(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4668985号
(P4668985)
(45)発行日　平成23年4月13日(2011.4.13)　(24)登録日　平成23年1月21日(2011.1.21)

(51)Int.Cl.　　F I
G09C 1/00 (2006.01)　　G09C 1/00 610A
H04L 9/06 (2006.01)　　H04L 9/00 611Z

請求項の数 5　(全 8 頁)

(21)出願番号　特願2007-512586 (P2007-512586)
(86)(22)出願日　平成17年5月11日 (2005.5.11)
(65)公表番号　特表2007-537474 (P2007-537474A)
(43)公表日　平成19年12月20日 (2007.12.20)
(86)国際出願番号　PCT/IB2005/001409
(87)国際公開番号　WO2005/109183
(87)国際公開日　平成17年11月17日 (2005.11.17)
審査請求日　平成19年1月9日 (2007.1.9)
(31)優先権主張番号　04291204.8
(32)優先日　平成16年5月11日 (2004.5.11)
(33)優先権主張国　欧州特許庁 (EP)

(73)特許権者　504326572
アクサルト・エス・アー
フランス国、９２１２０・ムドン、リユ・ドウ・ラ・ベレリー・６
(74)代理人　100062007
弁理士　川口　義雄
(74)代理人　100114188
弁理士　小野　誠
(74)代理人　100140523
弁理士　渡邊　千尋
(74)代理人　100119253
弁理士　金山　賢教
(74)代理人　100103920
弁理士　大崎　勝真

最終頁に続く

(54)【発明の名称】ホモグラフィックマスキングにより暗号アセンブリを保護する方法

(57)【特許請求の範囲】
【請求項１】
入力手段、プロセッサ、メモリ及び出力手段を有する電子アセンブリを保護する方法であって、前記方法は、前記メモリ内に格納された暗号計算手順を使用し、マスク化変数に作用する下記タイプのホモグラフィック関数ｆ
・（ｃｚ＋ｄ）が０に等しくない場合、ｆ（ｚ）＝（ａｚ＋ｂ）／（ｃｚ＋ｄ）
・ｆ（－ｄ／ｃ）＝ａ／ｃ
を使用し、
前記方法は、
前記入力手段で入力ｘを取得するステップと、
全てのｋについて、前記プロセッサが前記ホモグラフィック関数ｆの出力ｙ＝ｆ（ｘ＋ｋ）を計算し、前記計算は、ｍ＿ｉがマスクであり＋がビット毎のＸＯＲ演算であり、直接的にマスク化値ｘ＋ｍ＿ｉからマスク化値ｙ＋ｍ＿ｊへ移行するように暗号計算手順を使用し、この直接的な移行は、（ａｘ＋ｂ）／（ｃｘ＋ｄ）の形式で定義される、無限量の追加をともないＧＦ（２＾ｋ）に作用する数個の変換の合成を用い及び、二つの点を交換する変換を用いて実行されるステップと、
前記出力手段でｙを出力するステップを有する、方法。
【請求項２】
演算ｆがＩｎｖ関数であり、前記Ｉｎｖ関数は、ＡＥＳと同じく０が０へ引き渡されるＧＦ（２＾ｋ）における逆関数であることを特徴とする、請求項１に記載の方法。

【請求項３】
保護される計算プロセスがラインドールまたはＡＥＳであることを特徴とする、請求項１または２に記載の方法。
【請求項４】
入力手段、
プロセッサ、
暗号計算手順を格納したメモリ、
出力手段、を含む電子システムであって、
前記格納された暗号計算手順がマスク化変数に作用する下記タイプのホモグラフィック関数ｆ
・（ｃｚ＋ｄ）が０に等しくない場合、ｆ（ｚ）＝（ａｚ＋ｂ）／（ｃｚ＋ｄ）
・ｆ（－ｄ／ｃ）＝ａ／ｃ
を使用して前記プロセッサにより実行可能であり、
前記暗号計算手順の実行は、
全てのｋについて、ｘが入力手段により提供される入力であり、ｙ＝ｆ（ｘ＋ｋ）が前記ホモグラフィック関数ｆの出力の場合に、前記計算は、ｍ＿ｉがマスクであり＋がビット毎のＸＯＲ演算であり、直接的にマスク化値ｘ＋ｍ＿ｉからマスク化値ｙ＋ｍ＿ｊへ移行するように暗号計算手順を使用し、この直接的な移行は、（ａｘ＋ｂ）／（ｃｘ＋ｄ）の形式で定義される、無限量の追加をともないＧＦ（２＾ｋ）に作用する数個の変換の合成を用い及び、二つの点を交換する変換を用いて実行されて、計算し、出力手段で適用するように構成される、電子システム。
【請求項５】
プログラムコード命令を含むコンピュータプログラムであって、電子システムに請求項１から３のいずれか一項に記載の方法の各ステップを実行させる、コンピュータプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、秘密鍵等の秘密量を使用する暗号アルゴリズムを実施する、電子アセンブリの安全を保証する方法に関する。より厳密には、この方法は、例えば、計算の実行中に電子アセンブリの電力消費量を調べることによって秘密鍵に関する情報の入手を試みる、高次差分電力解析として知られる攻撃等、ある種の物理的攻撃に直面し脆弱でないアルゴリズムのバージョンを作ることを目的とする。
【背景技術】
【０００２】
１．１　背景
ここで検討する暗号アルゴリズムは、入力情報に従って出力情報を計算するように秘密鍵を使用する。これらのアルゴリズムには数多くの応用、例えば暗号化、復号化、署名、署名検査、認証または否認防止、その他の操作がある。数多くの応用は現在、ＤＥＳ等、より最近では２０００年以降に全世界的暗号化標準となったＡＥＳ等の、秘密鍵暗号アルゴリズムの上にそのセキュリティの基礎を置く。Ｊｏａｎ　Ｄａｅｍｅｎ，Ｖｉｎｃｅｎｔ　Ｒｉｊｍｅｎ：ＡＥＳ　ｐｒｏｐｏｓａｌ：Ｒｉｊｎｄａｅｌ（ＡＥＳ提案：ラインドール）を参照されたい。
【０００３】
最新バージョンはインターネットで入手できる、ｈｔｔｐ：／／ｃｓｒｃ．ｎｉｓｔ．ｇｏｖ／ｅｎｃｒｙｐｔｉｏｎ／ａｅｓ／ｒｉｊｎｄａｅｌ／Ｒｉｊｎｄａｅｌ．ｐｄｆ。暗号技術においてはこれらの暗号アルゴリズムが研究されており、知られている最良の攻撃に対し安全であることが証明されている。したがってこれらの暗号ソリューションの場合、セキュリティは主に使用する秘密鍵のセキュリティに左右される。残念ながら、ＰＣに蓄積されるデータ項目のセキュリティも、人間が記憶するパスワードのセキュリテ

ィも、深刻には受け止められない。したがって、スマートカード等の独立した安全なモジュールに秘密量を蓄積することが不可欠となっている。

【０００４】

１．２　問題：埋め込みアルゴリズムの保護

暗号アルゴリズムは、理想的な数学的世界の中では完璧に安全ではあるが、実世界においてはこの限りでない。スマートカードはエネルギーを放射し、電流を消費し、結果として、秘密量に依存する情報はサイクルごとにカードから逃げ出る。

【０００５】

真に安全となるためには、アルゴリズムの中間データはこの秘密量についていかなる情報をも提供してはならない。加えて、新しい攻撃が開発されている。以下の文書を参照されたい。

【０００６】

Ｐ．Ｋｏｃｈｅｒ，Ｊ．Ｊａｆｆｅ，Ｂ．Ｊｕｎ，「差分電力解析と関係する攻撃の紹介（Ｉｎｔｒｏｄｕｃｔｉｏｎ　ｔｏ　Ｄｉｆｆｅｒｅｎｔｉａｌ　Ｐｏｗｅｒ　Ａｎａｌｙｓｉｓ　ａｎｄ　Ｒｅｌａｔｅｄ　Ａｔｔａｃｋｓ）」Ｔｅｃｈｎｉｃａｌ　Ｒｅｐｏｒｔ，Ｃｒｙｐｔｏｇｒａｐｈｙ　Ｒｅｓｅａｒｃｈ　Ｉｎｃ．，１９９８。ｈｔｔｐ：／／ｗｗｗ．ｃｒｙｐｔｏｇｒａｐｈｙ．ｃｏｍ／ｄｐａ／ｔｅｃｈｎｉｃａｌ／ｉｎｄｅｘ．ｈｔｍｌから入手可能。

【０００７】

Ｔ．Ｓ．Ｍｅｓｓｅｒｇｅｓ，「二次電力解析を用いたＤＰＡ抵抗ソフトウェア攻撃（Ｕｓｉｎｇ　Ｓｅｃｏｎｄ－Ｏｒｄｅｒ　Ｐｏｗｅｒ　Ａｎａｌｙｓｉｓ　ｔｏ　Ａｔｔａｃｋ　ＤＰＡ　Ｒｅｓｉｓｔａｎｔ　ｓｏｆｔｗａｒｅ）」Ｉｎ　Ｐｒｏｃｅｅｄｉｎｇｓ　ｏｆ　ＣＨＥＳ’２０００，ＬＮＣＳ　１９６５，ｐｐ．２３８－２５１，Ｓｐｒｉｎｇｅｒ－Ｖｅｒｌａｇ，２０００。

【０００８】

これらは高次攻撃として知られている（例えば「二次ＤＰＡ」）。これは攻撃者が、暗号アルゴリズムの実行中に逃げ出る情報を二回以上にわたり組み合わせることを意味する。この種の攻撃から保護されるためには、アルゴリズムの中間データが秘密量についていかなる情報をも提供しないだけでは、もはや不十分である。攻撃に際し、秘密量に関する任意の情報を入手するように、実行中の異なる時点に入手したデータを組み合わせることもまた不可能でなければならない。

【発明の開示】

【発明が解決しようとする課題】

【０００９】

１．３　制約

問題の解決法では、ＤＰＡタイプの攻撃に対し保護を提供するばかりでなく、これを「二次ＤＰＡ」以上の攻撃にまで拡張することもまた可能でなければならない。解決法はまた、実行時間と使用するメモリの量に関し相応の制約を満たさなければならない。本発明の一目的は、実行時間とメモリとが、安全が保証されない実施に比べて、ブロックサイズにもＡＥＳ反復の数にも依存しない、本発明によって達成できる、小さな定数倍されることである。

【００１０】

本発明は、第一、第二、またはそれ以上のＤＰＡタイプ攻撃、ＳＰＡ攻撃またはその他電子的攻撃、そして他の隠れた経路を介する攻撃に対し、ＡＥＳのセキュリティを保証する。

【００１１】

本明細書の残りの部分では、ＡＥＳアルゴリズムに特に適し、ただし他の暗号アルゴリズムにも適用できる、一般的な解決法を説明する。この問題に対する公知の解決策はどれもこれまで、その性能水準とそのメモリ使用の点で批判されてきたし、文献で公表されている攻撃にさらされてきた。

【課題を解決するための手段】
【００１２】
本発明は、メモリに蓄積された暗号計算プロセスを実施する、プロセッサとメモリとを備える電子システムの安全を保証する方法に関し、同暗号計算プロセスは秘密量ｋを使用し、且つ下記タイプのホモグラフィック関数ｆを使用する。
・（ｃｚ＋ｄ）が０に等しくない場合、ｆ（ｚ）＝（ａｚ＋ｂ）／（ｃｚ＋ｄ）
・ｆ（－ｄ／ｃ）＝ａ／ｃ
【００１３】
関数ｆはマスク化変数に作用する。本方法は、任意のｋについてｘが関数ｆの入力であって且つｙ＝ｆ（ｘ＋ｋ）が関数ｆの出力の場合に、直接的にマスク化値ｘ＋ｍ＿ｉ（ＸＯＲタイプの加法マスキング）からマスク化値ｙ＋ｍ＿ｊへとなるように、（ａｘ＋ｂ）／（ｃｘ＋ｄ）と定義される、無限量の加算をともないＧＦ（２＾ｋ）に作用する数個の変換と、二つの点を交換する変換との合成を用いて、この演算を実行することにある。
【００１４】
本発明はまた、この方法を実施するシステムに関する。
【００１５】
本発明による方法の目的は、秘密鍵を用いる暗号計算プロセスを実施する、電子システムと、例えばスマートカード等の埋め込みシステムとの安全を保証することにある。電子システムはプロセッサとメモリとを備える。暗号計算プロセスは、このシステムの、例えばＲＯＭタイプのメモリにインストールされる。このシステムのプロセッサは、例えばＥ２ＰＲＯＭタイプのメモリの秘密エリアに蓄積された秘密鍵を用いて、計算プロセスを実行する。
【００１６】
本発明による方法は、ホモグラフィック保護を提供することである。
【発明を実施するための最良の形態】
【００１７】
まず、保護の一般原理を説明する。
【００１８】
２．１　分解原則
各々の暗号システムは、加算、ＸＯＲ等、いくつかの基本的演算に分解できる。
【００１９】
ＡＥＳの場合、演算は二つのカテゴリに分けることができる。
－従来の加法マスキングによって容易に保護される「線形」演算。これは公知であって、本発明の主題ではない。
－線形演算を除いた場合、ただひとつの演算が、すなわち０へマップされる０をともなう、有限体ＧＦ（２５６）等における逆演算から導き出されるラインドール（Ｒｉｊｎｄｅａｌ）Ｉｎｖ演算が残る。
本発明者らの関心は専らＩｎｖ演算を保護することにある。
説明される解決法は、他の同様の演算にも適用される。
【００２０】
２．２　準備
Ｋを有限体とする。ＡＥＳ　Ｋ＝Ｇ（２５６）の場合。
Ｋにおける加算と乗算との実施に相当する、Ｋの実施がいくつか存在すると仮定する。例えば［ＡＥＳ］に定義されたものである。
【００２１】
Ｉｎｖを修正されたラインドール逆関数［ＡＥＳ］と仮定する、すなわち、
・ｘが非ヌルである場合のＫにおいて、Ｉｎｖ（ｘ）＝１／ｘ
・Ｉｎｖ（０）＝０
【００２２】
Ｋに対する無限量として知られる点の加算によって、Ｋ’を定義する。

よってＫ’＝Ｋ∪ｏｏ
【００２３】
以下の演算としてＩｎｖ’を定義する。
・ｘが非ヌルであって且つｏｏに等しくない場合のＫにおいて、Ｉｎｖ’（ｘ）＝１／ｘ
・Ｉｎｖ’（０）＝ｏｏ
・Ｉｎｖ’（ｏｏ）＝０
【００２４】
本発明は、Ｉｎｖを計算するにあたって、本発明者らがＩｎｖ’の、および０とｏｏとを交換する演算の合成を適用できるとみなす。
【００２５】
点ａおよびｂを交換するＫ’において、Ｅ［ａ，ｂ］をＫ’の演算とする。
・ｘがａにもｂにも等しくない場合、Ｅ［ａ，ｂ］（ｘ）＝ｘ
・Ｅ［ａ，ｂ］（ａ）＝ｂ
・Ｅ［ａ，ｂ］（ｂ）＝ａ
【００２６】
２．３　演算Ｉｎｖをいかに表現するか
Ｋにおいて任意のｘの場合（Ｉｎｖはいかなる Ｋ’においても定義されない）：
Ｉｎｖ（ｘ）＝Ｉｎｖ’（Ｅ［０，ｏｏ］（ｘ））
【００２７】
保護原則は次のとおりである：Ｉｎｖ’は、適度のサイズの、合成により安定的な群の一部であって、Ｉｎｖの場合と異なる。結果的に、Ｉｎｖの場合には存在しえない保護を、Ｉｎｖ’の場合に果たすことができる。
【００２８】
この群は以下の関数の集合として定義される。
【００２９】
Ｋの要素の任意の４－ｕｐｌｅｔ（ａ，ｂ，ｃ，ｄ）においてａｃ＜＞ｂｄの場合、本発明者らは次のとおりに定義する。
関数ＨＯＭ［ａ，ｂ，ｃ，ｄ］＝以下の関数：
・（ｃｘ＋ｄ）が０に等しくない場合のＫにおいて、ＨＯＭ［ａ，ｂ，ｃ，ｄ］（ｘ）＝（ａｘ＋ｂ）／（ｃｘ＋ｄ）
・ＨＯＭ［ａ，ｂ，ｃ，ｄ］（－ｄ／ｃ）＝ｏｏ
・ＨＯＭ［ａ，ｂ，ｃ，ｄ］（ｏｏ）＝ａ／ｃ
【００３０】
本発明者らはＩｎｖを実施するように、集合ＫにてＩｎｖに一致する以下の関数Ｋ’－＞Ｋ’を書く。
Ｉｎｖ’ｏＥ［０，ｏｏ］
【００３１】
符号「ｏ」は、通常の関数の合成を表わす。
【００３２】
本発明者らは次に、ホモグラフィック関数の積としてＩｎｖ’を書く。
Ｉｎｖ’＝Ｆ＿１ｏＦ＿２ｏ．．．ｏＦ＿ｎｏＧ＿１ｏ．．Ｇ＿ｎ
【００３３】
関数Ｆ＿ｉおよびＧ＿ｉの各々は、ＨＯＭ［ａ，ｂ，ｃ，ｄ］の形をとる。
【００３４】
Ｉｎｖ’は一群に属するため、この分解は恣意的に遂行される。例えば、２＊ｎ－１個の関数を無作為に選び、欠けている関数を再計算し、合成することによりＩｎｖ’を作ることができる。
【００３５】
本発明者らは次に、ＫにてＩｎｖに一致する以下の関数Ｋ’－＞Ｋ’を得る。

Ｆ＿１ｏＦ＿２ｏ．．．ｏＦ＿ｎｏＧ＿１ｏ．．Ｇ＿ｎｏＥ［０，ｏｏ］

【００３６】

ただし、Ｋ’においてこれらの関数はどれも全単射性であるため、この関数が下記に等しくなるよう二つの点ａおよびｂを計算できる。

Ｆ＿１ｏＦ＿２ｏ．．．ｏＦ＿ｎｏＥ［ａ，ｂ］ｏＧ＿１ｏ．．Ｇ＿ｎ

【００３７】

これらの点は、ａ＝Ｇ＿１（．．．Ｇ＿ｎ（０））およびｂ＝Ｇ＿１（．．．Ｇ＿ｎ（ｏｏ））である。

【００３８】

本発明における保護は次のとおりに実施されるであろう。

１．Ｆ＿１，Ｆ＿２，．．．，Ｆ＿ｎ，Ｇ＿１，Ｇ＿ｎを生成する。各々はＫの４要素、すなわちラインドール／ＡＥＳにおける４バイトによって記述される。

２．本発明者らはａおよびｂを計算する。

３．次に本発明者らはこの一連の演算を適用することにより、Ｉｎｖを計算する。

４．ＡＥＳには数個のＩｎｖがある。１－３に定義する、実施される一連の演算は、ある一つの計算から別の計算にかけて異なってよい。

【００３９】

２．４　演算Ｉｎｖをいかに保護するか

安全なＡＥＳ実施において、ｙ＝Ｉｎｖ（ｘ）はｘから計算せず、代わりにマスク化値ｘ＋ｍ＿ｉから直接的に計算することにより、情報を提供する中間値ｘおよびｙを使用せず、ｙ＋ｍ＿ｊを直接的に得る。従って本発明者らは以下の関数を計算しなければならない。

ｙ＝Ｉｎｖ（ｘ＋ｍ＿ｉ）＋ｍ＿ｊ

【００４０】

Ｉｎｖと同じく、この関数は、ＨＯＭ［ａ，ｂ，ｃ，ｄ］の形をとる基礎的演算の組み合わせとして数多くのあり方を認めることができ、二つの点を交換する。

【００４１】

さらに進むことも推奨される。Ｋ＿ｉをＡＥＳの中間鍵とする。演算ｘ｜－＞Ｉｎｖ（ｘ＋Ｋ＿ｉ）は同じ仕方で直接的に保護できる。二点を交換した後、この演算は任意のＫにて群内の特定のＨＯＭ［ａ，ｂ，ｃ，ｄ］：Ｋ’－＞Ｋ’に等しく、これはＩｎｖの場合と同じ仕方で分解できる。本発明者らは加法マスクによって保護される実施において、以下の関数を分解する必要がある。

ｘ｜－＞Ｉｎｖ（ｘ＋Ｋ＿ｉ＋ｍ＿ｉ）＋ｍ＿ｊ

【００４２】

これは同じ仕方で遂行される。

【００４３】

２．５　改善

本発明者らは一つの演算の代わりに、数個の演算Ｅ［ａ，ｂ］を使用できる。

【００４４】

各々の演算ＨＯＭ［ａ，ｂ，ｃ，ｄ］について、ａが０または１に等しいと仮定できることは明白である。

【００４５】

加法または乗法マスキングが使用されるときに実施を保護するように同じ方法を使用することもできるが、これは推奨されない。これらのマスキングは全単射性ではない、または特定の点を固定する、例えば乗法マスキングは０をマスクしない。ホモグラフィックマスキングはどのタイプであれ常に全単射性となるが、２５７値のうち一つを蓄積する必要があり、これはあまり現実的でない。すなわち１バイトで蓄積できない。

【００４６】

本明細書は、ＡＥＳ保護実施の全体を説明するものではない。

【００４７】

その目的は、保護することが最も困難な非線形成分をいかに保護するかを説明することである。アセンブリの保護は、広く知られる他の従来の保護を含んでよく、且つ含まなければならない。

【００４８】

したがって本発明は、説明した特別な実現形態において、少なくとも関数Ｉｎｖ（ＡＥＳと同じく０が０へマップされるＧＦ（２＾ｋ）における逆関数）を用いる暗号計算プロセスを実施する、アセンブリを保護する方法であり、計算の中間変数ｘは加法マスキングｘ＋ｍ＿ｉにより処理され、ｍ＿ｉはマスクであって且つ＋はＸＯＲ演算子である方法に関し、任意のｋについてｘが入力であって且つｙ＝ｆ（ｘ＋ｋ）の場合に、中間値を露呈することなく直接的にマスク化値ｘ＋ｍ＿ｉからマスク化値ｙ＋ｍ＿ｊへとなるように、この演算が、（ａｘ＋ｂ）／（ｃｘ＋ｄ）の形に定義される、無限量の加算をともないＧＦ（２＾Ｋ）に作用する数個の変換と、二つの点を交換する変換との合成を用いて遂行されることを特徴とする。

【００４９】

これはまた、関数Ｉｎｖ（ＡＥＳと同じく０が０へマップされるＧＦ（２＾ｋ）における逆関数）を用いる暗号計算プロセスを実施する、蓄積手段を備えるシステムであり、計算の中間変数ｘは加法マスキングｘ＋ｍ＿ｉにより処理され、ｍ＿ｉはマスクであって且つ＋はＸＯＲ演算子であるシステムに関し、任意のｋについてｘが入力であって且つｙ＝Ｉｎｖ（ｘ＋ｋ）の場合に、中間値を露呈することなく直接的にマスク化値ｘ＋ｍ＿ｉからマスク化値ｙ＋ｍ＿ｊへとなるように、本発明者らがこの演算を、（ａｘ＋ｂ）／（ｃｘ＋ｄ）の形に定義される、無限量の加算をともないＧＦ（２＾Ｋ）に作用する数個の変換と、二つの点を交換する変換との合成とみなすことを特徴とする。

フロントページの続き

(74)代理人 100124855
弁理士 坪倉 道明
(72)発明者 クールトワ，ニコラ
フランス国、エフ－７８４３１・ルーブシエンヌ、ベー・ペー・４５、リユ・ドウ・ラ・プランセス、３６－３８、デイレクシヨン・プロプリエテ・アンテレクチユエル、アクサルト・エス・アー

審査官 中里 裕正

(56)参考文献 特開２００２－３６６０２９（ＪＰ，Ａ）
Trichina E., et al., Simplified Adaptive Multiplicative Masking for AES, Lecture Notes in Computer Science, ２００２年, Vol.2523, p.187-197
Golic J., et al., Multiplicative Masking and Power Analysis of AES, Lecture Notes in Computer Science, ２００２年, Vol.2523, p.198-212

(58)調査した分野(Int.Cl., ＤＢ名)
G09C 1/00
H04L 9/06
JSTPlus/JMEDPlus/JST7580(JDreamII)