制御情報を含み得る。図1に示される例示的なコンピュータシステムを参照すると、プロセッサ105は、ソフトウェアプログラムから受信される記憶命令を受信したことに応答して、プロセッサ105によって実行されているソフトウェアプログラムのサブルーチンと関連付けられる情報を、スタック115に記憶するように構成され得る。たとえば、プロセッサ105は、フレームポインタ120およびリターンアドレス125を、スタックフレーム145と関連付けられるサブルーチンのためにスタックに記憶される制御情報の一部として、スタック115にプッシュするように構成され得る。他の実装形態では、制御情報は、フレームポインタ120およびおリターンアドレス125に加えて、かつ/またはその代わりに、他の情報を含むことがある。

【００２６】
制御情報は次いで、安全な制御情報を生成するように変更され得る(段階405)。プロセッサ105は、本明細書で論じられる制御データ認証および/または制御データ暗号化の技法を使用して、ソフトウェアプログラムと関連付けられる制御情報を変更するように構成され得る。制御情報を変更することは、攻撃者が、フレームポインタ120、リターンアドレス125、および/またはスタック115上の他の制御情報を変更することによってコンピュータシステム100の制御権を奪い取るのをより難しくする。いくつかの実装形態では、プロセッサ105は、安全な制御情報がスタック115上にある間に変更されたかどうかを確認することができる。制御データ暗号化技法において、プロセッサ105は、安全な制御情報を生成するように制御情報を暗号化するように構成される。制御データ暗号化技法を使用して

安全な制御情報を生成するための例示的なプロセスが、図9に示される。制御データ認証技法において、プロセッサ105は、制御情報に追加され、安全な制御情報がスタック115上にある間に変更されなかったことを確認するために使用され得る、認証タグを生成するように構成される。認証タグは、プロセッサ105に知られているがプロセッサ105によって実行されているソフトウェアコードに対して利用可能ではない秘密鍵を使用して作成され得る。いくつかの実装形態では、この鍵は、コンピュータシステム100の信用されるメモリコンポーネントのコンテンツがコンピュータシステム100によって実行されているソフトウェアコードによってアクセスまたは変更され得ないように、プロセッサのハードウェアにおいて実装されてよく、またはその信用されるメモリコンポーネントに記憶されてよい。認証タグがどのように生成されることがあるかの追加の詳細が、以下で詳細に論じられる。たとえば、図5に示されるプロセスは、認証タグを生成するために使用されることがある1つの技法を論じる。認証タグは、フレームポインタ120、リターンアドレス125、および/または保護されるべき他の制御情報を考慮することができる。

【００２７】

安全な制御情報は次いで、スタック115に記憶され得る(段階410)。いくつかの実装形態では、安全な制御情報は、制御情報の代わりにスタック115に記憶され得る。図3に示される例示的な実装形態はある実装形態の例を提供し、その実装形態では、図1に示される例のように、暗号化されたフレームデータ135が、フレームポインタ120およびリターンアドレス125を含む変更されていない制御情報の代わりにスタック115にプッシュされている。そのような実装形態では、変更されていない制御情報はスタック上で利用可能ではなく、このことは、悪意のあるコードおよび/または悪意のある者による攻撃が、保護されていない制御データにアクセスするのを防ぐ。他の実装形態では、安全な制御情報の少なくとも一部分が、制御情報に加えてスタックにプッシュされてよく、制御情報がスタック上にある間に変更されなかったことを決定するために使用され得る。図2に示される例示的な実装形態はある実装形態の例を提供し、その実装形態では、図1に示される例のように、認証情報140が、フレームポインタ120の代わりにスタック115にプッシュされている。リターンアドレス125は、認証情報140に加えてスタック115にプッシュされている。図2に示される例における認証情報は、スタックフレームと関連付けられるフレームポインタおよびリターンアドレスに少なくとも一部基づいて決定されてよく、認証情報140は、リターンアドレスおよび/またはスタック115上の他の情報が変更されているかどうかを決定するために使用されてよい。

【００２８】

ロード命令は、プロセッサ105によって実行されているソフトウェアプログラムから受信され得る(段階415)。プロセッサ105は、ソフトウェアプログラムからのロード命令に応答してスタック115から制御情報をロードするように、かつ、ロード命令を出すソフトウェアプログラムに制御情報を提供するように構成され得る。いくつかの実装形態では、プロセッサ115は、スタック115上のメモリを空けるために、ロード命令に応答してスタック115から制御情報(および/または安全な制御情報)を削除するように構成され得る。プロセスは、プロセッサ105によって実行されているソフトウェアプログラムからロード命令を受信したことに応答して、段階417、420、および425に続き得る。

【００２９】

安全な制御情報は、スタックからロードされ得る(段階417)。サブルーチンと関連付けられるスタック115のコンテンツは、サブルーチンの完了に応答して、スタック115上のメモリを空けるためにスタック115からポップされ得る。このクリーンアッププロセスの一部は、スタック115から安全な制御情報をポップすることを含んでよく、いくつかの実装形態では、スタックから他の制御情報をポップすることを含んでよい。

【００３０】

プロセスは、制御情報を復元するように安全な制御情報を変更することに続き得る(段階420)。プロセッサ105は、安全な制御情報から制御情報を復元するように構成され得る。安全な制御情報が変更されているかどうかをプロセッサ105がどのように復元するかは

、安全な制御情報を生成するためにどの技法が使用されたかによる。安全な制御情報を生成するために制御データ認証技法が使用された場合、スタックフレーム145のための制御情報に基づいて生成される認証タグは、スタック上にプッシュされる認証タグを生成するために使用された鍵を使用して再生成されてよく、新たに生成された鍵が、スタック115上のスタックフレーム145のための安全な制御情報に含まれていた鍵と一致するかどうかの決定が行われ得る。新たに生成された認証タグが、安全な制御情報に含まれる認証タグと一致しない場合、安全な制御情報は、スタック115上にある間に破損しており、または変更されている。図8は、安全な制御情報から制御情報を復元するために使用され得る。段階405において安全な制御情報を生成するために制御データ暗号化技法が使用された場合、スタックフレーム145のためにスタックにプッシュされた安全な制御情報を備える暗号化されたフレームデータ135は、段階405においてデータを暗号化するために使用される暗号化技法と関連付けられる適切な鍵を使用して復号され得る。暗号化されたフレームデータ135が、制御データを暗号化するために使用される鍵と関連付けられる復号鍵を使用して適切に復号されない場合、安全な制御情報は、スタック115上にある間に破損しており、または変更されている。図12は、制御データ暗号化技法を使用して生成される安全な制御情報を復元するための例示的なプロセスを示す。

【００３１】

制御情報は次いで、ソフトウェアプログラムに返され得る(段階425)。プロセッサ105は次いで、ロード命令をプロセッサ105に出したソフトウェアプログラムに制御情報を返すことができる。いくつかの実装形態では、プロセッサ105は、安全な制御データが変更されたかどうかを決定するように、かつ、ソフトウェアプログラムの処理を中止できる、および/または、悪意のあるコードもしくは低品質に実装されたコードが変更されたもしくは破損した制御データを利用するのを防ぐことができる他の動作を実行できる、例外をスロー(throw)するように構成され得る。たとえば、制御データ認証が使用され、認証タグが一致しない場合、プロセッサ105は、ソフトウェアプログラムの実行を中止し、または他の処理タスクを実行することができる。別の例では、制御データ暗号化が使用される場合、プロセッサ105は、フレームポインタ120と関連付けられるアドレスおよび/またはリターンアドレス125がアドレスの有効範囲内に入るかどうかを決定するように構成され得る。プロセッサ105は、アドレスがアドレスの許容可能範囲の外側にある場合、ソフトウェアプログラムの実行を中止し、または他の処理タスクを実行するように構成されてよく、許容可能範囲の外側にあることは、アドレス情報がスタック115上にある間に変更されたこと、または破損したことを示し得る。

【００３２】

図5は、制御データ認証技法を使用して、安全な制御情報を生成するように制御情報を変更するためのプロセスの流れ図である。図5に示されるプロセスは、図4に示されるプロセスの段階405を実施するために使用され得る。制御データ認証技法は、保護されるべき制御情報に少なくとも一部基づいて認証タグを生成するために使用されてよく、認証タグは、制御情報がスタック115上にある間に変更されたかどうかを決定するために使用され得る。図5に示されるプロセスは、別段示されない限りプロセッサ105によって実施され得る。

【００３３】

スタックフレームと関連付けられる鍵がアクセスされ得る(段階505)。いくつかの実装形態では、同じ鍵が、スタックフレームの各々のための認証タグを生成するために使用され得る。他の実装形態では、各スタックフレームが複数の鍵の1つと関連付けられるように、異なる鍵が複数の鍵から選択され得る。いくつかの実装形態では、各スタックフレームは異なる鍵と関連付けられてよく、その鍵がそのスタックフレームのための認証情報を生成するために使用され得る。いくつかの実装形態では、スタック115上の各スタックフレームは、フレームポインタ120および/またはリターンアドレス125のようなスタックフレーム145と関連付けられる制御情報がスタック115上にある間に変更されなかったことを確認するために使用され得る認証情報を生成するために使用される、固有の鍵と関連付け

られる。いくつかの実装形態では、プロセッサ105は、スタックフレームがスタック115上で作成されるにつれて各スタックフレームのための鍵を生成し、プロセッサ105がアクセス可能であるがコンピュータシステム100上で実行されているソフトウェアコードがアクセス可能ではない保護されたメモリに鍵を記憶するように構成され得る。いくつかの実装形態では、1つまたは複数の鍵は、プロセッサがアクセス権を有するコンピュータシステム100の信頼されるメモリコンポーネントに記憶されてよく、プロセッサ105は、信頼されるメモリコンポーネントに記憶されている1つまたは複数の鍵からスタックフレームと関連付けられる認証情報を生成するために使用すべき鍵を選択するように構成され得る。スタック115上のスタックフレームのための認証情報を生成するために使用される鍵は、悪意のあるユーザが制御情報を変更し変更された制御情報に基づいて新たな認証情報を生成できないようにするために、秘密にされるべきである。この鍵は、その鍵と関連付けられるスタックフレームがスタック115からポップされると再使用され得る。

【００３４】

プロセスはまた、フレームポインタ120のためのデルタ値を計算するステップを含み得る(段階510)。フレームポインタ120は、スタック115上のスタックポインタのアドレスに対するオフセットを表すデルタ値として符号化され得る。デルタ値は、フレームポインタ120を記憶するために最初に割り振られたメモリ中の空間がスタックフレーム145のためのデルタ値および認証情報を記憶するために使用され得る程度に、フレームポインタ120よりも十分に小さいことがある。

【００３５】

プロセスはまた、スタックフレームと関連付けられる制御情報のための認証情報を計算するステップを含み得る(段階515)。認証情報は、リターンアドレス125、フレームポインタ120、フレームポインタのデルタ値、および/または、スタック115上にある間に認められていない変更について監視されるべき他の制御情報に基づいて、計算され得る。いくつかの実装形態では、認証タグは、メッセージ認証コード(MAC)アルゴリズムおよび段階505においてアクセスされる鍵を使用して計算される、MACタグを備える。MACタグは、同じ鍵を使用して生成され、かつ検証され得る。MACアルゴリズムは、暗号学的ハッシュ関数、ブロック暗号アルゴリズム、ユニバーサルハッシュ化、または他のタイプの技法に基づき得る。プロセッサ105は、スタックフレームの少なくともいくつかが異なる鍵を使用して認証される場合に、スタック115上のどのスタックフレームを認証するためにどの鍵が使用されたかを追跡するように構成され得る。他の実装形態では、データの完全性および安全な制御情報の信頼性を検証するために使用され得る制御情報にある処理を適用することによって、認証情報を生成するために、他のタイプの技法が使用されることがある。

【００３６】

図6は、制御データ認証技法を使用して生成される安全な制御情報をスタックに記憶するためのプロセスの流れ図である。図6に示されるプロセスは、図5に示されるプロセスが図4に示されるプロセスの段階405を実施するために使用される場合、図4に示されるプロセスの段階410を実施するために使用され得る。図6に示されるプロセスは、別段示されない限りプロセッサ105によって実施され得る。

【００３７】

デルタ値および認証情報が、スタック115に記憶され得る(段階605)。プロセッサ105は、フレームポインタ120の代わりにデルタ値および認証情報をスタック115にプッシュするように構成されてよく、それは、デルタ値がフレームポインタ120から導出され、フレームポインタ120の値を計算するために後で使用され得るからである。デルタ値および認証情報のサイズは、デルタ値および認証タグがフレームポインタ120と同じ量のスタック115上のメモリに適合することを可能にし得る。いくつかの実装形態では、認証情報は、保護されるべき安全な制御情報に基づいて計算されるMACタグを備え、これは、フレームポインタ120、フレームポインタ120と関連付けられるデルタ値、リターンアドレス125、および/または他の制御情報を含むことがある。

【００３８】

リターンアドレス125はまた、スタック115に記憶され得る(段階610)。リターンアドレス125は、変更されておらず、スタック115にプッシュされ得る。リターンアドレス125がスタック115上にある間に変更される場合、段階605においてスタックにプッシュされる安全な制御情報に含まれる認証タグは、制御情報がスタック115からポップされる時点で変更されたリターンアドレスに基づいて生成される認証タグと一致せず、このことは、制御情報がスタック115上にある間に変更されたことと、例外処理手順が実行されるべきであることとをプロセッサ105に伝える。

【００３９】

図7は、制御データ認証技法を使用して生成されたスタック115から安全な制御情報を取り出すためのプロセスの流れ図である。図7に示されるプロセスは、図4に示されるプロセスの段階417を実施するために使用され得る。図7に示されるプロセスは、別段示されない限りプロセッサ105によって実施され得る。図7に示されるプロセスは、図6に示されるプロセスにおいてスタック115に記憶される情報を取り出すために使用され得る。

【００４０】

プロセスは、スタック115からリターンアドレス125をロードするステップを含み得る(段階705)。デルタ値および認証情報も、スタックからロードされ得る(段階710)。図6に示されるプロセスに関して上で論じられたように、リターンアドレス125は、リターンアドレス125を変更することなくスタック115にプッシュされ得る。リターンアドレス125がスタック115上にある間に変更されていない場合、スタックにプッシュされる安全な制御情報とともに含まれる認証タグは、スタックからポップされた制御情報から生成される認証タグと一致しない。

【００４１】

図8は、制御データ認証技法を使用して生成される安全な制御情報がスタック上にある間に変更されたかどうかを決定するためのプロセスの流れ図である。図8に示されるプロセスは、図4に示されるプロセスの段階425の少なくとも一部分を実施するために使用されてよく、ここで、プロセッサ105は任意選択で、制御情報を要求したソフトウェアプログラムに制御情報を返す前に制御情報がスタック115上にある間に変更されなかったことを確認するように構成される。図8に示されるプロセスは、図5に示されるプロセスを使用して作成された安全な制御情報がスタック115上にある間に変更されなかったことを確認するために使用され得る。図8に示されるプロセスは、プロセッサ105によって実施され得る。

【００４２】

スタックフレームと関連付けられる鍵がアクセスされ得る(段階805)。スタックフレームと関連付けられる鍵は、制御情報がスタック115上にある間に変更されなかったことを確実にするために使用される認証情報を生成するために使用される鍵へのアクセス権を悪意のある者が取得するのを防ぐために、プロセッサ105がアクセス可能であるがプロセッサ105によって実行されている認められていないプログラムコードがアクセス可能ではない保護されたメモリに記憶され得る。上で論じられたように、鍵はメッセージ認証コード(MAC)と関連付けられ得る。スタックフレームと関連付けられる鍵は、制御情報がスタック115からロード/ポップされる時点でスタック115から取得される制御情報に基づいて認証タグを計算することによって、安全な制御情報のコンテンツを認証するために使用され得る。制御情報がスタック115上にある間に変更されている場合、計算された認証タグは、図6に示されるプロセスの段階605においてスタック115に記憶/プッシュされた認証タグと一致しない。

【００４３】

スタックフレームと関連付けられるフレームポインタ120は、スタック115からポップされるデルタ値から計算され得る(段階810)。フレームポインタ120は、スタックポインタに対するスタック115上のアドレスを表すデルタ値として符号化され得る。フレームポインタの代わりにデルタ値をスタック115に記憶することで、スタックフレームのための認証情報を記憶するために使用され得るスタック115上の空間を空けることができる。フレー

ムポインタ120は、スタックポインタによって表されるアドレスにデルタ値のオフセットを加算することによって、デルタ値から計算され得る。

【００４４】

認証情報は、フレームポインタ120、リターンアドレス125、および/またはスタックフレーム145と関連付けられる鍵を使用する他の制御情報を使用して計算され得る(段階815)。認証情報は、スタック115に記憶/プッシュされた認証情報を計算するために使用されたのと同じ技法を使用して計算されるべきである。たとえば、フレームポインタ120、リターンアドレス125、および/または他の制御情報からMACタグを生成するために特定のMACアルゴリズムが使用された場合、同じMACアルゴリズムが、スタック115に記憶/プッシュされた認証情報に含まれるMACタグを生成するために使用されたのと同じ制御情報要素に適用されるべきである。

【００４５】

フレームポインタ120、リターンアドレス125、および/または他の制御情報を使用して、かつ、スタックフレームと関連付けられる鍵を使用して計算される認証情報は、安全な制御情報の一部としてスタック115上に記憶された安全な制御情報とともに含まれる認証情報と比較され得る(段階820)。たとえば、認証情報がMACタグを備える場合、安全な制御情報に含まれるMACタグは、フレームポインタ120、リターンアドレス125、および/またはスタック115からポップされた他の制御情報を使用して計算されるMACタグと比較され得る。計算される認証タグがスタック115から取得された認証タグと一致するかどうかの決定が行われ得る(段階825)。計算された認証タグがスタック115から取得された認証タグと一致しない場合、安全な制御情報は、安全な制御情報がスタック115にプッシュされた後に変更されたものとして識別され得る(段階835)。そうではなく、計算された認証タグがスタック115から取得された認証タグと一致する場合、安全な制御情報は、安全な制御情報がスタック115にプッシュされた後に変更されなかったものとして識別され得る(段階840)。

【００４６】

図9は、制御データ暗号化技法を使用して、安全な制御情報を生成するように制御情報を変更するためのプロセスの流れ図である。図9に示されるプロセスは、図4に示されるプロセスの段階405を実施するために使用され得る。制御データ暗号化技法は、スタック115上にある間に保護されるべき制御情報を暗号化するために使用され得る。図9に示されるプロセスは、別段示されない限り図1に示されるプロセッサ105によって実施され得る。

【００４７】

スタックフレーム145と関連付けられる鍵がアクセスされ得る(段階905)。スタックフレームと関連付けられる暗号化鍵は、安全な制御データを暗号化し復号して安全な制御情報がスタック115上にある間に変更されなかったことを確実にするために使用される鍵へのアクセス権を攻撃者が取得するのを防ぐために、プロセッサ105がアクセス可能であるがプロセッサ105によって実行されている認められていないプログラムコードがアクセス可能ではない保護されたメモリに記憶され得る。スタックフレームと関連付けられる暗号化鍵は、制御情報がスタック115からポップされる時点で安全な制御情報のコンテンツを復号するために使用され得る。制御情報がスタック115上にある間に変更されている場合、復号鍵は無効な結果を生み出す。いくつかの実装形態では、安全な制御データを暗号化するために使用される暗号化鍵は、安全な制御データを復号するために使用されるのと同じ鍵であってよい。他の実装形態では、暗号化鍵および復号鍵は異なる鍵であってよい。いくつかの実装形態では、同じ暗号化鍵が複数のスタックフレームのためのデータを暗号化するために使用されてよく、各スタックフレームは異なる鍵と関連付けられないことがある。いくつかの実装形態では、プロセッサ105は、暗号化鍵と復号鍵のプールへのアクセス権を有してよく、プロセッサ105は、利用可能な暗号化鍵のプールから、ある特定のスタックフレームのための安全な制御情報を生成するために使用されるべき暗号化鍵を選択することができる。

【００４８】

スタックフレームと関連付けられる制御情報は、安全な制御情報を生み出すために、スタックフレームと関連付けられる鍵を使用して暗号化され得る(段階910)。制御情報は、リターンアドレス125および/またはフレームポインタ120を含み得る。制御情報は、他の実装形態では追加の情報を含むことがある。制御情報を暗号化することで、安全な制御情報がスタック115上にある間に制御情報に対して認められていない変更が行われるのを防ぐことができる。

【００４９】

図10は、制御データ暗号化技法を使用して生成されたスタックに安全な制御情報を記憶するためのプロセスの流れ図である。図10に示されるプロセスは、図4に示されるプロセスの段階410を実施するために使用され得る。安全な制御情報は、暗号化されたフレームデータ135を備え、追加の制御情報または他の情報を含むことがある。

【００５０】

暗号化された制御情報は、スタック115に記憶/プッシュされ得る(段階1005)。暗号化された制御情報(暗号化されたフレームデータ135とも本明細書では呼ばれる)は、特定のスタックフレーム145と関連付けられることがある。プロセッサ105は、図9に示されるプロセスの段階910において暗号化された安全な制御情報をスタック115にプッシュするように構成され得る。暗号化されたフレームデータ135は、プロセッサ105が別のサブルーチンを実行することに備えてスタック115にプッシュされてよく、現在実行されているサブルーチンのための情報は、現在実行されているサブルーチンのためのスタックにプッシュされる。暗号化された形式でスタックに制御情報をプッシュすることで、攻撃者がスタック115のコンテンツを変更することによって制御情報を変更するのを防ぐことができる。攻撃者は、リターンアドレスまたは安全な制御情報に含まれる他の情報を変えることに成功できるようになるためには、安全な制御情報を暗号化するために使用される鍵についての知識がなければならない。

【００５１】

リターンアドレス125はまた任意選択で、スタックに記憶/プッシュされ得る(段階1010)。いくつかの実装形態では、リターンアドレス125はまた、安全な制御情報に加えてスタックにプッシュされることがある。そのような実装形態では、リターンアドレスは、分岐予測および/またはリターンアドレスを利用する他の技法のために使用され得る。リターンアドレス125は、安全な制御情報を復号して、リターンアドレスがスタック115上にある間に変更されなかったことを確認することによって、検証され得る。

【００５２】

図11は、制御データ暗号化技法を使用して生成されたスタック115からの安全な制御情報を取り出すためのプロセスの流れ図である。図11に示されるプロセスは、図4に示されるプロセスの段階415を実施するために使用され得る。図11に示されるプロセスは、スタックフレーム145と関連付けられるサブルーチンが完了すると、プロセッサ105によって実行されてよい。プロセッサ105は、メモリ110中の空間を空けるために、スタック115からロードされた情報を削除するように構成され得る。

【００５３】

リターンアドレスは、リターンアドレスがスタック115上にプッシュされた場合、スタック115からロードされ得る(段階1105)。プロセッサ105は、リターンアドレスが図10に示されるプロセスの段階1010においてスタックにプッシュされた場合、スタック115からリターンアドレスをポップするように構成され得る。いくつかの実装形態では、リターンアドレスはスタックにプッシュされず、プロセスは段階1110に進むことができ、そこで安全な制御情報がスタックからポップされる。

【００５４】

暗号化された安全な制御情報は、スタック115からロードされ得る(段階1110)。プロセッサ105は、スタック115から安全な制御情報をポップすることができる。たとえば、プロセッサ105は、スタックフレーム145のクリーンアップの一部として、スタックフレーム145と関連付けられるサブルーチンが完了すると、スタック115から安全な制御情報をポップ

することができる。安全な制御情報は次いで、復号され検証され得る。安全な制御情報がスタック115上にある間に変更されていない場合、プロセッサ105は、リターンアドレスによって示されるサブルーチンを実行することに進み得る。それ以外の場合、プロセッサ105は、安全な制御情報がスタック115上にある間に変更されている場合、例外処理手順を実行することができる。

【００５５】

図12は、制御データ暗号化技法を使用して生成される安全な制御情報がスタック115上にある間に変更されたかどうかを決定するためのプロセスの流れ図である。図12に示されるプロセスは、図4に示されるプロセスの段階425の少なくとも一部分を実施するために使用されてよく、ここで、プロセッサ105は任意選択で、制御情報を要求したソフトウェアプログラムに制御情報を返す前に制御情報がスタック115上にある間に変更されなかったことを確認するように構成される。

【００５６】

スタックフレームと関連付けられる鍵がアクセスされ得る(段階1205)。スタックフレームと関連付けられる鍵は、暗号化されるフレームデータ135を暗号化して安全な制御情報がスタック115上にある間に変更されなかったことを確実にするために使用される鍵へのアクセス権を悪意のある者が取得するのを防ぐために、プロセッサ105がアクセス可能であるがプロセッサ105によって実行されている、認められていないプログラムコードがアクセス可能ではない保護されたメモリに記憶され得る。スタックフレームと関連付けられる鍵は、制御情報がスタック115からポップされる時点で安全な制御情報のコンテンツを復号するために使用され得る。いくつかの実装形態では、暗号化されたフレームデータ135を暗号化するために使用される鍵は、安全な制御情報を復号するために使用され得る鍵と同じであってよい。他の実装形態では、安全な制御情報を暗号化して復号するために使用される鍵は、異なる鍵である。プロセッサ105は、複数の暗号化鍵が利用可能である場合にどのスタックフレームがどの暗号化鍵によって暗号化されたかを追跡し、暗号化されたフレームデータ135を復号するための適切な復号鍵を選択するように構成され得る。

【００５７】

暗号化された安全な制御情報は、制御情報を復元するための鍵を使用して復号されてよく(段階1210)、暗号化された制御情報がスタック115上にある間に変更されたかどうか、または破損したかどうかの決定が行われ得る(段階1215)。プロセッサ105は、段階1205において取得された鍵を使用して、暗号化されたフレームデータ135を復号するように構成され得る。プロセッサ105は、暗号化されたフレームデータ135を復号することによって復元される制御情報中のアドレスをアドレスの有効範囲と比較することによって、暗号化されたフレームデータ135が変更されたかどうかを決定するように構成され得る。暗号化されたフレームデータ135を復号することによって復元される制御情報中のフレームポインタ120および/またはリターンアドレス125がこの範囲の外側にある場合、暗号化されたフレームデータ135は、スタック115上にある間に変更されたか、または破損した可能性がある。

【００５８】

データが変更された場合、または破損した場合、安全な制御情報は、変更されたものとして識別され得る(段階1220)。それ以外の場合、安全な制御情報は、変更されていないものとして識別され得る(段階1225)。データが変更されたという指示に応答して、プロセッサ105は、ソフトウェアプログラムの処理を中止できる、かつ/または、悪意のあるコードもしくは低品質に実装されたコードが変更されたもしくは破損した制御データを利用するのを防ぐことができる他の動作を実行できる、例外をスローするように構成され得る。データが変更されていないという指示に応答して、プロセッサ105は、ロードコマンドをプロセッサ105に出したソフトウェアプログラムに制御データを返すように構成され得る(図4に示されるプロセスの段階405参照)。

【００５９】

本明細書で説明された方法は、適用例に応じて様々な手段によって実装されてよい。た

とえば、これらの方法は、ハードウェア、ファームウェア、ソフトウェア、またはそれらの任意の組合せにおいて実装されてよい。ハードウェア実装の場合、各処理ユニットは、本明細書で説明された機能を実行するように設計された、1つもしくは複数の特定用途向け集積回路(ASIC)、デジタル信号プロセッサ(DSP)、デジタル信号処理デバイス(DSPD)、プログラマブル論理デバイス(PLD)、フィールドプログラマブルゲートアレイ(FPGA)、プロセッサ、コントローラ、マイクロコントローラ、マイクロプロセッサ、電子デバイス、他の電子ユニット、またはそれらの組合せにおいて実装されてよい。

【００６０】

ファームウェアおよび/またはソフトウェアの実装形態の場合、方法は、本明細書で説明された機能を実行するモジュール(たとえば、手順、機能など)を用いて実装されてよい。命令を有形に具現化するいずれの機械可読媒体も、本明細書で説明される方法を実装する際に使用されてよい。たとえば、ソフトウェアコードは、メモリに記憶されてよく、プロセッサユニットによって実行されてよい。メモリは、プロセッサユニット内またはプロセッサユニットの外部に実装されてよい。本明細書で使用される場合、「メモリ」という用語は、任意のタイプの長期、短期、揮発性、不揮発性、または他のメモリを指し、特定のメモリのタイプもしくはメモリの数、または媒体のタイプに限定されるものではない。有形の媒体は、ランダムアクセスメモリ、磁気ストレージ装置、光学ストレージ媒体のような、機械可読媒体の1つまたは複数の物品を含む。

【００６１】

ファームウェアおよび/またはソフトウェアで実装される場合、機能は、コンピュータ可読媒体上に1つまたは複数の命令またはコードとして記憶されてよい。例として、データ構造体で符号化されたコンピュータ可読媒体、およびコンピュータプログラムで符号化されたコンピュータ可読媒体がある。コンピュータ可読媒体は、物理的なコンピュータ記憶媒体を含む。記憶媒体は、コンピュータによってアクセスされ得る任意の利用可能な媒体であってよい。限定ではなく例として、そのようなコンピュータ可読媒体は、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、または、命令もしくはデータ構造の形式で所望のプログラムコードを記憶するために使用されコンピュータによってアクセスされ得る任意の他の媒体を備えてよく、disk(ディスク)およびdisc(ディスク)は、本明細書で使用される場合、コンパクトディスク(CD)、レーザーディスク(登録商標)、光ディスク、デジタル多目的ディスク(DVD)、フロッピーディスクおよびブルーレイディスクを含み、ディスク(disk)は通常、データを磁気的に再生し、一方、ディスク(disc)は、レーザを用いてデータを光学的に再生する。上記のものの組合せも、コンピュータ可読媒体の範囲内に含まれるべきである。そのような媒体はまた、機械可読であり得る非一時的媒体の例を提供し、コンピュータは、そのような非一時的媒体から読み取ることができる機械の一例である。

【００６２】

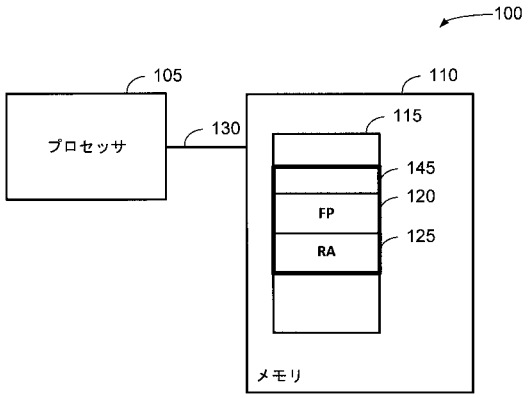
本明細書で論じられた一般的な原理は、本開示または特許請求の範囲の精神または範囲から逸脱することなく、他の実装に適用されてよい。

【符号の説明】

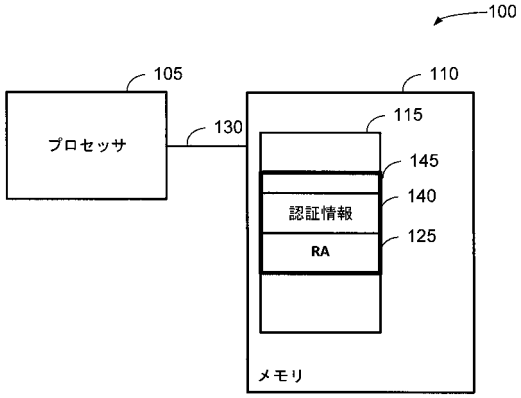
【００６３】

100 コンピュータシステム
105 プロセッサ
110 メモリ
120 フレームポインタ
125 リターンアドレス
135 暗号化されたフレームデータ
140 認証情報
145 スタックフレーム

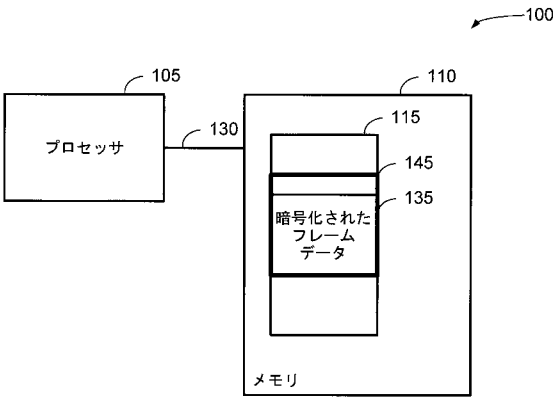
【図１】



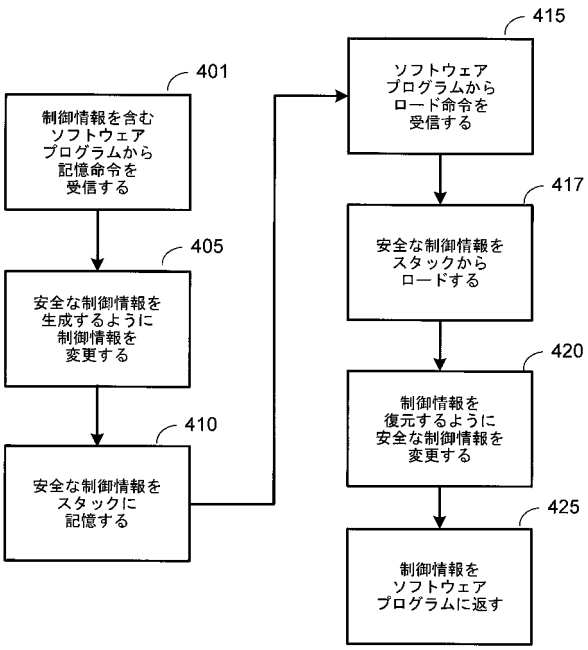
【図２】



【図３】

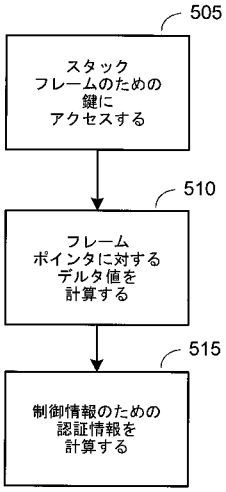


【図４】



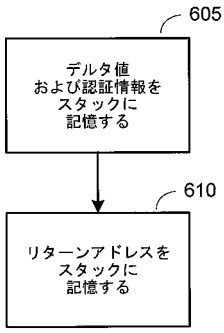
コールスタックの
コンテンツを保護する

【図５】



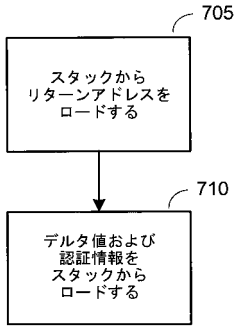
制御データ認証

【図６】



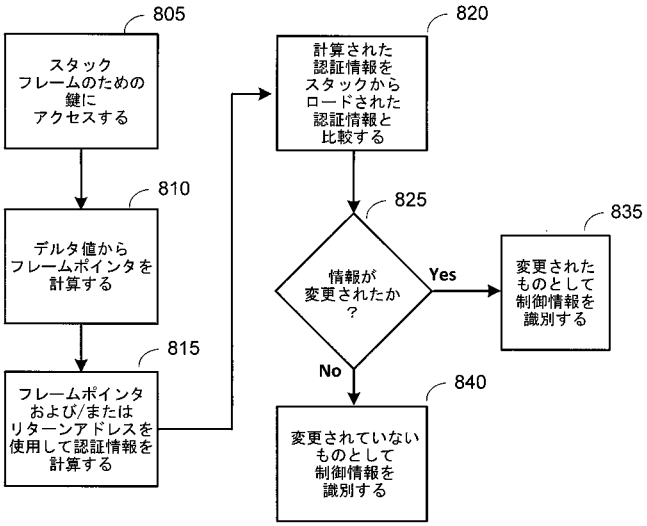
制御データ認証

【図７】



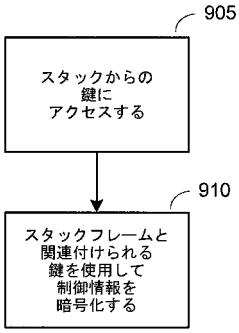
スタックからのロード情報

【図８】



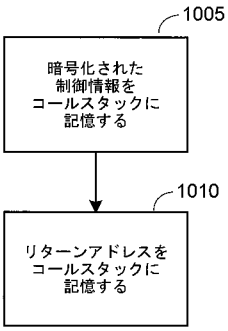
制御データ認証
ー制御データが
変更されていないことを
確認する

【図９】



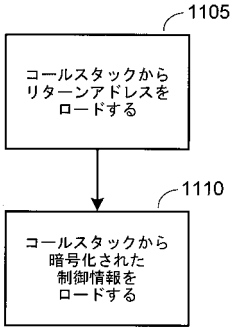
制御データ暗号化

【図１０】



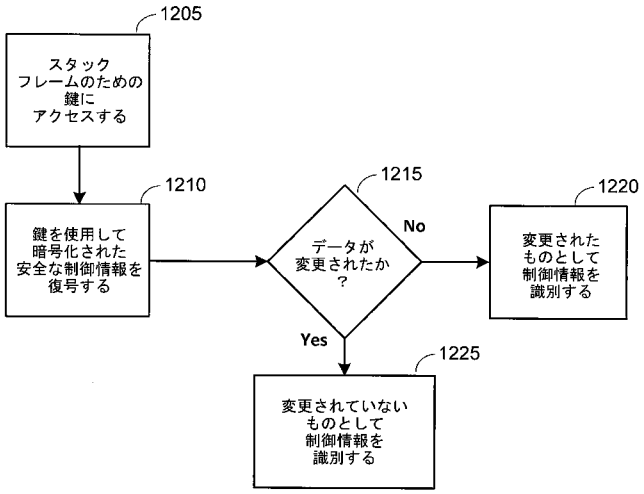
暗号化された制御情報を記憶する

【図１１】



スタックから暗号化された
情報をロードする

【図１２】



制御データ暗号化
－制御データが
変更されていないことを
確認する

【手続補正書】
【提出日】平成28年10月19日(2016.10.19)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
プロセッサと関連付けられるスタックのコンテンツを保護するための方法であって、
前記プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信するステップであって、前記記憶命令がサブルーチンと関連付けられる制御情報を含む、ステップと、
前記スタックのスタックポインタから、前記制御情報と関連付けられるアドレスからのオフセット値を計算するステップと、
現在のスタックフレームおよび前記オフセットと関連付けられる鍵を使用して認証情報を生成するするステップと
によって、前記サブルーチンと関連付けられる前記現在のスタックフレームと関連付けられる前記鍵を使用して、前記ソフトウェアプログラムから前記記憶命令を受信したことに応答して、安全な制御情報を生成するように前記制御情報を変更するステップと、
前記安全な制御情報を前記スタックに記憶するステップと、
前記ソフトウェアプログラムからロード命令を受信するステップと、
前記ソフトウェアプログラムから前記ロード命令を受信したことに応答して、
前記安全な制御情報を前記スタックからロードするステップと、
前記制御情報を復元するように前記安全な制御情報を変更するステップと、
前記制御情報を前記ソフトウェアプログラムに返すステップと
を備える、方法。
【請求項２】
前記安全な制御情報を生成するように前記制御情報を変更するステップが、
前記安全な制御情報を生成するように前記制御情報を暗号化するステップを備える、請求項1に記載の方法。
【請求項３】
前記安全な制御情報を生成するように前記制御情報を変更するステップが、前記フレーム固有の鍵およびブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して認証タグを生成するステップを備える、請求項1に記載の方法。
【請求項４】
前記安全な制御情報を生成するように前記制御情報を暗号化するステップが、前記フレーム固有の鍵を使用して前記制御情報を暗号化するステップを備える、請求項2に記載の方法。
【請求項５】
前記制御情報を復元するように前記安全な制御情報を変更するステップが、
前記制御情報を復元するように前記安全な制御情報を復号するステップを備える、請求項2に記載の方法。
【請求項６】
前記安全な制御情報を生成するように前記制御情報を変更するステップが、
前記安全な制御情報を生成するように認証タグを前記制御情報に追加するステップを備える、請求項1に記載の方法。
【請求項７】
前記認証タグが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる、請求項6に記載の方法。

【請求項８】
前記制御情報を復元するように前記安全な制御情報を変更するステップが、前記安全な制御情報からフレームポインタおよびリターンアドレスを導出するステップを備える、請求項6に記載の方法。
【請求項９】
前記制御情報を復元するように前記安全な制御情報を変更するステップが、
タグ値を生成するためにメッセージ認証コードを前記フレームポインタおよび前記リターンアドレスに適用ステップと、
前記制御情報が前記スタック上にある間に変更されなかったことを確認するために、前記タグ値を前記安全な制御情報に含まれる前記認証タグと比較するステップと
を備える、請求項8に記載の方法。
【請求項１０】
プロセッサと関連付けられるスタックのコンテンツを保護するための前記プロセッサであって、
前記プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信するための手段であって、前記記憶命令がサブルーチンと関連付けられる制御情報を含む、受信するための手段と、
前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して、前記ソフトウェアプログラムから前記記憶命令を受信したことに応答して、安全な制御情報を生成するように前記制御情報を変更するための手段であって、
前記スタックのスタックポインタから、前記制御情報と関連付けられるアドレスからのオフセット値を計算するための手段と、
現在のスタックフレームおよび前記オフセットと関連付けられる前記鍵を使用して認証情報を生成するための手段と
をさらに備える、変更するための手段と、
前記安全な制御情報を前記スタックに記憶するための手段と、
前記ソフトウェアプログラムからロード命令を受信するための手段と、
前記ソフトウェアプログラムから前記ロード命令を受信したことに応答して、
前記安全な制御情報を前記スタックからロードするための手段と、
前記制御情報を復元するように前記安全な制御情報を変更するための手段と、
前記制御情報を前記ソフトウェアプログラムに返すための手段と
を備える、プロセッサ。
【請求項１１】
前記安全な制御情報を生成するように前記制御情報を変更するための前記手段が、
前記安全な制御情報を生成するように前記制御情報を暗号化するための手段を備える、請求項10に記載のプロセッサ。
【請求項１２】
前記安全な制御情報を生成するように前記制御情報を変更するための前記手段が、前記フレーム固有の鍵およびブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して認証タグを生成するための手段を備える、請求項10に記載のプロセッサ。
【請求項１３】
前記安全な制御情報を生成するように前記制御情報を暗号化するための前記手段が、前記フレーム固有の鍵を使用して前記制御情報を暗号化するための手段を備える、請求項11に記載のプロセッサ。
【請求項１４】
前記制御情報を復元するように前記安全な制御情報を変更するための前記手段が、
前記制御情報を復元するように前記安全な制御情報を復号するための手段を備える、請求項11に記載のプロセッサ。
【請求項１５】
前記安全な制御情報を生成するように前記制御情報を変更するための前記手段が、

前記安全な制御情報を生成するように認証タグを前記制御情報に追加するための手段を備える、請求項10に記載のプロセッサ。

【請求項１６】

前記認証タグが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる、請求項15に記載のプロセッサ。

【請求項１７】

前記制御情報を復元するように前記安全な制御情報を変更するための前記手段が、前記安全な制御情報からフレームポインタおよびリターンアドレスを導出するための手段を備える、請求項15に記載のプロセッサ。

【請求項１８】

前記制御情報を復元するように前記安全な制御情報を変更するための前記手段が、

タグ値を生成するためにメッセージ認証コードを前記フレームポインタおよび前記リターンアドレスに適用するための手段と、

前記制御情報が前記スタック上にある間に変更されなかったことを確認するために前記タグ値を前記安全な制御情報に含まれる前記認証タグと比較するための手段と

を備える、請求項17に記載のプロセッサ。

【請求項１９】

メモリと、

1つまたは複数のサブルーチンと関連付けられるデータを記憶するための前記メモリ中のスタックと、

前記メモリに結合されるプロセッサとを備え、前記プロセッサが、

前記プロセッサによって実行されているソフトウェアプログラムから記憶命令を受信することであって、前記記憶命令がサブルーチンと関連付けられる制御情報を含む、受信することと、

前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる鍵を使用して、前記ソフトウェアプログラムから前記記憶命令を受信したことに応答して、安全な制御情報を生成するように前記制御情報を変更することであって、前記プロセッサがさらに、

前記スタックのスタックポインタから、前記制御情報と関連付けられるアドレスからのオフセット値を計算し、

前記現在のスタックフレームおよび前記オフセットと関連付けられる前記鍵を使用して認証情報を生成する

ように構成される、変更することと、

前記安全な制御情報を前記スタックへ記憶することと、

前記ソフトウェアプログラムからロード命令を受信することと、

前記ソフトウェアプログラムから前記ロード命令を受信したことに応答して、前記プロセッサが、

前記安全な制御情報を前記スタックからロードすることと、

前記制御情報を復元するように前記安全な制御情報を変更することと、

前記制御情報を前記ソフトウェアプログラムに返すことと

を前記プロセッサに行わせるように構成されるプロセッサ実行可能命令を実行するように構成される、システム。

【請求項２０】

前記安全な制御情報を生成するように前記制御情報を変更するように構成されている前記プロセッサがさらに、

前記安全な制御情報を生成するように前記制御情報を暗号化するように構成される、請求項19に記載のシステム。

【請求項２１】

前記プロセッサが、前記フレーム固有の鍵を使用して、かつブロック暗号ベースのメッセージ認証コードアルゴリズムを使用して認証タグを生成することによって、前記制御情

報を変更するように構成される、請求項19に記載のシステム。
【請求項２２】
前記プロセッサが、前記フレーム固有の鍵を使用して前記制御情報を暗号化するように構成される、請求項20に記載のシステム。
【請求項２３】
前記プロセッサが、前記制御情報を復元するように前記安全な制御情報を復号するように構成される、請求項20に記載のシステム。
【請求項２４】
前記安全な制御情報を生成するように前記制御情報を変更するように構成されている前記プロセッサがさらに、
前記安全な制御情報を生成するように認証タグを前記制御情報に追加するように構成される、請求項19に記載のシステム。
【請求項２５】
前記認証タグが、前記サブルーチンと関連付けられる現在のスタックフレームと関連付けられる、請求項24に記載のシステム。
【請求項２６】
前記制御情報を復元するように前記安全な制御情報を変更するように構成されている前記プロセッサがさらに、前記安全な制御情報からフレームポインタおよびリターンアドレスを導出するように構成される、請求項24に記載のシステム。
【請求項２７】
前記制御情報を復元するように前記安全な制御情報を変更するように構成されている前記プロセッサがさらに、
タグ値を生成するためにメッセージ認証コードを前記フレームポインタおよび前記リターンアドレスに適用し、
前記制御情報が前記スタック上にある間に変更されなかったことを確認するために前記タグ値を前記安全な制御情報に含まれる前記認証タグと比較する
ように構成される、請求項26に記載のシステム。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/025685

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/52
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/065929 A1 (MILLIKEN WALTER CLARK [US]) 3 April 2003 (2003-04-03) paragraph [0010] paragraph [0038] paragraph [0070] - paragraph [0093]; figure 6 paragraph [0094] - paragraph [0106]; figure 7	1-27
A	US 2003/217277 A1 (NARAYANAN RAM GOPAL LAKSHMI [US]) 20 November 2003 (2003-11-20) abstract paragraph [0037]	6-9, 15-18, 24-27
A	US 2003/182572 A1 (COWAN STANLEY CRISPIN [US] ET AL) 25 September 2003 (2003-09-25) abstract the whole document	1-27

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 July 2015	06/08/2015

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Vinck, Bart

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2015/025685

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003065929 A1	03-04-2003	US 2003065929 A1	03-04-2003
		US 2011022855 A1	27-01-2011
US 2003217277 A1	20-11-2003	NONE	
US 2003182572 A1	25-09-2003	US 2003182572 A1	25-09-2003
		US 2008060077 A1	06-03-2008

フロントページの続き

(81)指定国 AP(BW,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV,MC,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,CO,CR,CU,CZ,DE,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IR,IS,JP,KE,KG,KN,KP,KR,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,ME,MG,MK,MN,MW,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL,SM,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US

(72)発明者 エーリヒ・ジェームズ・プロンドケ
アメリカ合衆国・カリフォルニア・９２１２１－１７１４・サン・ディエゴ・モアハウス・ドライヴ・５７７５
(72)発明者 ロバート・ジェイ・ターナー
アメリカ合衆国・カリフォルニア・９２１２１－１７１４・サン・ディエゴ・モアハウス・ドライヴ・５７７５
(72)発明者 ビリー・ビー・ブラムリー
アメリカ合衆国・カリフォルニア・９２１２１－１７１４・サン・ディエゴ・モアハウス・ドライヴ・５７７５

Ｆターム(参考) 5B017 AA01 BA01 CA01
5J104 AA08 AA12 LA02 NA02 NA12 NA37 PA07