(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6267207号
(P6267207)
(45)発行日　平成30年1月24日(2018.1.24)　(24)登録日　平成30年1月5日(2018.1.5)

(51)Int.Cl.　F I
H04L 9/10 (2006.01)　H04L 9/00 621A

請求項の数 16 (全 24 頁)

(21)出願番号	特願2015-534947 (P2015-534947)
(86)(22)出願日	平成25年9月10日 (2013.9.10)
(65)公表番号	特表2015-532549 (P2015-532549A)
(43)公表日	平成27年11月9日 (2015.11.9)
(86)国際出願番号	PCT/EP2013/068746
(87)国際公開番号	WO2014/053286
(87)国際公開日	平成26年4月10日 (2014.4.10)
審査請求日	平成28年8月29日 (2016.8.29)
(31)優先権主張番号	12187213.9
(32)優先日	平成24年10月4日 (2012.10.4)
(33)優先権主張国	欧州特許庁 (EP)
(31)優先権主張番号	12194713.9
(32)優先日	平成24年11月29日 (2012.11.29)
(33)優先権主張国	欧州特許庁 (EP)

(73)特許権者　510273499
イントリンシツク・イー・デー・ベー・ベー
オランダ国、エン・エル－５６５６・アー・エー・アイントホーフエン、ハイ・テツク・キヤンパス・９
(74)代理人　110001173
特許業務法人川口國際特許事務所
(72)発明者　ファン・デル・スライス，エリック
オランダ国、３５８２・フェー・イクス・ユトレヒト、イナ・ブーディエル－バッケルホフ・８８
(72)発明者　ファン・フルスト，マールテン
オランダ国、５０６１・アー・ベー・オーイステルウェイク、ベルケンラーン・９
最終頁に続く

(54)【発明の名称】物理的クローン不能関数として使用されるメモリから暗号化キーを生成するためのシステム

(57)【特許請求の範囲】
【請求項１】
暗号化キーを生成するための電子システム（１００、２００）であって、電子システムが、
物理的クローン不能関数として使用されるメモリ（１１０）であって、メモリは書き込み可能で、揮発性であり、メモリに電源投入したたびに、メモリはメモリの少なくとも部分的にランダムな物理的特性に応じたメモリコンテンツに確定するように構成されており、メモリはメモリインタフェース（１２０）を通してアクセス可能であるメモリと、
メモリが確定したメモリコンテンツから暗号化キーを導出するように構成されたキー導出ユニット（１５０、２３０）とを備え、
暗号化キーを生成するための電子システムがさらに、
メモリインタフェースを通してメモリ、およびキー導出ユニットに接続されたメモリ読み出しユニット（２２０）を備え、メモリ読み出しユニットが
スクランブルされた順序でメモリインタフェースを介してメモリコンテンツを取り出すためのアドレススクランブラー（１４０）と、
メモリを暗号化するための暗号化ユニット（２４０）であって、メモリインタフェースからスクランブルされた順序でメモリコンテンツを受信し、受信したメモリコンテンツを暗号化し、暗号化されたメモリコンテンツをメモリに書き戻すように構成されている暗号化ユニット（２４０）と、
メモリを復号するための復号ユニット（２５０）であって、メモリインタフェースか

ら事前に定義されたさらなる順序で暗号化されたメモリコンテンツを受信し、受信した暗号化されたメモリコンテンツを復号して事前に定義されたさらなる順序でメモリコンテンツを取得するように構成されている復号ユニット（２５０）と
を備える、電子システム。

【請求項２】
不揮発性のヘルパーデータメモリ（２８５）を備え、ヘルパーデータメモリ（２８５）が、物理的クローン不能関数として使用されるメモリのために構築されたヘルパーデータを記憶し、さらに、
訂正可能なビットストリングを確立するためのコンバイナ（２８０）を備え、訂正可能なビットストリングはエラー訂正コードの訂正可能な近傍にあり、コンバイナ（２８０）が、ヘルパーデータメモリからヘルパーデータを取り出して、取り出したヘルパーデータと、メモリ読み出しユニットからされた事前に定義されたさらなる順序でメモリが確定したメモリコンテンツとを組み合わせるように構成され、さらに、
エラー訂正アルゴリズムを使用して訂正可能なビットストリングからエラー訂正コードのコードワードを確立するように構成されたエラーコレクタ（２９０）を備える、請求項１に記載の電子システム。

【請求項３】
暗号化ユニットが暗号化されたメモリコンテンツをスクランブルされた順序でメモリに書き戻すように構成され、それによって、メモリコンテンツを暗号化されたメモリコンテンツで上書きする、請求項１または２に記載の電子システム。

【請求項４】
スクランブルされた順序が、読み出しユニットの各電源投入後に少なくとも部分的にランダムに生成されるシードから導出される、請求項１から３のいずれか一項に記載の電子システム。

【請求項５】
読み出しユニットが新しいシードを取得するためにシードに適用されるように構成された暗号化一方向性関数を備え、アドレススクランブラーが新しくスクランブルされた順序でメモリインタフェースを介してメモリコンテンツの再フェッチのための新しいシードから新しくスクランブルされた順序を導出するように構成された、請求項４に記載の電子システム。

【請求項６】
読み出しユニットが、さらなるメモリに電源投入したたびに、第２のメモリがノイズのあるメモリコンテンツに確定するように構成されたさらなる揮発性メモリを備え、シードが、第２のメモリが確定したメモリコンテンツから導出される、請求項４および５のいずれか一項に記載の電子システム。

【請求項７】
難読化ユニットを備え、難読化ユニットがメモリへいくつか追加で難読化アクセスするように構成され、難読化アクセスがメモリからの光子放出とメモリが電源投入時に確定したメモリコンテンツ間の相関を低減化するように構成された、請求項１から６のいずれか一項に記載の電子システム。

【請求項８】
難読化ユニットがメモリのメモリ位置への難読化アクセスを行うように構成され、難読化アクセスが、暗号化ユニットが暗号化されたメモリコンテンツをメモリ位置に書き戻した後に行われる、請求項７に記載の電子システム。

【請求項９】
物理的クローン不能関数として使用されるメモリがＳＲＡＭメモリである、請求項１から８のいずれか一項に記載の電子システム。

【請求項１０】
物理的クローン不能関数として使用されるメモリが揮発性ＦＰＧＡメモリである、請求項１から９のいずれか一項に記載の電子システム。

【請求項１１】
請求項１から１０のいずれか一項に記載の暗号化キーを生成するための電子システムを含む集積回路。
【請求項１２】
集積回路がスマートカード、ＡＳＳＰ、ＤＳＰ、アプリケーションプロセッサ、ＳＩＭ、およびＮＦＣチップのいずれか一つである、請求項１から１１のいずれか一項に記載の暗号化キーを生成するための電子システムを含む集積回路。
【請求項１３】
暗号化キーを生成するための電子システム（１００、２００）において、事前に定義されたさらなる順序でメモリを安全に読み出すための電子メモリ読み出しユニットであって、システムが、
物理的クローン不能関数として使用されるメモリ（１１０）であって、メモリは書き込み可能で、揮発性であり、メモリに電源投入したたびに、メモリはメモリの少なくとも部分的にランダムな物理的特性に応じたメモリコンテンツに確定するように構成されており、メモリはメモリインタフェース（１２０）を通してアクセス可能であるメモリと、
メモリが確定したメモリコンテンツから暗号化キーを導出するように構成されたキー導出ユニット（１５０、２３０）とを備え、電子メモリ読み出しユニットはメモリインタフェースを通してメモリ、およびキー導出ユニットに接続可能であり、電子メモリ読み出しユニットが、
スクランブルされた順序でメモリインタフェースを介してメモリコンテンツを取り出すように構成されたアドレススクランブラーと、
メモリを暗号化するための暗号化ユニットとを備え、暗号化ユニットが、メモリインタフェースからスクランブルされた順序でメモリコンテンツを受信し、受信したメモリコンテンツを暗号化し、暗号化されたメモリコンテンツをメモリに書き戻すように構成され、電子メモリ読み出しユニットがさらに、
メモリを復号するための復号ユニットを備え、復号ユニットが、メモリインタフェースから事前に定義されたさらなる順序で暗号化されたメモリコンテンツを受信し、受信した暗号化されたメモリコンテンツを復号して事前に定義されたさらなる順序でメモリコンテンツを取得するように構成された、電子メモリ読み出しユニット。
【請求項１４】
暗号化キーを生成するための方法（４００）であって、方法が、
物理的クローン不能関数として使用されるメモリの電源投入するステップ（４１０）を含み、メモリが書き込み可能で揮発性であり、方法がさらに、
メモリが、メモリの少なくとも部分的にランダムな物理的特性に応じたメモリコンテンツに確定できるようにするステップ（４２０）、
スクランブルされた順序でメモリインタフェースを介してメモリコンテンツを取り出すステップ（４３０）、および
メモリが確定したメモリコンテンツから暗号化キーを導出するステップ（４８０）を含み、
メモリの暗号化が、
メモリインタフェースからスクランブルされた順序でメモリコンテンツを受信するステップ（４３０）、
受信したメモリコンテンツを暗号化するステップ（４４０）、および
暗号化されたメモリコンテンツをメモリに書き戻すステップ（４５０）によって行われること、
メモリの復号が、
メモリインタフェースから事前に定義されたさらなる順序で暗号化されたメモリコンテンツを受信するステップ（４６０）、および
事前に定義されたさらなる順序でメモリコンテンツを取得するために受信した暗号化されたメモリコンテンツを復号するステップ（４７０）によって行われることを含む方法。

【請求項１５】
コンピュータプログラムがコンピュータで実行される時に、請求項１４に記載のすべてのステップを行うように適合されたコンピュータプログラムコード手段を含む、コンピュータプログラム。
【請求項１６】
コンピュータ読み取り可能媒体上で具現化される、請求項１５に記載の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、暗号化キーを生成するための電子システムに関するもので、本システムは物理的クローン不能関数として使用されるメモリであって、本メモリは書き込み可能で、揮発性であり、メモリに電源投入したたびに、メモリがメモリの少なくとも部分的にランダムな物理的特性に応じたメモリコンテンツに確定するように構成されており、本メモリはメモリインタフェースを通してアクセス可能であるメモリ、およびメモリが確定したメモリコンテンツから暗号化キーを導出するように構成されたキー導出ユニットを備えている。
【背景技術】
【０００２】
物理的クローン不能関数（ＰＵＦ）は、安全なメモリへのキー、識別子などの保存を含む、多くの形の安全な識別情報の利点となる代替物であることが証明されている。
【０００３】
物理的クローン不能関数は、製造上の変動を利用してデジタル識別子を導出する。こうして、デジタル識別子は物理的な媒体と関係されている。物理的クローン不能関数はランダムプロセスの変動に応じて異なるので、ＰＵＦを作成するのは容易であるが、特定の事前に定義された識別子を生み出すＰＵＦを作成することは、不可能でないにしても非常に困難である。製造上の変動は、メモリ素子の様々な物理的特性という結果になる。例えば、物理的特性には、すなわち、ドーピング濃度、酸化膜厚、チャネル長、（例えば、金属層の）構造上の幅、寄生（例えば、抵抗、静電容量）などが含まれる。デジタル回路設計が何度も製造される時、これらの物理的特性はわずかに変化し、同時にそのため、ＩＣ素子の動作、例えば、メモリ素子などが一部の状態で様々に動作するようになる。例えば、起動時の動作は、物理的特性の製造上の変動で決定される。
【０００４】
ＰＵＦの便利な選択は、揮発性メモリ、具体的には、フリップフロップベースのメモリ、より具体的には、スタティックランダムアクセスメモリ（ＳＲＡＭ）である。そのようなメモリは評価が容易で、製造コストが低価である。ＳＲＡＭベースのＰＵＦは、ＳＲＡＭ　ＰＵＦと呼ばれている。ＳＲＡＭは、電源投入後に、オンビットとオフビットのランダムパターンで満たされる。パターンは、ＳＲＡＭが次回、電源投入される場合にそれ自体を正確に繰り返さない可能性があるが、そのような２つのパターン間の相違は通常、当該状態でのビットの半数をかなり下回るものである。同じＳＲＡＭのメモリ電源投入コンテンツ間の相違は通常、異なるＳＲＡＭのメモリ電源投入コンテンツ間の相違よりもかなりわずかなものである。
【０００５】
ＰＵＦは、同じチャレンジが２度評価される時、正確に同じ結果を生み出さない可能性があるので、ファジィエクストラクタとも呼ばれる、いわゆるヘルパーデータアルゴリズムが使用され、導出される場合には必ず、キーが同じになることが確保され得る。ノイズの多い測定値から再現可能な値を構築するためにヘルパーデータを使用する１つのやり方が、例えば、ＷＯ２００６／１２９２４２「ヘルパーデータシステムでのテンプレートの更新（Ｔｅｍｐｌａｔｅ　Ｒｅｎｅｗａｌ　ｉｎ　Ｈｅｌｐｅｒ　Ｄａｔａ　Ｓｙｓｔｅｍｓ）」などで記述されている。

【０００６】
特定のＳＲＡＭ　ＰＵＦでのＰＵＦの一適用例は、電子回路で暗号化キーを導出することである。電子回路には通常、集積回路（ＩＣ）および／またはプログラム可能論理が含まれる。
【０００７】
ＰＵＦの１つの利点は、それらが耐タンパ特性を生来所持していることである。ＰＵＦがないと、暗号化キーは攻撃者により、キーが従来保存されている不揮発性メモリに対する物理的攻撃を展開することによって回復され得る。例えば、攻撃者はメモリを開き、当該コンテンツを綿密に調べ得る。ＰＵＦの使用によって、このタイプの攻撃はかなり困難になる。その理由は、ＰＵＦを開いても通常、それが妨害されること、つまり動的コンテンツのＳＲＡＭの綿密な調査は、埋め込み不揮発性メモリの綿密な調査よりも非常に困難であることである。したがって、攻撃者が自分の綿密な調査から習得する情報は、暗号化キーの作成に使用されたインタラクションに関係がないものである。これによって、攻撃者が物理的攻撃を使用してキーを見つけることはより困難になる。
【０００８】
あいにく、侵入性の物理的攻撃は、攻撃者がＰＵＦの内部の状態の少なくとも一部の情報を取得し得る攻撃ベクトルのみではない。いわゆるサイドチャネルでも情報がリークされ得る。サイドチャネルとは、システム内部で発生する物理的現象に関連したシステムの情報ソースで、システム外部から観察され得、少なくともある程度、意図された、観察可能な、出入力動作だけではなく、システムの内部の動作および／または状態に相関した情報を明らかにするものである。
【０００９】
電力消費、時間消費、および電磁放射は、暗号化システムに関連するサイドチャネルの例である。例えば、システムが暗号化キーを使用する間にモニタされる暗号化システムの電力消費は、ある程度、キーに相関され得る。暗号化キーを内密に保つことが最も重要なので、当該キーに相関された情報が少しでもリークすることは問題である。
【００１０】
「物理システムに応じて暗号化キーを確立するためのシステム（Ｓｙｓｔｅｍ　ｆｏｒ　ｅｓｔａｂｌｉｓｈｉｎｇ　ａ　ｃｒｙｐｔｏｇｒａｐｈｉｃ　ｋｅｙ　ｄｅｐｅｎｄｉｎｇ　ｏｎ　ａ　ｐｈｙｓｉｃａｌ　ｓｙｓｔｅｍ）」と題する、ＷＯ２０１０／１００１１５として公開された、ＷＯ２０１０／０５１６３１では、ＰＵＦからのキーの導出のエラー訂正部分の間、すなわち、ヘルパーデータアルゴリズムの実行の間に発生するサイドチャネルのリークを低減化するための解決策が開示されている。エラー訂正はサイドチャネルのリークを回避するためにとりわけ重要な段階で、その理由は、それが非線形相関を導入している、機密データを複数回、処理しているからである。その上、エラー訂正がソフトウェアに実装されている場合、リークは増加する。
【先行技術文献】
【特許文献】
【００１１】
【特許文献１】国際公開第２００６／１２９２４２号
【特許文献２】国際公開第２０１０／１０００１５号
【特許文献３】国際公開第２０１０／０５１６３１号
【発明の概要】
【発明が解決しようとする課題】
【００１２】
エラー訂正の間に発生し得るサイドチャネルのリークが対処されると、サイドチャネルのリークのより小さなソースが残るということが判明している。これらのより小さなソースはより精巧な測定を必要とし、測定に成功しても、比較的わずかな機密情報のみを提供するが、それにもかかわらず、秘密キーの導出間に発生し得るサイドチャネルの他のソースに対処したいという要望がある。

【００１３】
ＳＲＡＭ　ＰＵＦなどのメモリベースのＰＵＦの電力消費が測定される時、１つのサイドチャネルがメモリの読み出しプロセス間に発生する。例えば、ＰＵＦの制御ブロックがメモリインタフェースを通じて接続されるメモリの起動の値（電源投入時のメモリコンテンツ）を読み出す間である。メモリインタフェースの電力消費は転送されるデータワードのハミング重みに依存する。ＳＲＡＭメモリコンテンツの読み出しによって、電力または電磁解析またさらに光子放出攻撃を通じて読まれるデータバイトのハミング重みの情報がリークし得る。
【００１４】
ハミング重みと電力消費の間が完全に対応していると想定する場合、８ビット幅のメモリインタフェースのデータリークは、理論上、バイトごとに２．５４ビット位、すなわち、３０％に上ることになる。実際のところ、対応関係は完全よりかなり少なく、したがって情報のリークは非常にわずかであるが、メモリベースのＰＵＦのメモリインタフェースにおいてサイドチャネルのリークを低減化する必要がある。理論上、ビットあたりの最大リークは、メモリインタフェースがより広くなるにつれ急速に低下する。例えば、ビットあたりの理論上のリークは、メモリインタフェースが１６ビットに増大する場合には、ほとんど半減される。しかし、メモリインタフェースがよりコスト高になり、より電力を消費すると、一部の環境では使用できなくなる。
【００１５】
光子放出攻撃は、集積回路で状態が変わる時に放出され得る数個の光子により形成される光サイドチャネルを利用する。
【課題を解決するための手段】
【００１６】
暗号化キーを生成するための電子システムが提供されている。本システムは物理的クローン不能関数として使用されるメモリであって、本メモリは書き込み可能で、揮発性であり、メモリに電源投入したたびに、メモリがメモリの少なくとも部分的にランダムな物理的特性に応じたメモリコンテンツに確定するように構成されており、本メモリはメモリインタフェースを通してアクセス可能であるメモリ、およびメモリが確定したメモリコンテンツから暗号化キーを導出するように構成されたキー導出ユニットを備えている。
【００１７】
暗号化キーを生成するための電子システムはさらに、メモリインタフェースを通してメモリに、そしてキー導出ユニットに接続されたメモリ読み出しユニットを備え、本メモリ読み出しユニットはスクランブルされた順序でメモリインタフェースによりメモリコンテンツを取り出すためのアドレススクランブラーを備えている。
【００１８】
メモリコンテンツはスクランブルされた順序でメモリインタフェースを介し転送されるので、サイドチャネルのリークが低減化される。データワード自体のハミング重みはなおリークされるが、ハミング重みとメモリコンテンツの特定のデータワード間の対応関係、それは失う。メモリコンテンツがメモリコンテンツを構成する個々のデータワードのサイズに比べると大きい場合、サイドチャネルのリークでの低減化も大きくなる。これは、この場合、測定されたハミング重みが属し得る可能性のある多くのデータワードがあるからである。これがとりわけ都合の良いのは、潜在的リークはデータワードが小さい場合（８ビット以下）にはより大きくなるからである。１０２４ビットのメモリコンテンツ以上に対するサイドチャネルのリークは無視できると推定される。
【００１９】
ハミング重みに依存する電力の相違からリークされる情報量を低減化することを試みる多くの潜在的な対応策は不適当であることに留意されたい。例えば、いわゆる定重み符号は対応策として提案されている。定重み符号では、すべての機密データは、通常、データワードのビットサイズの半分に等しい、固定されたハミング重みを有するデータワードに符号化される。この対応策はハミング重みに依存する電力の相違を低減化するが、たとえ

ば安全なメモリにキーを記憶する場合、メモリベースのＰＵＦには不適当である。メモリコンテンツを決定する物理的な特性がランダムなので、すなわち、メモリコンテンツの符号化の特定のタイプを定めることができない。

【００２０】

暗号化キーを生成するための電子システムは、スマートカードなどのモバイルコンピューティングデバイス、携帯電話などのモバイル通信デバイス、タブレットなどに含まれ得る。メモリコンテンツから導出されたキーは、システムがキーの所持を有すると判定するチャレンジ応答プロトコルで使用され得る。キーは、例えば、暗号化記憶装置または通信において、機密性保護のために使用され得る。導出されたキーは対称キーであり得るが、例えば、非対称キーを見つけるためのシードとしてキーを使用することによって、非対称の公開／秘密キーのペアであってもよい。

【００２１】

暗号化キーを生成するための電子システムは、集積回路に含まれ得る。例えば、集積回路はＡＳＳＰ、すなわち、広範囲な市場に適用可能な特定の機能を実装する特殊な集積回路であり得る。ＡＳＳＰは、例えば、オーディオ／ビデオコーデックの実装に使用される。暗号化キーを使用する暗号化関数をサポートするＡＳＳＰの安全性は、本明細書に記述されている暗号化キーを生成するための電子システムで暗号化キーを生成することによって改善される。

【００２２】

例えば、集積回路はＤＳＰとし得る。これによって、例えば、ＰＵＦ生成されたキーまたはそれから導出されたものによる暗号化を使用するアプリケーションを改善し、例えば、ストリーミングコンテンツを保護すると同時に、キーを安全に保持する。

【００２３】

ＰＵＦ生成されたキーは認証および／または安全な通信チャネルの確立のための信頼のルートとして使用され得る。後者は、例えば、暗号化キーを生成するための電子システムを備えるＮＦＣチップにとって重要になり得る。

【００２４】

メモリは物理的クローン不能関数として使用される。メモリは、２つの安定状態に構成され得る、複数のバイナリ記憶素子を備え得る。起動時、各バイナリ記憶素子は２つの安定状態の１つに確定する。どちらの状態に確定するかは、記憶素子の精密な製造、例えば、ドーピングレベルにより大半が決まるが、素子が確定する状態はさらに、偶然変動にも影響される。例えば、メモリは複数のフリップフロップ、特に、Ｄタイプのフリップフロップを備え得るもので、特に、メモリはＳＲＡＭメモリであり得る。メモリの電源投入コンテンツは、ノイズおよび摂動の影響を受けやすい。一部のＦＰＧＡのものは、物理的クローン不能関数として使用されるメモリとして使用され得る初期化されていないＳＲＡＭブロックを備えている。

【００２５】

メモリインタフェースはアドレスチャネルおよびデータチャネルを備え得る。メモリへのスクランブルされたアクセスは、例えば、メモリアドレスがアドレスチャネルに置かれる前に、メモリアドレスを暗号化するなどのスクランブルによって達成され得る。その場合、異なるキーを選択すると、異なるスクランブル順序が取得され得る。好適には、スクランブル順序が攻撃者に知られず、すなわち、秘密であり、より好適には、スクランブル順序がデバイスで一意であり、例えば、製造時に選択されており、より好適には、スクランブル順序がシステム自体で頻繁に変更され、さらにより好適には、スクランブル順序がメモリ読み出しユニットの起動時に少なくとも一部がランダムに決定されることである。

【００２６】

電力解析、電磁解析、および光子放出攻撃では通常、優れた信号対雑音比を得るために多くの繰り返された測定（追跡）を必要とする。そして統計的解析がその繰り返された測定に対して行われる。読み出し順序をランダム化し、またさらに定期的に読み出し順序を変更することによって、この解析が妨害される。

【００２７】
キー導出ユニットは、メモリが確定したメモリコンテンツから暗号化キーを導出するように構成されている。導入ユニットはメモリコンテンツからノイズを除去する。例えば、これは不揮発性ヘルパーデータメモリで、物理的クローン不能関数として使用されるメモリのために構築されたヘルパーデータを記憶するヘルパーデータメモリ、訂正可能なビットストリングを確立するためのコンバイナで、本訂正可能なビットストリングがエラー訂正コードの訂正可能な近傍にあり、ヘルパーデータメモリからヘルパーデータを取り出し、メモリが確定したメモリコンテンツと取り出されたヘルパーデータを組み合わせるように構成されたコンバイナ、およびエラー訂正アルゴリズムを使用して訂正可能なビットストリングからエラー訂正コードのコードワードを確立するように構成されたエラーコレクタを備えている。

【００２８】
ヘルパーデータはメモリの１つの特定な物理的具体化に対して構築される。その理由は、様々なメモリチップのメモリの起動コンテンツは異なり過ぎているからである。メモリコンテンツがスクランブルされた順序で取得される場合、ヘルパーデータはスクランブルされた同じ順序で適用され得るか、またはスクランブルされたメモリコンテンツがスクランブル解除される。ハイブリッド解決策が可能である。例えば、スクランブルは、すなわち、スクランブル順序の逆転である並べ替えを行い、同時に、固定した事前に定義された順序が適用されることによって除去される。事前に定義された順序は、攻撃者には秘密であり得る。例えば、システムは製造中に事前に定義された順序で構成され得る。事前に定義された順序は、デバイスごとに異なるものであり得る。

【００２９】
メモリ読み出しユニットは、メモリを安全に読み出すように構成されている。スクランブルされた順序でメモリインタフェースによりメモリコンテンツを取り出すために、アドレススクランブラーが使用され得る。メモリの暗号化は、コンテンツが定められ得ないので、ＰＵＦの保護には直接的には適していない。順序を変更すること、すなわち、並べ替えることによってデータの難読化を示すスクランブルのことを言及しており、これはデータ自体の、すなわち暗号化を通じての変更によるデータの難読化とは異なるものである。

【００３０】
ＰＵＦからの暗号化キーの再構築の場合、ＰＵＦデータは、特に事前に定義された順序で、ヘルパーデータと組み合わされていなければならない。ＰＵＦの制御ブロックの設計では、ヘルパーデータが線形順序で読み出され得る場合に、より効率的である。特に、ヘルパーデータがランダムアクセスの外部アドレスバス提供していないが、データがブロック単位ベースで読み出される必要があり、何百から何千のビットの通常のブロックサイズを伴うメモリである場合である。例えば、ヘルパーデータメモリフラッシュメモリの場合、ヘルパーデータを異なる順序で読み出すことは非効率である。ヘルパーデータが線形的に読み出されうる場合、これは利点となるはずである。一実施形態では、ヘルパーデータがブロックごとに読まれ、各ブロックは線形的に読まれる。同時に、メモリインタフェースを介して受信された後、メモリコンテンツのスクランブル解除を回避することも利点となる。

【００３１】
このことは、メモリ読み出しユニットに暗号化および復号ユニットを有することによって、達成され得る。暗号化ユニットは、メモリを暗号化するように構成され、本暗号化ユニットはメモリインタフェースからメモリがスクランブルされた順序で確定したメモリコンテンツを受信し、受信したメモリコンテンツを暗号化し、暗号化されたメモリコンテンツをメモリに書き戻すように構成されている。復号ユニットは、メモリを復号するように構成され、メモリインタフェースから事前に定義された順序で暗号化されたメモリコンテンツを受信し、受信した暗号化されたメモリコンテンツを復号して事前に定義された順序でメモリが確定したメモリコンテンツを取得するように構成されている。

【００３２】

これによって、２つの段階で、メモリ電源投入データなど、事前に符号化され得ないデータの安全な読み出しが提供される。第１の段階で、安全性はスクランブル（並べ替え）により達成され、第２の段階で、安全性は暗号化により達成される。第１の段階では、ランダムアクセスを提供しないが、第２の段階で、メモリは望む任意の順序、特に、ヘルパーデータと組み合わせるために必要とされる順序でアクセスされ得る。後者は線形であり得るが、さらに追加の難読化のために何らかの他の事前に定義された順序にもし得る。後者の場合、ヘルパーデータは、同じ事前に定義された順序でＳＲＡＭデータに対して準備された。ヘルパーデータの観点からすれば、ＳＲＡＭデータが読み出される順序は、当該順序が、ヘルパーデータが準備された時と同じである限り、問題ではない。

【００３３】

暗号化では、データの各ブロックを一時キーで暗号化し得る。一時キーは、システム内部で導出され得る。例えば、一時キーは、おそらくＰＵＦ制御ブロックの内部である、第２のＰＵＦから導出され得、第２のＰＵＦは読み出される順序がバラバラなものおよび／または暗号化に対してランダム性を提供する。一実施形態では、暗号化されデータブロックは、直ちにＳＲＡＭに書き戻される。一実施形態では、暗号化ユニットは、スクランブルされた順序で暗号化コンテンツデータを書き戻し、それによって、メモリコンテンツを暗号化されたメモリコンテンツで上書きする。

【００３４】

スクランブルされた読み出しおよび暗号化された書き戻しは、ヘルパーデータが処理される前に行われる準備ステップである。暗号化キーが構築される必要がある時、すべてのＳＲＡＭ　ＰＵＦデータが順々に読み出され、ヘルパーデータとの排他的論理和をとられる前に暗号化される。

【００３５】

事前に定義された順序で書き込み可能メモリを安全に読み出すための電子メモリ読み出しユニットは特に利点となる。発明者らは、読み出しユニットが他の書き込み可能メモリの安全な読み出しのためにも同様に使用され得ることを理解している。こうして、安全で、保護されたサイドチャネル、読み出しを必要とし、何らかの理由で、事前に符号化または暗号化、つまり、ハミング重みの平衡コードで暗号化または符号化され得ないすべてのメモリは安全に読まれ得る。メモリは書き込み可能であることが好適である。その理由は、ツーパス方法が可能だからである。メモリは、例えば、揮発性でありうる（例えばＰＵＦとして使う）、または不揮発性でありうる（例えばデータ転送のために）。

【００３６】

一実施形態では、読み出しユニットは、ＰＵＦとして使用されるメモリとは異なり、さらなるメモリに電源投入したたびに、第２のメモリがノイズのあるメモリコンテンツに確定するように構成されたさらなる揮発性メモリを備え、該シードは第２のメモリが確定したメモリコンテンツから導出される。

【００３７】

メモリの読み出しを繰り返すことが必要な場合がある。例えば、一部の実施形態では、ＰＵＦとして使用されるメモリは、ＰＵＦ制御ブロックの電源、あるいは特に、メモリ読み出しユニットから独立して、再度電源投入され得る。例えば、このことは、暗号化キーを再度導出するために行われ得、これによって、キーが使用されると直ぐに削除されることが可能になる。しかし、ランダム性が新しい値を作成するための電源投入に依存するＰＵＦ、例えば、メモリベースのＰＵＦから取得される場合、ＰＵＦは新しいスクランブル順序を提供しない。このことは、新しいシードを取得するためにシードに適用されるように構成された暗号化一方向性関数を備える読み出しユニットによって回避され、アドレススクランブラーは電源投入しなくても、新しくスクランブルされた順序でメモリインタフェースによりメモリコンテンツの再フェッチのための新しいシードから新しくスクランブルされた順序を導出するように構成される。例えば、メモリ読み出しユニットは、ＰＵＦとして使用されたメモリが再度電源投入されたという信号を受信するように構成され得る。信号を受信すると、一方向性関数が適用され、新しいシードおよび／または暗号化キー

が導出される。

【００３８】

さらなるＰＵＦ、または第２のＰＵＦがＰＵＦ制御ブロックおよび／またはメモリ読み出しユニット内部にあることは好適であり、したがって、これは別個のＳＲＡＭ　ＰＵＦであり得る。第２のＰＵＦは、ＳＲＡＭ　ＰＵＦの測定のために読み出される必要のあるメモリアドレスのランダムな並べ替えを作成するために使用され得るランダムシードを生成するために使用され得る。例えば、ハッシュ関数などの調整アルゴリズムは、ＰＵＦ応答からランダムシードを計算するために使用され得る。

【００３９】

第２のＰＵＦもＩＰブロックに容易に統合され得る標準的な構成部品から構成されることが好適である。ＤＦＦ　ＰＵＦはこの状態のための良い選択である。Ｄフリップフロップの起動値は十分にランダムである。一方向性関数は、ＡＥＳ、ＤＥＳ、ＳＨＡなどの、暗号ブロックであり得る。

【００４０】

本発明の一態様は、暗号化キーを生成するための方法である。本方法は物理的クローン不能関数として使用されるメモリであって、本メモリは書き込み可能で、揮発性であるメモリの電源投入、メモリがメモリの少なくとも部分的にランダムな物理的特性に応じたメモリコンテンツに確定し、スクランブルされた順序でメモリインタフェースによりメモリコンテンツを取り出すことを可能にすること、メモリが確定したメモリコンテンツから暗号化キーを導出することを含んでいる。

【００４１】

本方法の一実施形態は、メモリインタフェースから、メモリがスクランブルされた順序で確定したメモリコンテンツを受信し、受信したメモリコンテンツを暗号化し、暗号化したメモリコンテンツをメモリに書き戻すことによってメモリを暗号化すること、および事前に定義された順序で暗号化されたメモリコンテンツをメモリインタフェースから受信することでメモリを復号すること、メモリが事前に定義された順序で確定したメモリコンテンツを取得するために受信した暗号化されたメモリコンテンツを復号することを含んでいる。

【００４２】

本発明による一方法は、コンピュータ実装方法としてコンピュータで、または専用ハードウェアで、または両方の組み合わせで実施され得る。本発明による方法の実行可能コードは、コンピュータプログラム製品に格納され得る。コンピュータプログラム製品の例には、メモリデバイス、光学式記憶装置、集積回路、サーバ、オンラインソフトウェアなどが含まれる。コンピュータプログラム製品が、前記プログラム製品がコンピュータで実行される時に、本発明による方法を行うためにコンピュータ読み取り可能媒体に格納された非一時的プログラムコード手段を含んでいることは好適である。

【００４３】

一好適な実施形態では、コンピュータプログラムは、コンピュータプログラムがコンピュータで実行される時に本発明による方法のすべてのステップを行うように適合されたコンピュータプログラムコード手段を含んでいる。コンピュータプログラムがコンピュータ読み取り可能媒体で具現化されることは好適である。

【００４４】

本発明のこれらおよび他の態様は、これ以後に記述される実施形態を参照する時に明らかになり、解明されるであろう。各図面は次のとおりである。

【図面の簡単な説明】

【００４５】

【図１ａ】暗号化キーを生成するための電子システムを例示するブロック図である。

【図１ｂ】図１ａのシステムで使用されるキー導出ユニットを例示するブロック図である。

【図２】暗号化キーを生成するための電子システムを例示するブロック図である。

【図３ａ】スクランブルを保持するコードワード境界を例示するブロック図である。
【図３ｂ】スクランブルを保持する非コードワード境界を例示するブロック図である。
【図４】暗号化キーを生成するための方法を例示する流れ図である。
【図５ａ】メモリのスクランブルおよび暗号化のための可能性を例示している図である。
【図５ｂ】メモリのスクランブルおよび暗号化のための可能性を例示している図である。
【図５ｃ】メモリのスクランブルおよび暗号化のための可能性を例示している図である。
【図５ｄ】メモリのスクランブルおよび暗号化のための可能性を例示している図である。
【発明を実施するための形態】
【００４６】
異なる図面で同じ参照番号を有する要素は同じ構造的特徴および同じ機能、または同じ信号を有していることに留意していただきたい。そのような要素の機能および／または構造が説明されている箇所では、詳細な説明で、それを繰り返して説明する必要性はない。
【００４７】
ブロック図の参照番号のリスト
１００　暗号化キーを生成するための電子システム
１１０　物理的クローン不能関数として使用されるメモリ
１１２　メモリの位置
１１４　メモリの位置
１１６　メモリの位置
１２０　メモリインタフェース
１２２　データチャネル
１２４　アドレスチャネル
１３０　ＰＵＦ制御ブロック
１４０　アドレススクランブラー
１５０　キー導出ユニット
１５２　バッファ
１５４　並べ替え装置
１５６　エラー訂正論理
２００　暗号化キーを生成するための電子システム
２１０　ＰＵＦ制御ブロック
２２０　メモリ読み出しユニット
２３０　キー導出ユニット
２４０　暗号化ユニット
２５０　復号ユニット
２５５　アドレス生成器
２６０　さらなるメモリ
２６５　ランダムキーおよびシード導出ユニット
２７０　バッファ
２８０　コンバイナ
２８５　ヘルパーデータメモリ
２９０　エラー訂正器
２９５　キー導出関数ユニット
３１０　メモリコンテンツ
３１２　コードワードサイズ調整メモリチャンク
３１４　コードワードサイズ調整メモリチャンク
３１６　コードワードサイズ調整メモリチャンク
３２０　スクランブルされたメモリアクセス
３２２　コードワードサイズ調整メモリチャンク
３２４　コードワードサイズ調整メモリチャンク
３２６　コードワードサイズ調整メモリチャンク

３３０　スクランブルされたメモリアクセス
５１０　暗号化段階
５１１　暗号化段階
５１２　暗号化段階
５１４　暗号化段階
５２０　復号段階

【００４８】

本発明は多くの異なる形の実施形態で可能であるが、図面で示しているとおり、本明細書では、１つ以上の特定な実施形態を詳細に説明することにしており、そのため、本開示は本発明の原理の例示として考慮されるべきであり、示され、記述される特定の実施形態に本発明を制限する意図はないことを理解していただきたい。

【００４９】

図１ａおよび図１ｂは、暗号化キーを生成するための電子システム１００をブロック図として例示している。

【００５０】

システム１００は、物理的クローン不能関数（ＰＵＦ）として使用されるメモリ１１０を備えている。メモリは書き込み可能および揮発性の種類からできている。その上、メモリは、メモリの各電源投入時に、メモリがメモリの少なくとも部分的にランダムな物理的特性に応じたメモリコンテンツに確定するという特性を有している。物理環境的変動は通常、製造中のわずかなプロセス変動により引き起こされる。

【００５１】

メモリが確定するメモリコンテンツはそのような物理的特性に応じて異なるので、メモリコンテンツはメモリの特定の具体化の識別となる。つまり、同じ設計の２つのメモリ１１０は、メモリを識別するために十分異なるメモリコンテンツを起動時に示す。例えば、メモリ１１０はフリップフロップに基づき得、特に、メモリ１１０はＳＲＡＭであり得る。

【００５２】

メモリ１１０は、電源投入時のコンテンツがＰＵＦとして使用されるメモリ位置のシーケンスを備えている。メモリ位置はアクセス可能で、つまり、対応するメモリアドレスのシーケンスを通して、読み取りまたは書き込みアクセスが可能である。メモリ位置のシーケンスの３つが１１２、１１４、および１１６で示されている。

【００５３】

システム１００はＰＵＦ制御ブロック１３０を備えている。ＰＵＦ制御ブロック１３０は、メモリ１１０が暗号化キーを作成するために起動時に確定するメモリコンテンツを処理するように構成されている。未処理の状態では、メモリコンテンツはキーとして直接使用できない。電源投入時のメモリ１１０のメモリコンテンツは、物理的特性だけではなく、メモリコンテンツはさらにノイズにも影響される。その上、温度、メモリに対する機械的ストレスなど、環境の物理的変動の影響もある。メモリコンテンツは、ノイズの影響を受けやすいので、暗号化キーとして直接使用することはできない。さらに、単一ビットのエントロピが低過ぎる場合もある。ＰＵＦ制御ブロック１３０は、エラー訂正および任意選択でキーの導出によってこれらに対応する。エラー訂正などは、キー導出関数１５０によって行われ、これは制御ブロック１３０に含まれている。

【００５４】

ＰＵＦ制御ブロック１３０は、メモリインタフェース１２０を通じてメモリ１１０に接続されている。メモリインタフェース１２０は、例えば、データ回線などのデータチャネル１２２、および例えば、アドレス線などのアドレスチャネル１２４を備えている。メモリインタフェース１２０はさらに、制御線なども備えている。

【００５５】

メモリインタフェースの使用には、これが潜在的サイドチャネルであるという欠点がある。データチャネル１２２により通信されるワードのハミング重みはシステム１００の電

力消費を観察することによって決定され得、同時に、ワードはデータチャネル１２２により転送される。例えば、データチャネル１２２が８ビット幅の場合、各ワードは情報の１～２ビットのオーダーでリークしうる。そのようなリークには、システム全体の安全性が含まれる。

【００５６】

スクランブルされた順序でメモリインタフェースによりメモリコンテンツを取り出すために、制御ブロック１３０はアドレススクランブラー１４０を備えている。アドレススクランブラー１４０が（疑似）ランダムな並べ替えを実施することは好適である。例えば、アドレススクランブラー１４０は、スクランブルされた順序でメモリアドレスのシーケンスを生成する。アドレススクランブラー１４０を実装する都合の良いやり方は、適切にサイズ調整された暗号化関数によってメモリアドレスのシーケンスを暗号化することによるものである。都合の良いことに、メモリ１１０のサイズは２の累乗なので、ビットの整数値によって１対１でアドレス指定され得る。この場合、ブロック暗号が使用され、ビットの整数値に等しいブロック幅を有し得る。そのようなブロック暗号は、ファイステル構造を使用して構築され得る。ファイステル構造は不平衡になり得るので、例えば、いわゆるソープシャッフル（Ｔｈｏｒｐ　ｓｈｕｆｆｌｅ）が使用され得る。任意のアドレス指定スキームまたは任意のサイズを有するメモリ１１０では一般により多くが、いわゆるフォーマット保持暗号化を使用して収容され得る。フォーマット保持暗号化は、アドレスのシーケンスをそれ自体に全単射的にマッピングする。

【００５７】

暗号化が使用される時、アドレススクランブラー１４０は、任意の順序、たとえば線形的にアドレスのシーケンスを生成し、シーケンスを暗号化し得る。スクランブルされた順序でアドレスのシーケンスを生成する代替のやり方が使用され得る。例えば、アドレススクランブラー１４０は、スクランブルされた順序でメモリアドレスのシーケンスを作成するように構成された、フィードバックシフトレジスタ、例えば、線形フィードバックシフトレジスタを備え得る。

【００５８】

メモリ１１０が確定したメモリコンテンツがスクランブルされた順序で取り出される。

【００５９】

図１ａ、図１ｂの実施形態では、キー導出ユニット１５０はスクランブルされた順序でメモリコンテンツを受信する。図１ｂでは、メモリコンテンツを処理する特定のやり方を示している。

【００６０】

キー導出関数１５０は、バッファ１５２、並べ替え装置１５４、およびエラー訂正論理１５６を備えている。

【００６１】

バッファ１５２は、メモリ１１０から、例えば、メモリ読み出しユニット（図１ａでは個別に図示されていない）から受信したメモリコンテンツをバッファリングする。並べ替え装置１５４はアドレススクランブラー１４０により適用される逆転並べ替えを行う。並べ替え装置１５４は、メモリ１１０で作成されたものと同じ順序でメモリコンテンツをバッファ１５２で作成し得る。しかし、並べ替え装置１５４は、事前に定義された順序を表す並べ替えとスクランブルされた並べ替えの逆転との関数合成を行うことによって、事前に定義された任意の順序でメモリコンテンツをバッファ１５２で作成し得る。並べ替え装置１５４は、正しいスクランブル解除動作を取得するために、スクランブラー１４０に接続されるか、またさらに統合され得る。逆転並べ替えの結果がエラー訂正論理１５６に渡される。エラー訂正論理１５６はヘルパーデータと組み合わされることによってノイズを訂正し、任意選択で、キー導出関数（ＫＤＦ）を結果に適用し、例えば、結果は暗号化ハッシュ関数でハッシュされ得るはずである。ＰＵＦとして使用されるメモリのメモリコンテンツのノイズの訂正はそれ自体知られているキー導出関数ではなく、メモリ読み出しユニットとともにバッファ１５２および並べ替え装置１５４を含み得る。

【００６２】
メモリコンテンツはスクランブルされた形でメモリインタフェースを介してのみ移動するので、依然、ハミング重みを示すが、ハミング重みが対応するワードがもはや不明なので、リークされる情報はかなり少なくなる。
【００６３】
このことはかなりの利点であるが、図１ａ、図１ｂの実施形態には、いくつかの欠点がある。まず第１に、これには、並べ替え装置１５４によって、行われるスクランブル解除動作が必要である。これは、とりわけ、所定の位置で行われる必要のある場合、比較的コストのかかる動作である。
【００６４】
エラー訂正論理１５６が単一のコードワードを使用する場合には、バッファ１５２は、所定の位置ではないスクランブル解除を使用するために、メモリ１１０のサイズの２倍であるはずである。これは比較的コストがかかる。しかし、より小さいコードワードの訂正はリソースの効率をより増大させるので、エラー訂正では複数のコードワードを使用することが好適である。この場合、スクランブル動作は、コードワードの境界を考慮するように構成され得るが、例えば、より大きなバッファの使用または本明細書で記述される他の解決策の使用という他のオプションもある。図３ａは、スクランブル動作を例示している。
【００６５】
メモリコンテンツ３１０はコードワードサイズのメモリチャンクに分割される。図３ａはそのような３つのチャンク３１２、３１４、３１６を示している。チャンクのそれぞれは、他のチャンクから独立してエラー訂正され得る。３２０では、メモリの同じ部分が図示されているが、メモリアクセスがスクランブルされている。矢印は、元のアドレスに対応したスクランブルされたアドレスを指している。このマッピングはコードワード境界を考慮していることに留意されたい。つまり、メモリアドレスのシーケンスは一連のメモリアドレスの複数のコードシーケンスを含み、アドレススクランブラー１４０は、１つの複数のコードシーケンスが異なるシーケンスのアドレスによってインターリーブされることなく、共に生成されるようにアドレスのシーケンスを生成する。そのようなアドレスのスクランブルによってサイドチャネルのリークの防止はより少なくなるが、著しい改善がある。図３ｂは、コードワード境界を考慮しないスクランブルされたメモリアクセス３３０を示している。このようなスクランブル関数では、リークをより防止できるが、図１のキー導出関数によって収容するのは（不可能ではないが）より困難である。図２に関連して以下で記述するシステムでは、２つのパスシステムを使用して、必ずしもコードワード境界を考慮する必要のない、スクランブルされたメモリアクセス３３０などのスクランブルされたメモリアクセスを処理できる。
【００６６】
並べ替え装置１５４を使用する代わりに、スクランブルをヘルパーデータと組み合わせの間に考慮に入れることも可能である。この場合、キー導出ユニット１５０は、バッファ１５２から、およびヘルパーデータメモリから（図１ｂではどちらも図示していない）読むコンバイナを含み得る。このアプローチではさらに、ヘルパーデータメモリは通常、不揮発性メモリで、通常、ブロック単位で読み出され、そのため、ヘルパーデータのランダムアクセスではよく、高い性能上のペナルティがつくという欠点がある。
【００６７】
図２は、暗号化キーを生成するための改善された電子システム２００を例示している。図４は、バリエーションが可能であるが、システム２００を使用して行い得る暗号化キーを生成するための方法を流れ図で例示している。
【００６８】
システム１００と同様に、システム２００は、メモリインタフェース１２０を通して、ＰＵＦ制御ブロック２１０に接続されたＰＵＦとして使用されるメモリ１１０を備えている。ＰＵＦ制御ブロックは、メモリ読み出しユニット２２０およびキー導出ユニット２３

０を備えている。

【００６９】

メモリ読み出しユニット２２０は、事前に定義された順序で安全にメモリ１１０を読み出すように構成されている。事前に定義された順序は、後続のキー導出のために何が都合が良いかによって決められる。通常、事前に定義された順序は自然な線形順序であるが、任意の順序が可能である。例えば、難読化が増大している場合、ランダムであるが、固定された事前に定義された順序も選択され得、ヘルパーデータは、当該ランダムであるが、固定された事前に定義された順序にしたがって並べ替えられたメモリコンテンツに対して計算され得る。ここから、事前に定義された順序は線形であるものとするが、これは変更され得ることに留意していただきたい。メモリコンテンツは、メモリ１１０の電源投入によって取得され（ステップ４１０）、メモリはメモリコンテンツに確定される（ステップ４２０）。確定時間はメモリにより異なるが、必要に応じて、実験的に決められ得る。確定時間は通常、電子コンピューティングデバイスで通常、行われるブート機能よりもかなり短いので、通常、メモリの読み出しユニットのソフトウェア実装のためにメモリの読み出しを遅延させる遅延素子を導入することは必要とはされない。

【００７０】

キー導出ユニット２３０は、メモリ１１０がメモリ読み出しユニット２２０から事前に定義された順序で確定したメモリコンテンツを受信する。一実施形態では、あるものは普通のＰＵＦ処理（エラー訂正および通常のキー導入関数）を行うように構成された、知られているキー導出ユニットを使用し得る。知られているキー導出ユニットが使用される場合には、キー導出ユニット２３０の観点から、これは通常のメモリコンテンツデータを受信する。こうして、システムのこの部分のみがおおまかな説明を与えられる。

【００７１】

キー導出ユニットはバッファ２７０を備え得る。バッファ２７０は単一のコードワードを保持できるほど大きい。キー導出関数は、ヘルパーデータメモリ２８５で記憶されたヘルパーデータとバッファのコンテンツを組み合わせるためのコンバイナ２８０を備え得る。組み合わせる関数は通常のビット単位の排他的論理和をとる動作であると想定する。しかし、任意の可逆なバイナリ動作が、例えば、異なるワード値の数を法として、例えば、バイトに対して２５６を法として使用され得る。ヘルパーデータはエラー訂正コードのコードワードとともにメモリコンテンツの事前に取得したコピーの排他的論理和をとることによって取得され得る。コードワードは秘密で、エラー訂正コードからランダムに選択されるか、または一部のキー管理スキームにしたがって選択される。動作中、コンバイナ２８０は秘密のコードワードに近傍にある訂正可能なビットストリングを取得するためにバッファ２７０で取得されるメモリコンテンツともにヘルパーデータの排他的論理和をとる。エラー訂正器２９０は、再度秘密のコードワードを取得するためにエラー訂正コードに対応するエラー訂正コードアルゴリズムを適用する。必要に応じて、キー導出関数ユニットは、キー導出関数（ＫＤＦ）、例えば、暗号化ハッシュ関数などを適用する。暗号化関数は、任意の暗号化の目的、例えば、認証、暗号化などに使用され得る。

【００７２】

興味深いことに、エラー訂正では、エラー訂正のためのツーパスシステムも同様に使用し得る。メモリはより小さなワードに分割され得、それは個別に、例えば、アダマールコードからコードワードへ訂正される。訂正されたより小さなコードワードは、より大きなブロックサイズエラー訂正コード、たとえばＢＣＨコードで訂正されるより大きなワードへ組み合わされる。より大きなワードへの組み合わせはより小さなコードワードをインターリーブすることが好適である。キー導出関数は、訂正されたより小さなコードワードをメモリに暗号化された形で書き戻し得る。キー導出関数は、例えば、インターリーブを行うために、暗号化ユニットを使用して、メモリへのランダムアクセスを行い得る。このようにして、メモリ１１０はより複雑化したエラー訂正スキームのためのワーキングメモリとして使用され得、同時に引き続き、サイドチャネルのリークを低減化することができ、メモリ１１０が暗号化されると、暗号化および復号ユニットを通じて、ランダムアクセス

の読み取り／書き込みアクセスをサポートする。

【００７３】

キー導出ユニットは対応策を使用して、必要に応じて、サイドチャネルのリークを低減化し得る。例えば、ＷＯ２０１０／１０００１５で記述されたデバイスおよび方法が使用され得る。

【００７４】

メモリ読み出しユニットは、スクランブルされた順序でメモリインタフェースを介してメモリコンテンツを取り出すためのアドレススクランブラー１４０、および暗号化ユニット２４０を備えている。暗号化ユニット２４０はスクランブラー１４０によって取り出されたデータを受信し、それを暗号化する。暗号化されたデータ、すなわち、暗号化されたメモリコンテンツがメモリに書き込まれる。任意のメモリが使用され得るが、暗号化されたメモリコンテンツはメモリ１１０に書き戻されることが好適である。必要とされるメモリ量を低減化するために、メモリコンテンツは同じスクランブルされた順序で書き戻され得る。それによって、メモリコンテンツが暗号されたメモリコンテンツで上書きされ、各暗号されたメモリワードは、読み出された同じスクランブルされたアドレスに書き戻される。暗号化ユニット２４０とスクランブルユニット１４０は共に、メモリを所定の場所、しかし、スクランブルされた順序で暗号化する効果があり得る。

【００７５】

暗号化ユニット２４０はブロック暗号であり得る。例えば、暗号化ユニット２４０は、電子コードブックモード（ＥＣＢ）で動作し得る。例えば、暗号化ユニット２４０は、カウンタモード（ＣＴＲ）、例えば、アドレスをＥＣＢモードで暗号化して、メモリコンテンツへの結果の排他的論理和をとり動作し得る。ブロックサイズはよく非常に小さい、つまり、８、１６、または３２ビットなので、後者が好適である。その理由は、偶然に等しい値を有するメモリコンテンツのワード間の関係を曖昧にするからである。

【００７６】

暗号化ユニットではキーを必要とする。アドレススクランブラーでは、シード、またはアドレス暗号化が使用される場合にはキーが必要とされる。キーは固定され得るが、メモリ読み出しユニット２２０の起動時に作成されることがより好適である。キー／シードは、ランダムに、真または疑似ランダムのいずれかで作成され得る。例えば、メモリ読み出しユニット２２０は、真のランダム数発生器（図示せず）を含み得る。

【００７７】

しかし、ランダム性は通常、メモリ読み出しユニット２２０の電源投入の間にのみ必要とされることが観察されている。このことによって、図２で示されている別の解決策が可能になる。メモリ読み出しユニット２２０はさらなるメモリ２６０を備えている（ＰＵＦ制御ブロック２１０内でメモリ１１０とは異なるものであることが好適である）。また、さらなるメモリ２６０も揮発性である。さらなるメモリの各電源投入時に、第２のメモリはノイズのあるメモリコンテンツに確定する。さらなるメモリのコンテンツはさらに物理的特性によっても異なるが、この目的のために、メモリコンテンツのノイズが利用されることに留意されたい。ランダムキーおよびシード導出ユニット２６５は、例えば、ハッシュ関数をメモリ２６０のコンテンツに適用することによって、メモリ２６５のメモリコンテンツからキーおよび／またはシードを導出する。例えば、後でメモリ１１０のコンテンツを再読み込みするために、新しいキー／シードが必要な場合には、新しく電源投入しなくても、新しいキーおよび新しいシードは、一方向性関数を古いシードに適用することによって取得され得る。

【００７８】

メモリ読み出しユニット２２０はさらに、復号ユニット２５０およびアドレス生成器２５５を備えている。アドレス生成器２５５は、復号ユニット２５０と組み合わせられ得る。アドレス生成器２５５は、事前に定義された順序でメモリアドレスのシーケンスを作成するように構成されている。したがって、復号ユニット２５０は受信した暗号化されたメモリコンテンツを復号する。

【００７９】
こうして、安全な読み出しが２つのパスで得られる。第１のパスでは、メモリが暗号化状態にある。このパスは、より少なくサイドチャネルのリークにさらされる。その理由は、メモリへのアクセスがスクランブルされた順序だからである。第２のパスでは、メモリは、事前に定義された順序など、望ましい任意の順序で読み出される。第２のパスがより安全なのは、メモリインタフェースを介して渡されるすべてのデータが暗号化されているからである。システム１００を上回るシステム２００の利点は、任意のスクランブル関数がコードワード境界を考慮しなくても使用され得ることで、同時に、キー導出がより小さくサイズのコードワードでも引き続き機能し得ることである。

【００８０】
システム１００および２００は、例えば、半導体デバイスなどの電子デバイスとして実装され得る。システム１００および２００は専用ハードウェアに実装され得る。システム１００および２００の一部はソフトウェアとして実装され得る。後者の場合、通常、システム、たとえば、ＰＵＦ制御ブロックは、デバイスで格納される適切なソフトウェアを実行するマイクロプロセッサ（図示せず）を備え、例えば、ソフトウェアはダウンロードされ、対応するメモリ、例えば、ＲＡＭ（図示せず）またはフラッシュなどの不揮発性メモリに記憶され得る。

【００８１】
図４は、暗号化キーを生成するための方法を流れ図で例示している。図４の方法は、バリエーションが可能であるが、システム２００を使用して行われ得る。図４のステップ４４０、４５０、４６０、および４７０は方法のためのオプションで、例えば、システム１００を使用して、省略され得る。

【００８２】
システム２００の動作は、次のとおり行われ得る。最初に、システムに電源を投入する。（ステップ４１０）電源投入の間、メモリ１１０は電源投入され、メモリコンテンツで確定され得る（ステップ４２０）。メモリ１１０のメモリコンテンツは、ノイズがあり得るが、この特定のメモリの見本である。このメモリコンテンツの読み出しは、次の２つのパスで行われる。第１のパスで、メモリコンテンツは、たとえばステップ４３０のスクランブラーによって、スクランブルされた順序で取り出され（ステップ４３０）、たとえば暗号化ユニット２４０による暗号化のために受信される。メモリコンテンツは暗号化され（ステップ４４０）、書き戻される（ステップ４５０）。第２のパスで、暗号化されたメモリコンテンツは、事前に定義された順序、たとえば線形シーケンス順序で、たとえばアドレス生成器２５５によって取り出され、復号ユニット２５０によって受信される（ステップ４６０）。暗号化されたメモリコンテンツは、メモリ１１０が事前に定義された順序で確定したメモリコンテンツを取り出すために復号される（ステップ４７０）。

【００８３】
この点で、メモリコンテンツは、サイドチャネルのリークがない、または最低限で、メモリインタフェース１２０を介してメモリ１１０から安全に渡される。ＰＵＦの処理は、１つ以上の訂正可能なビットストリングを取得するためにメモリコンテンツとヘルパーデータを組み合わせることによって、進み得る。訂正可能なビットストリングは、エラー訂正コードの訂正可能な近傍にある。１つ以上の訂正可能なビットストリングは、エラー訂正アルゴリズムを使用してエラー訂正コードのコードワードを確立するためにエラー訂正され得る。キーは、ＫＤＦを適用することによって、確立されたコードワードから導出され得る。

【００８４】
ステップ４６０および４７０で、メモリコンテンツは事前に定義された順序で取得される。このことは、メモリコンテンツが、特定の順序を必要とするヘルパーデータなどの、データと組み合わされる場合に望ましいものである。しかし、ステップ４３０、４４０、４５０、４６０、および４７０は、単なるＰＵＦデータではなく、書き込み可能メモリから安全にすべてのデータを読み取るため、例えば、ステップ４１０、４２０、および／ま

たは４８０なしで、共に使用され得る。この場合、メモリコンテンツがステップ４６０および４７０において事前に定義された順序で受信される必要は必ずしもなく、代わりに、任意の望ましいさらなる順序も可能である。例えば、さらなる順序でメモリを安全に読み出すための方法は、請求項１１のように暗号化キーの生成のために、スクランブルされた順序でメモリインタフェースからメモリコンテンツを受信し（４３０）、受信したメモリコンテンツを暗号化し（４４０）、暗号化されたメモリコンテンツをメモリに書き戻す（４５０）ことによってメモリを暗号化すること、およびさらなる順序で暗号化されたメモリコンテンツをメモリインタフェースから受信し（４６０）、さらなる順序でメモリコンテンツを取得するために受信した暗号化されたメモリコンテンツを復号する（４７０）ことによってメモリを復号することを含んでいる。

【００８５】

方法４００を実行するには多くの異なる方法が可能で、それは当業者には明らかになるであろう。例えば、ステップの順序は変更可能であり、一部のステップは並行して実行され得る。さらに、ステップの間に、他の方法ステップが挿入され得る。挿入されたステップは、本明細書で記述されているような方法の改善点を表すか、または本方法には関係しないものであり得る。例えば、ステップ４７０および４８０は、少なくとも部分的に平行して実行され得る。その上、所与のステップは、次のステップが開始される前に完全に完了しなくても構わない。

【００８６】

本発明による方法はソフトウェアを使用して実行され得、これにはプロセッサシステムが方法４００を実行できるようにするための命令が含まれる。ソフトウェアには、システムの特定のサブエンティティによって行われる当該ステップのみが含み得る。ソフトウェアは、ハードディスク、フロッピー（登録商標）、メモリなどの適切な記憶媒体に記憶され得る。ソフトウェアは、有線、または無線によって、あるいは、例えば、インターネットなどのデータネットワークを使用して送信され得る。ソフトウェアはダウンロードおよび／またはサーバ上でのリモート使用で使用可能にされ得る。

【００８７】

本発明はさらに、コンピュータプログラム、特に、本発明を実施するために適合されたキャリア上または中の特定のコンピュータプログラムに拡張していることを認識していただきたい。プログラムは、ソースコード、オブジェクトコード、コソースとオブジェクトの中間的コード、たとえば部分的にコンパイルされた形、または本発明による方法の実施での使用のために適切な他の任意の形であり得る。コンピュータプログラム製品に関連した一実施形態では、述べられた方法の少なくとも１つの処理ステップのそれぞれに対応するコンピュータ実行可能命令を備えている。これらの命令は、静的または動的にリンクされ得るサブルーチンに細分化および／または１つ以上のファイルに格納され得る。コンピュータプログラム製品に関連した別の実施形態では、述べられたシステムおよび／または製品の少なくとも１つの手段のそれぞれに対応するコンピュータ実行可能命令を備えている。

【００８８】

図５ａ～図５ｄは、メモリのスクランブルおよび暗号化のための様々な可能性を例示している。簡単にするために、メモリの８つのメモリ位置が示されており、メモリコンテンツは少なくともこれら８つのメモリ位置を超えて拡張する。実際のところ、かなりより大きなメモリが使用され得る。図面では、メモリ位置は１から８の数字で示されている。メモリ位置は、例えば、８ビットワード、または１６ビットワードなどのワードであり得、メモリアクセスへの読み取りアクセスは「Ｒ」で示され、書き込みアクセスは「Ｗ」で、難読化アクセスは「Ａ」で示されている。難読化アクセスは、同じメモリ位置に対する読み取り、書き込み、または読み取りと書き込みの組み合わせとし得る。時間は左から右へと増大する。

【００８９】

図５ａでは、２つの段階、暗号化段階５１０および後続の復号段階５２０が示されてい

る。暗号化段階の間、メモリ読み出しユニットは次を繰り返す。すなわち、メモリ位置から、すなわち、メモリアドレスから読み出して、メモリアドレスから読み出したコンテンツを暗号化し、そして同じメモリアドレスに書き戻す。メモリアドレスは、アドレススクランブラーによってスクランブルされている。暗号化段階５１０の最後に、メモリがスクランブルされた順序で読み出されても、メモリコンテンツ全体が暗号化されている。図５ａで示されている特定のスクランブルされた順序は例示的なものである。

【００９０】

復号段階の間、復号ユニットはさらなる順序で暗号化されたメモリコンテンツを受信した。図５ａで使用されたさらにある順序は、自然なシーケンス順序である。通常、スクランブルされた順序は、同じデバイスの後続の電源投入が比較されると異なるが、さらなる順序は同じである。キー導出またはエラー訂正などの間に最善の性能を有するには、暗号化段階が終了後に復号段階５２０を開始するのが好適である。このことは、例えば、さらなる順序の第１のメモリ位置への暗号化された書き戻しが行われた後に、復号段階が当該位置の読み取りを開始し得、さらなる順序の次のメモリ位置で行われた後に、復号段階が１つのさらなる復号に続き得る場合には、必ずしも厳密に必要とされる訳ではない。この複雑さの増大は、難読化が追加の複雑さを犠牲にしても最大化する必要がある時にのみ、その価値があることになる。図５ｂ～図５ｄで、このことは行われないと想定すると、復号段階はそのため、図５ａと同じであり、別個には示されていない。

【００９１】

暗号化段階がスクランブルされたメモリアドレスのシーケンスに対する読み取りおよび暗号化された書き戻しの繰り返されたサイクルを含む図５ａの実施形態は、複数の異なるやり方に変化し得る。そのようなバリエーションは、新型の攻撃を防止するのに役立つ。

【００９２】

暗号化段階の間の読み取りおよび書き込みアクセスは厳密に交互である必要はない。例えば、暗号化段階は、複数のメモリコンテンツを読み取り、それらを暗号化して複数の暗号化されたメモリコンテンツを書き戻しうる。書き戻しが行われる順序は、同じか、または読み取られた順序と同じ場合も、または同じではない場合もある。暗号化された順序は、線形または再スクランブルされたものであり得る。図５ｂは暗号化段階５１１を示している。ここで、読み取りおよび書き込み動作は、この場合、それぞれ４つの読み取りおよび４つの書き込みという複数のバッチで行われる。バッチでは線形の書き戻しを使用し、第２のバッチはスクランブルされた書き戻しを使用する。バッチは、アドレスごとに、読み取り値とその暗号化を相関させることを難しくすることによって、５ａの場合に対して改善される。

【００９３】

最近、光子放出攻撃は、アクセス中にＳＲＡＭ構造のバックサイドから光子活動を記録することによって直接そのターゲットＳＲＡＭで威力が実証されている。光子放出攻撃は電子デバイスが、動作時に特定の確率で光子を放出する傾向にあるということに基づいている。光子の生成レートは、供給電圧およびトランジスタスイッチング周波数に比例する。特に、ＳＲＡＭ値の読み取りも光子の放出の原因となる。ＩＣのバックサイドをＣＣＤでねらう時、読み取り動作を記録することによってメモリコンテンツでの情報を収集することは可能である。光子放出解析は他の暗号解読方法および／またはサイドチャネルを支援し得る。

【００９４】

潜在的に、光子攻撃は、例えば、各ＰＵＦ　ＳＲＡＭアドレスのまさに最初に読み取りを記録するためにＳＲＡＭ　ＰＵＦを攻撃するために使用され得るはずである。ＳＲＡＭの電源投入のメモリコンテンツの読み出し全部が取得される場合、このことは深刻な安全上の問題になるはずである。例えば、光子放出の記録がすべてのまさに最初の読み取りをキャプチャし、ＳＲＡＭからはそれ以上何もキャプチャせず、多くの回数、測定を繰り返すようにプログラムされる場合、ＳＲＡＭ開始値の信頼できるイメージが組み立てられ得るはずである。

【００９５】
それ自体での読み取り順序のスクランブルは光子攻撃に対処するために十分ではない、すなわち、ランダム化では、ピクチャは、順序に関係なく、単にすべての最初の読み取りアクションの総計なので、ピクチャを変更しないためである。幸い、図５ａおよび、より少ない程度、５ｂでの読み出しでは、電源投入時にメモリが確定したメモリコンテンツに相関された読み出しには、それほど相関がない書き戻しが散在している。これによって、光子情報の使用はかなり困難になる。このセットアップを攻撃するために、攻撃者は（最初の）読み取りアクセスを選び出すためにシャッターメカニズムを使用し得る。シャッターは読み取り中は開いた位置および書き込み中は閉じた位置で構成される。このようにして、暗号化されない読み取りの間の光子放出のみが記録される。一実施形態では、メモリ読み出しユニットは少なくとも部分的にランダムでバッチサイズを選択するように構成され、暗号化ユニットにより暗号化されたバッチサイズ数の書き戻し動作が続く、アドレススクランブラーによるバッチサイズ数の読み取り動作を行うように構成されている。例えば、バッチサイズは起動時に選択され得るが、例えば、バッチサイズは各バッチサイズが読み取られた後などに繰り返し選択され得る。少なくとも部分的なランダムがＰＵＦの一部に応じたバッチサイズを作成することで達成され得る。例えば、バッチサイズは、１、２、３、４バイトからランダムに選択され得る。

【００９６】
しかし、電源投入コンテンツに相関された信号をさらに低減化することが望ましい。

【００９７】
このことは、追加のランダム化されたアクセス、例えば、各メモリ位置への書き込みおよび／または読み取りを導入することで達成され得る。書き込みは最初の値を上書きするので、すべての後続の読み取りは混乱させるような情報を放出することになる。この対応策をより効果的なものにするために、アドレスごとに追加の書き込みおよび読み取りが他のアドレスと類似した書き込みおよび読み取りに対してランダムに順序付けされる必要がある。

【００９８】
ランダム化された順序でのＳＲＡＭに対する追加の読み取りおよび書き込みを行うことで、最初のＰＵＦデータの読み取りと分離することが難しいより多くの光子活動が生成される。追加の書き込みに使用されるデータはランダムデータ、固定されたパターン、またさらにデバイス固有のパターンであり得、ここで、ＰＵＦの一部はデバイス固有のパターンを導出するために使用される。さらに、これら３つの組み合わせも使用され得る。この順序のランダム化は、それ自体がＰＵＦ起動時に基づくランダムシードに基づき得る。ランダム化によって、すべてのメモリ位置で最初の時の読み取りアクセスを選び出すことが困難になる。

【００９９】
スクランブルされた読み出しおよび暗号化された書き込み準備ステップは、この攻撃を大いに阻害するために強化され得る。このことは、読み取り順序をスクランブルすることに加えて、読み出しおよび暗号化された書き戻し段階を併合することにより達成される。

【０１００】
暗号化されたデータの追加の読み取り、例えば、各アドレスからの数回の読み取り、最初の時の読み取りと後続の、例えば、ＰＵＦで得られたランダムシードに基づく、ランダム化での暗号化されたデータ読み取りを混合するなどを追加することによって、このスケジュールはそれぞれの電源投入で異なるので、攻撃者が読み取りおよび書き込みの正確なスケジュールを予測することは不可能である。その結果、ＳＲＡＭ起動値の信頼できるピクチャを作成することは非常に困難である。

【０１０１】
例えば、図５ｃの暗号化段階５１２は図５１０と同じスクランブルされた順序を示しているが、追加の難読化アクセスが追加されている。難読化アクセスは、例えば、読み出しユニット２１０に含まれる、難読化ユニットによって行われる。例えば、難読化ユニット

は、ユニット２１０の残りの各読み取りおよび／または書き込みアクセスの後に難読化アクセスを挿入する機会を有し得る。図５ｃでは、難読化アクセスは、書き戻しが既に発生したメモリ位置にのみ行われる。

【０１０２】

異なるタイプのアクセスも利点となる。アクセスは、追加の読み取りにし得るはずである。これは、書き込みに必要な値についての不明点を回避するので、実施がより簡単である。アクセスは、読み取り／書き込みサイクルにし得るはずである。書き込みアクセスは、光子放出の原因になる可能性がより高いので、したがって、追加の難読化をもたらす。アクセスは、読み取り／書き込み／書き込みサイクルであり得るはずで、最初の書き込みでは、ランダム値または固定された値、たとえば、すべて１、または読み取り値の反転が書き込まれ、次いで、第２の書き込みでは、読み取られた値が書き戻される。これによって、メモリ位置でのスイッチが増大し、光子放出もさらに増大する。したがって、難読化アクセスの実行には、メモリ位置、すなわちアドレス、好適には、暗号化された書き込みが既に行われたものを選択すること、および選択されたメモリアドレスへのアクセスを行うことが含まれ得る。

【０１０３】

アクセスはさらに、書き込みアクセスでもあり得る可能性があり、後者の場合、訂正値が書き込まれることに注意する必要がある。実施の効果と容易さとの特定の優れたトレードオフが図５ｄに示されている。図５ｄで、暗号化段階は、複数の読み取り／書き戻しサイクルからなっている。書き込みは繰り返され、暗号化された（したがって、相関されない）書き戻しの光子効果が増大する。さらに、このシナリオでは、第１の書き込みはランダム、反転、固定とされ得るはずである。あいにく、図５ｄでは、シャッターを使用してＲｉ選び出すために脆弱になる規則的なパターンＲｉＷｉＷｉを示しており、一実施形態では、読み取り／書き込みパターンはランダムなインターリーブで強化されている。例えば、書き込み動作の数は部分的にランダムになり得、例えば、上述のとおりバッチサイズ数が選択され得るが、バッチサイズ数は、書き戻しの繰り返しの数を制御するためにのみ使用される。他のランダムインターリーブも使用され得る。

【０１０４】

第１のものが暗号化された書き戻しの反転である２つの書き戻しを有することは、第１の読み取りで可能な光子放出の効果を低減化する光子放出を増大させる。

【０１０５】

一実施形態では、難読化ユニットは、メモリのメモリ位置への難読化アクセスを行うように構成されており、難読化アクセスは暗号化ユニットが暗号化されたメモリコンテンツをメモリ位置に書き戻した後に行われる。難読化ユニットが、どのメモリ位置がアクセスに安全かを把握し得る種々のやり方がある。さらなる実施形態では、難読化アクセスは、復号ユニットがメモリ位置から暗号化されたメモリコンテンツを受信した前である。

【０１０６】

別の実施形態では、難読化ユニットはランダムなメモリ位置から読み取りアクセスを行う。暗号化段階の間の固定された地点で、たとえば、スクランブルされた各読み取り後および各書き戻し後に、難読化ユニットは、難読化アクセス、好適には、読み取りアクセスをランダムなメモリ位置に挿入する機会を有している。難読化ユニットは、難読化ユニットが難読化アクセスを挿入する確率を示す確率値を備えている。暗号化段階の開始時点で、確率値はより低い値であるが、少なくとも暗号化段階の間に、確率値は、暗号化段階の終了時点で、確率値が高い値であるように増大する。例えば、確率値は、固定された開始値から固定された終了値に、各機会の後に線形に増大し得る。

【０１０７】

メモリのわずかな部分のみが暗号化される暗号化段階の最初では、追加の読み取りは攻撃者への信号を強化してしまうリスクがあるが、この段階では、読み取りの確率は低く、最後では、メモリの大部分が暗号化され、暗号化された値の読み取りの確率は高い。このスキームの利点は、どのメモリ位置が読み取りに安全か、すなわち、暗号化された値を有

しているかを把握するための記憶処理が不必要であることである。一実施形態では、確率値は、暗号化段階の最初から最後まで線形に０．１から０．９に増大する。さらなる順序でメモリを安全に読み出すための電子メモリ読み出しユニットの一実施形態では、本明細書で記述された難読化ユニットのいずれか一つを備えている。

【０１０８】

上述の実施形態は本発明を制限するものではなく、当業者が多くの代替の実施形態を設計できることを例示していることに留意すべきである。

【０１０９】

請求項で、括弧間に配置されたすべての参照記号は、請求項を制限するものとして解釈されるべきではない。動詞「ｃｏｍｐｒｉｓｅ（含む、備える）」およびその活用の使用は、請求項で述べられたもの以外の要素またはステップの存在を除外するものではない。要素の前の冠詞「ａ」または「ａｎ」は、そのような要素の複数の存在を除外するものでなない。本発明は、いくつかの別個の要素を備えるハードウェアによって、および適切にプログラミングされたコンピュータによって実装され得る。いくつかの手段を列挙するデバイスの請求項では、これらの手段のいくつかはハードウェアのいずれかおよび同じ要素によって具現化され得る。特定の方策が相互に異なる従属請求項で列挙されているということは、これらの手段の組み合わせは単に利点のために使用され得ないことを示してはいない。

【図１ａ】

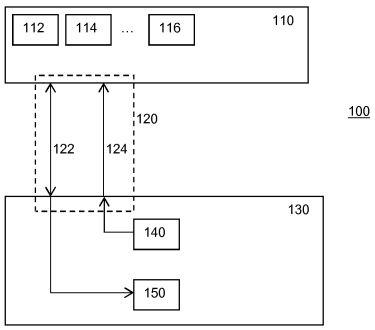

Figure 1a

【図１ｂ】

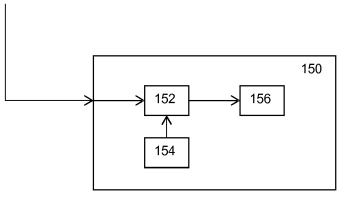

Figure 1b

【図２】

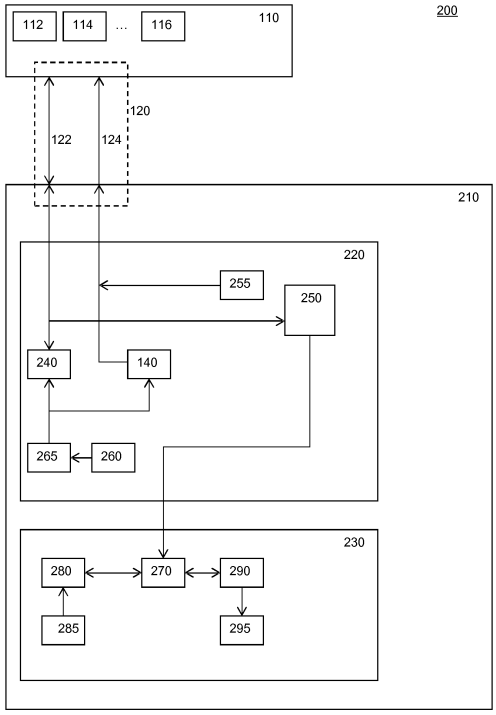

Figure 2

【図３ａ】

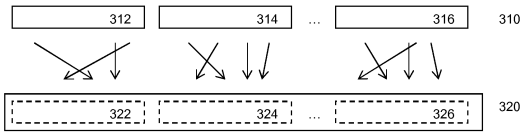

Figure 3a

【図３ｂ】

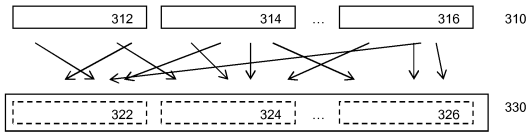

Figure 3b

【図４】

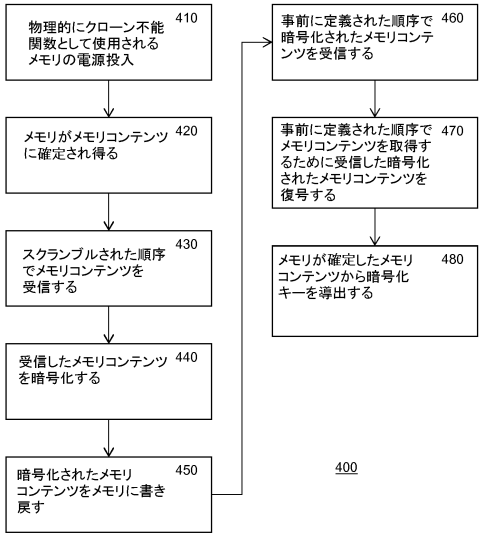

Figure 4

【図５ａ】

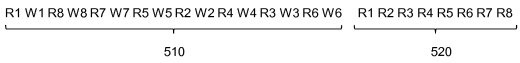

Figure 5a

【図５ｂ】

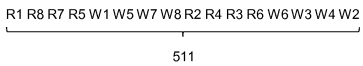

Figure 5b

【図５ｃ】

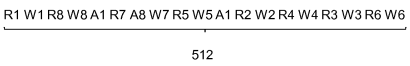

Figure 5c

【図５ｄ】

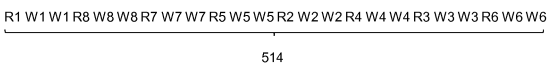

Figure 5d

フロントページの続き

審査官　中里　裕正

(56)参考文献　特表２０１２－５０９０３９（ＪＰ，Ａ）
米国特許出願公開第２００８／０１３３６２９（ＵＳ，Ａ１）
特開２００３－０１８１４３（ＪＰ，Ａ）
特表２０１２－５１９９８７（ＪＰ，Ａ）
米国特許出願公開第２００９／０１１３２１７（ＵＳ，Ａ１）
米国特許出願公開第２０１２／０１７９９５２（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　９／１０
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩＩ）
ＩＥＥＥ　Ｘｐｌｏｒｅ