(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-21637
(P2010-21637A)
(43)公開日　平成22年1月28日(2010.1.28)

(51)Int.Cl.	F I	テーマコード（参考）
H04L 9/10 (2006.01)	H04L 9/00 621A	5J104

審査請求 未請求　請求項の数 21　ＯＬ　（全 23 頁）

(21)出願番号　特願2008-178084 (P2008-178084)
(22)出願日　平成20年7月8日 (2008.7.8)

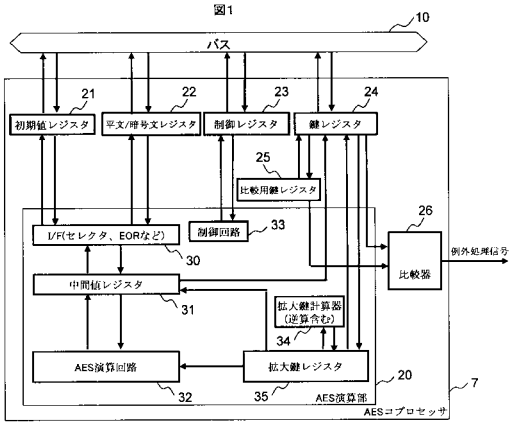
(71)出願人　503121103
株式会社ルネサステクノロジ
東京都千代田区大手町二丁目６番２号
(74)代理人　100089071
弁理士　玉村　静世
(72)発明者　成吉　雄一郎
東京都千代田区大手町二丁目６番２号　株式会社ルネサステクノロジ内
(72)発明者　佐伯　稔
東京都千代田区丸の内二丁目７番３号　三菱電機株式会社内
(72)発明者　鈴木　大輔
東京都千代田区丸の内二丁目７番３号　三菱電機株式会社内

Ｆターム(参考)　5J104 AA46　NA42

(54)【発明の名称】データ処理装置

(57)【要約】
【課題】暗号演算への故障攻撃に対して強固な検算対策を講ずることができるデータ処理装置を提供する。
【解決手段】データ処理装置は、暗号鍵を用いて平文の暗号化と暗号文の復号を行うプロセッサ（７）を備える。前記プロセッサは、前記暗号化又は復号を当初の演算として行ったとき、当該当初の演算の逆算を行い、この逆算において最終的に用いた暗号鍵を外部からアクセス可能に保持する記憶回路（２４）を有する。これにより、前記当初の演算で最初に用いた暗号鍵がデータ処理装置内で管理されているとき、その管理主体は、記憶回路が保有する前記逆算において最終的に用いた暗号鍵をアクセスする事によって、双方が一致するか否かによって、単なる逆算では検出不可能な故障攻撃の検出が可能になる。
【選択図】図１

図1



【特許請求の範囲】
【請求項１】
暗号鍵を用いて平文の暗号化と暗号文の復号を行うプロセッサを備えたデータ処理装置であって、
前記プロセッサは、前記暗号化又は復号を当初の演算として行ったとき、当該当初の演算の逆算を行い、この逆算において最終的に用いた暗号鍵を外部からアクセス可能に保持する記憶回路を有する、データ処理装置。
【請求項２】
暗号鍵を用いて平文の暗号化と暗号文の復号を行うプロセッサを備えたデータ処理装置であって、
前記プロセッサは、前記暗号化又は復号を当初の演算として行ったとき、当該当初の演算の逆算を行い、この逆算において最終的に用いた暗号鍵が当初の演算で最初に用いた暗号鍵に一致すか否かを検証する、データ処理装置。
【請求項３】
プロセッサを備えたデータ処理装置であって、
前記プロセッサは、その外部から入力した暗号鍵の初期値に基づいて順次更新した暗号鍵を用いた演算を順次複数回繰り返して暗号化を行い、また、外部から入力した暗号鍵に基づいて順次更新した暗号鍵を用いた演算を順次複数回繰り返して復号を行い、暗号化の演算又は復号の演算に用いた最新の前記更新した暗号鍵を外部からアクセス可能に保持する記憶回路を有する、データ処理装置。
【請求項４】
前記プロセッサを制御する中央処理装置を更に有し、前記記憶回路は前記中央処理措置のアドレス空間に配置される、請求項３記載のデータ処理装置。
【請求項５】
前記記憶回路は、前記暗号鍵の初期値が中央処理装置によって書き込まれる領域であって、その後、暗号鍵の初期値に基づいて順次更新した最新の暗号鍵によって書き換えられる領域である、請求項４記載のデータ処理装置。
【請求項６】
前記プロセッサは平文の暗号化に際して、その暗号化の結果に対し、当該暗号化の最後の暗号化演算で用いた暗号鍵を初期値とする復号を逆演算として行い、当該復号の最後に用いた暗号鍵を前記記憶回路に書き込む、請求項３記載のデータ処理装置。
【請求項７】
前記プロセッサは暗号文の復号に際して、前記暗号鍵の初期値を暗号化の場合と同様に繰り返し更新して暗号化の最後に用いる暗号鍵を生成し、生成した当該暗号鍵を初期値に用いた復号演算を順次複数回繰り返して復号を行い、復号の最後に用いた暗号鍵を前記記憶回路に書き込む、請求項３記載のデータ処理装置。
【請求項８】
前記記憶回路に書き込まれた前記復号の最後に用いた暗号鍵と、前記暗号化の最初に用いた暗号鍵とが一致するか否かを判別する比較器を更に有する、請求項６又は７記載のデータ処理装置。
【請求項９】
前記比較器による不一致の判別結果は例外処理を要求する信号であり、
前記例外処理が要求される中央処理装置を更に有する、請求項８記載のデータ処理装置。
【請求項１０】
前記例外処理は前記中央処理装置のリセット例外処理である、請求項９記載のデータ処理装置。
【請求項１１】
プロセッサを備えたデータ処理装置であって、
前記プロセッサは、その外部から入力した暗号鍵の初期値に基づいて順次更新した暗号

鍵を用いた演算を順次複数回繰り返す単位暗号化を複数単位分だけ直列的に実行して暗号化を行い、また、外部から入力した暗号鍵に基づいて順次更新した暗号鍵を用いた演算を順次複数回繰り返す単位復号を複数単位分だけ直列的に実行して復号を行い、

前記単位暗号化は、外部から入力した暗号鍵の初期値に基づいて順次更新した暗号鍵を用いた暗号化演算を順次複数回繰り返す暗号化処理とその暗号化処理による処理結果を用いる暗号化論理演算とから成り、

前記単位復号は、外部から入力した暗号鍵の初期値に基づいて順次更新した暗号鍵を用いた前記暗号化演算を順次複数回繰り返す暗号化処理とその暗号化処理による処理結果を用いる復号論理演算とから成り、

前記プロセッサは、前記暗号化において前ブロックの単位暗号化の暗号化処理の結果を受ける次ブロックの単位暗号化の暗号化処理に並行して、前記前ブロックの単位暗号化の暗号化処理の結果を用いて単位暗号化の暗号化処理の逆演算として復号処理を行い、前記逆演算に用いた最新の前記更新した暗号鍵を外部からアクセス可能に保持する記憶回路を有する、データ処理装置。

【請求項１２】

前記プロセッサは、前記復号において前ブロックの単位復号の暗号化処理の結果を受ける次ブロックの単位復号の暗号化処理に並行して、前記前ブロックの単位復号の暗号化処理の結果を用いて単位暗号化の暗号化処理の逆演算として復号処理を行い、

前記記憶回路は、前記逆演算に用いた最新の前記更新した暗号鍵を外部からアクセス可能に保持する、請求項１１記載のデータ処理装置。

【請求項１３】

前記プロセッサを制御する中央処理装置を更に有し、前記記憶回路は前記中央処理措置のアドレス空間に配置される、請求項１２記載のデータ処理装置。

【請求項１４】

前記記憶回路は、前記暗号鍵の初期値が中央処理装置によって書き込まれる領域であって、その後、暗号鍵の初期値に基づいて順次更新した最新の暗号鍵によって書き換えられる領域である、請求項１３記載のデータ処理装置。

【請求項１５】

前記暗号化論理演算及び前記復号論理演算は排他的論理和演算である、請求項１２記載のデータ処理装置。

【請求項１６】

前記記憶回路に書き込まれた前記復号の最後に用いた暗号鍵と、前記暗号化の最初に用いた暗号鍵とが一致するか否かを判別する比較器を更に有する、請求項１２記載のデータ処理装置。

【請求項１７】

前記比較器による不一致の判別結果は例外処理を要求する信号であり、

前記例外処理が要求される中央処理装置を更に有する、請求項１６記載のデータ処理装置。

【請求項１８】

前記例外処理は前記中央処理装置のリセット例外処理である、請求項１７記載のデータ処理装置。

【請求項１９】

前記プロセッサはＡＥＳに準拠する、請求項３又は１２記載のデータ処理装置。

【請求項２０】

データ処理装置は半導体基板に形成されたＩＣカード用のマイクロコンピュータである、請求項３又は１２記載のデータ処理装置。

【請求項２１】

前記プロセッサは、暗号アルゴリズムの暗号化機能のみを用いる暗号利用モードにおいて、復号処理による検算を処理する機能を備える請求項３又は１１記載のデータ処理装置。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、暗号演算への故障攻撃対策としての暗号演算の検算の信頼性を向上させるデータ処理技術に関し、例えばＡＥＳ（Advanced Encryption Standard）又はＤＥＳ（Data Encryption Standard）等に準拠した暗号コプロセッサに適用して有効な技術に関する。
【背景技術】
【０００２】
ICカード用マイコンは金融、交通、健保等を目的としたカードに実装され、電子マネーや個人情報を格納しているため、情報の漏洩や改ざんなどの攻撃をハードとソフトの組合せで防ぐ必要がある。ICカード用マイコンへの攻撃の一つとして故障解析が挙げられる。特に、局所的にパルスレーザを照射させることによる攻撃が近年健在化されている。例えば、FDTC(Fault Diagnosis and Tolerance in Cryptography)などの学会において、暗号演算に対するレーザ攻撃に関する発表がおこなわれている。ICカード用マイコンなどに搭載している暗号コプロセッサにおいて、誤動作により暗号演算結果が期待値と異なる値をチップ外部に出力すると、その誤暗号文が元で使用している鍵を抽出される可能性がある。これを回避する方法としては、再度同じ計算をする検査方法がある。例えば暗号化演算を複数回行い、複数回の演算結果が一致したときその結果を返すようにする。別の方法として、逆算を行う検算方法がある。例えば、暗号化演算の場合は復号演算を更に行い、復号演算の場合は暗号化演算を更に行い、逆算結果が一致したときその結果を返すようにする。そのような検算を行う方法は時間がかかるものの、パリティなどの冗長回路の導入することなく誤動作の検出が可能になる。尚、故障利用解析に対する対策について記載された文献として下記の非特許文献1がある。
【０００３】
【非特許文献１】神永 正博 渡邊 高志「情報セキュリティの理論と技術－暗号理論からICカードの耐タンパー技術まで」森北出版株式会社、2005年10月15日発行、第198－205頁。
【発明の開示】
【発明が解決しようとする課題】
【０００４】
暗号演算への故障攻撃の対策として上述のような二度以上の計算の実施か逆算が推奨されるが、本発明者の検討によれば以下の問題点のあることが明らかにされた。
【０００５】
〔１〕二度以上の計算をしても、その計算において同じデータを使用する場合、その値を攻撃することで、検算を無効化させることができる。例えば、AESの拡大鍵において、暗号化時に計算したものを検算でも同じ値を使用するシステムにすると、暗号化演算時に故障攻撃を受けた場合にはその影響を検出することができない。例えば図３に例示されるように拡大鍵レジスタへ一回攻撃を受けてその値が変化されると、同じ演算を何回繰り返しても同じ結果しか得られず、故障攻撃に対処することが出来ない。
【０００６】
〔２〕レーザなど故障をおこす装置の性能が非常に高い場合(出力エネルギーが高い、集光能力が高い、出力間隔が短いなど)には、拡大鍵を自律的に複数回更新しながら符号化又は復号演算を行うようにしても、複数回計算の各回の演算において同一タイミングで同一計算をしている場合、図４に例示されるように毎回、同じ故障注入を与え、同じ誤った値を生成させることで検算が無効化される可能性がある。
【０００７】
〔３〕ブロック暗号であるDESやAES等にはOFB（Output Feedback）モード、CTR（Counter）モード、CFB（Cipher Feedback ）モードなど、復号時においても共通鍵暗号そのものの演算は”暗号化”で演算する暗号利用モードがある。この場合、暗号化／復号演算で検算しても共通鍵暗号そのものの演算は”暗号化”であるため、同じ動作をすることにな

り、図４と同様の脆弱性が存在する。ECB(Electronic Codebook)モードならびにOFBモードでの暗号可／復号演算のデータフローをそれぞれ図５、図６に示す。平文P0～P2、暗号文C0～C2、鍵K、初期値IVにおいて、OFBモードの復号演算では共通鍵暗号演算部(図６の太枠部)は復号演算Dではなく、初期値IVを入力、鍵をKとした暗号化演算Eを実施しているため、図５の場合での検算と比較して脆弱である。

【０００８】

本発明の目的は、暗号演算への故障攻撃に対して強固な検算対策を講ずることができるデータ処理装置を提供することにある。

【０００９】

本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【課題を解決するための手段】

【００１０】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【００１１】

すなわち、逆算を行って検算を行う場合に、逆算において最終的に用いた暗号鍵が当初の演算で最初に用いた暗号鍵に一致するかの検証を追加する。これにより、逆算結果の検証では検出不可能な故障攻撃の検出が可能になる。

【発明の効果】

【００１２】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記のとおりである。

【００１３】

すなわち、暗号演算への故障攻撃に対して強固な検算対策を講ずることができる。

【発明を実施するための最良の形態】

【００１４】

１．実施の形態の概要

先ず、本願において開示される発明の代表的な実施の形態について概要を説明する。代表的な実施の形態についての概要説明で括弧を付して参照する図面中の参照符号はそれが付された構成要素の概念に含まれるものを例示するに過ぎない。

【００１５】

〔１〕本発明に係るデータ処理装置は、暗号鍵を用いて平文の暗号化と暗号文の復号を行うプロセッサ（７，７Ａ，７Ｂ）を備える。前記プロセッサは、前記暗号化又は復号を当初の演算として行ったとき、当該当初の演算の逆算を行い、この逆算において最終的に用いた暗号鍵を外部からアクセス可能に保持する記憶回路（２４）を有する。これにより、前記当初の演算で最初に用いた暗号鍵がデータ処理装置内で管理されているとき、その管理主体は、記憶回路が保有する前記逆算において最終的に用いた暗号鍵をアクセスする事によって、双方が一致するか否かによって、単なる逆算では検出不可能な故障攻撃の検出が可能になる。

【００１６】

〔２〕本発明に係る別のデータ処理装置は、暗号鍵を用いて平文の暗号化と暗号文の復号を行うプロセッサを備えたデータ処理装置において、前記プロセッサは、前記暗号化又は復号を当初の演算として行ったとき、当該当初の演算の逆算を行い、この逆算において最終的に用いた暗号鍵が当初の演算で最初に用いた暗号鍵に一致すか否かを検証する。プロセッサそれ自体で故障攻撃の検出が可能になる。

【００１７】

〔３〕本発明に係る更に別のデータ処理装置は、プロセッサを備え、前記プロセッサは、その外部から入力した暗号鍵の初期値に基づいて順次更新した暗号鍵（Ｋ０～Ｋ１０）を用いた演算を順次複数回繰り返して暗号化を行い、また、外部から入力した暗号鍵に基

づいて順次更新した暗号鍵（Ｋ１０～Ｋ０）を用いた演算を順次複数回繰り返して復号を行い、暗号化の演算又は復号の演算に用いた最新の前記更新した暗号鍵を外部からアクセス可能に保持する記憶回路（２４）を有する。ＥＣＢモードに代表される暗号化復号において、単なる逆算では検出不可能な故障攻撃の検出が可能になる。

【００１８】

〔４〕項３のデータ処理装置において、前記プロセッサを制御する中央処理装置（３）を更に有し、前記記憶回路は前記中央処理措置のアドレス空間に配置される。プロセッサが記憶回路から暗号鍵をリードして、期待値と比較することによって故障攻撃の検出が可能になる。

【００１９】

〔５〕項４のデータ処理装置において、前記記憶回路は、前記暗号鍵の初期値が中央処理装置によって書き込まれる領域であって、その後、暗号鍵の初期値に基づいて順次更新した最新の暗号鍵によって書き換えられる領域である。

【００２０】

〔６〕項３のデータ処理装置において、前記プロセッサは平文の暗号化に際して、その暗号化の結果に対し、当該暗号化の最後の暗号化演算で用いた暗号鍵（Ｋ１０）を初期値とする復号を逆演算として行い、当該復号の最後に用いた暗号鍵（Ｋ０）を前記記憶回路に書き込む。

【００２１】

〔７〕項３のデータ処理装置において、前記プロセッサは暗号文の復号に際して、前記暗号鍵の初期値を暗号化の場合と同様に繰り返し更新して暗号化の最後に用いる暗号鍵（Ｋ１０）を生成し、生成した当該暗号鍵を初期値に用いた復号演算を順次複数回繰り返して復号を行い、復号の最後に用いた暗号鍵（Ｋ０）を前記記憶回路に書き込む。

【００２２】

〔８〕項６又は７のデータ処理装置において、前記プロセッサは前記記憶回路に書き込まれた前記復号の最後に用いた暗号鍵と、前記暗号化の最初に用いた暗号鍵とが一致するか否かを判別する比較器（２６）を更に有する。プロセッサそれ自体で故障攻撃の検出が可能になる。

【００２３】

〔９〕項８のデータ処理装置において、前記比較器による不一致の判別結果は例外処理を要求する信号であり、前記例外処理が要求される中央処理装置を更に有する。

【００２４】

〔１０〕項９のデータ処理装置において、前記例外処理は前記中央処理装置のリセット例外処理である。

【００２５】

〔１１〕本発明に係る更に別のデータ処理装置はプロセッサ（７Ｃ）を有し、前記プロセッサは、その外部から入力した暗号鍵の初期値に基づいて順次更新した暗号鍵を用いた演算を順次複数回繰り返す単位暗号化を複数単位分だけ直列的に実行して暗号化を行い、また、外部から入力した暗号鍵に基づいて順次更新した暗号鍵を用いた演算を順次複数回繰り返す単位復号を複数単位分だけ直列的に実行して復号を行う。前記単位暗号化は、外部から入力した暗号鍵の初期値に基づいて順次更新した暗号鍵を用いた暗号化演算を順次複数回繰り返す暗号化処理（Ｅ）とその暗号化処理による処理結果を用いる暗号化論理演算とから成る。前記単位復号は、外部から入力した暗号鍵の初期値に基づいて順次更新した暗号鍵を用いた前記暗号化演算を順次複数回繰り返す暗号化処理（Ｅ）とその暗号化処理による処理結果を用いる復号論理演算とから成る。前記プロセッサは、前記暗号化において前ブロックの単位暗号化の暗号化処理の結果を受ける次ブロックの単位暗号化の暗号化処理（図９のＥ）に並行して、前記前ブロックの単位暗号化の暗号化処理の結果を用いて単位暗号化の暗号化処理の逆演算として復号処理（図９のＤ）を行い、前記逆演算に用いた最新の前記更新した暗号鍵を外部からアクセス可能に保持する記憶回路（２４）を有する。これにより、ＯＦＢモードに代表される動作モードによる暗号化において、単なる

逆算では検出不可能な故障攻撃の検出が可能になる。

【００２６】

〔１２〕項１１のデータ処理装置において、前記プロセッサは、前記復号において前ブロックの単位復号の暗号化処理の結果を受ける次ブロックの単位復号の暗号化処理に並行して、前記前ブロックの単位復号の暗号化処理の結果を用いて単位暗号化の暗号化処理の逆演算として復号処理を行い、前記記憶回路は、前記逆演算に用いた最新の前記更新した暗号鍵を外部からアクセス可能に保持する。ＯＦＢモードに代表される動作モードによる復号において、単なる逆算では検出不可能な故障攻撃の検出が可能になる。

【００２７】

〔１３〕項１２のデータ処理装置において、前記プロセッサを制御する中央処理装置（３）を更に有し、前記記憶回路は前記中央処理措置のアドレス空間に配置される。

【００２８】

〔１４〕項１３のデータ処理装置において、前記記憶回路は、前記暗号鍵の初期値が中央処理装置によって書き込まれる領域であって、その後、暗号鍵の初期値に基づいて順次更新した最新の暗号鍵によって書き換えられる領域である。

【００２９】

〔１５〕項１２のデータ処理装置において、前記暗号化論理演算及び前記復号論理演算は排他的論理和演算である。

【００３０】

〔１６〕項１２のデータ処理装置において、前記プロセッサは前記記憶回路に書き込まれた前記復号の最後に用いた暗号鍵と、前記暗号化の最初に用いた暗号鍵とが一致するか否かを判別する比較器を更に有する。

【００３１】

〔１７〕項１６のデータ処理装置において、前記比較器による不一致の判別結果は例外処理を要求する信号であり、前記例外処理が要求される中央処理装置を更に有する。

【００３２】

〔１８〕項１７のデータ処理装置において、前記例外処理は前記中央処理装置のリセット例外処理である。

【００３３】

〔１９〕項３又は１２のデータ処理装置において、前記プロセッサはＡＥＳに準拠する。

【００３４】

〔２０〕項３又は１２のデータ処理装置において、データ処理装置は半導体基板に形成されたＩＣカード用のマイクロコンピュータである。

【００３５】

〔２１〕項３又は項１１のデータ処理装置において、前記プロセッサは、暗号アルゴリズムの暗号化機能のみを用いる暗号利用モードにおいて、復号処理による検算を処理する機能を備える。

【００３６】

２．実施の形態の詳細

実施の形態について更に詳述する。以下、本発明を実施するための形態を図面に基づいて詳細に説明する。なお、発明を実施するための形態を説明するための全図において、同一の機能を有する要素には同一の符号を付して、その繰り返しの説明を省略する。

【００３７】

《ＩＣカード用マイコン》

図２には本発明の一例に係るＩＣカード用マイコンのブロックダイヤグラムが例示される。ＩＣカード用マイコン１は、非接触又は接触インタフェースを行うための通信モジュール２、ＩＣカード用マイコン1の全体的な制御を行う中央処理装置（ＣＰＵ）３、ＩＣカード用マイコンの割込みやリセット等の例外処理制御及び動作モードの制御等を行うシステムコントロールロジック４、ＣＰＵ３の動作プログラムやデータテーブル等を保有す

る不揮発性メモリ５、ＣＰＵ３のワーク領域等に用いられるＲＡＭ６、ＤＥＡ又はＡＥＳ等に準拠する共通鍵暗号プロセッサ７、べき乗剰余乗算や楕円暗号演算等を行う公開鍵暗号プロセッサ８、及びＩＣカード用マイコン1の動作基準クロックとされるクロック信号ＣＫを生成するクロック生成回路９等を有する。１０は内部バスである。不揮発性メモリ５にはマスクＲＯＭ、ＥＥＰＲＯＭ、又はフラッシュメモリ等を採用すればよい。このＩＣカード用マイコン１は、例えば単結晶シリコン等の1個の半導体基板にＣＭＯＳ等の集積回路製造技術によって構成される。

【００３８】

図１には共通鍵暗号プロセッサ７の一例としてＡＥＳコプロセッサの実装例が示される。ＡＥＳコプロセッサについては、“NIST (National Institute of Standards and Technology), ADVANCED ENCRYPTION STANDARD (AES), Federal Information Processing Standard Publications (FIPS PUBS) 197, 2001”の第１文献に詳細が記載される。ここでは本発明の特徴点に関連する構成についてその詳細を説明する。

【００３９】

共通鍵暗号プロセッサ７（単に暗号プロセッサ７とも記す）７は、ＡＥＳのサブバイト（SubByte）などを計算するＡＥＳ演算部２０、初期値レジスタ２１、平文／暗号文レジスタ２２、制御レジスタ２３、鍵レジスタ２４、比較用鍵レジスタ２５、及び比較器２６を有する。暗号演算における暗号利用モードについては“NIST, Recommendation for Block Cipher Modes of Operation, NIST Special Publication 800-38A, 2001”の第２文献に記載がある。暗号利用モードは例えば図５に概略が示されたECBモード、図６に概略が示されたＯＦＢモードがある。

【００４０】

《ＥＣＢモードにおける故障攻撃対策》

図１では暗号モードとして例えば図５のＥＣＢモードを用いる。図５の標記から明らかなように、ＥＣＢモードにおいてＥで示される暗号化演算とＤで示される復号演算とはその処理内容が相違される。図７にはＥＣＢモードにおいける暗号化と復号の演算手法が例示される。ここでは、図５の一つのＥで示される暗号化演算は図7のＥｎｃに示されるＳｅ０～Ｓｅ１０の複数ステップの繰り返しによって実現される。与えられた鍵Ｋは、ステップＳｅ０～Ｓｅ１０の各ステップにおいて順次Ｋ０～Ｋ１０に自律的に更新演算されて用いられる。Ｋ０＝Ｋであり、K１～K１０を拡大鍵と称する。Ｓｅ０～Ｓｅ１０のサフィックス１～１０をラウンド数と称する。すなわちラウンドｉのステップはＳｅｉとなる。ステップＳｅ０～Ｓｅ１０の各ラウンドで符号化演算が行われ、ＰがＭ１に符号化され、Ｍ１がＭ２に符号化され、Ｍ２がＭ３に符号化され、最後にＭ１０がＣに符号化される事によって、平文Ｐが暗号文Ｃに暗号化される。図５の一つのＤで示される復号演算は図７のＤｅｃに示されるＳｄ０～Ｓｄ１０の複数ステップの繰り返しによって実現される。ステップＳｄ０～Ｓｄ１０の各ステップにおいて鍵はＫ１０～Ｋ０に順次自律的に更新演算されて用いられる。復号においてもステップＳｄ０～Ｓｄ１０のサフィックスをラウンド数と称する。ステップＳｄ０～Ｓｄ１０の各ラウンドで復号演算が行われ、ＣがＭ１０に復号され、Ｍ１０がＭ９に復号され、Ｍ９がＭ８に復号され、最後にＭ１がＰに復号される事によって、暗号文Ｃが平文Ｐに復号される。各ラウンドの暗号化、復号演算には例えば第１文献及び第２文献に記載の具体的な処理を適用すればよく、ここではその詳細な説明は省略する。

【００４１】

このとき、暗号化に際して、演算された暗号文Ｃを復号する逆演算を行って平文Ｐが得られるかを検証する。併せて、上記逆演算において最終的にステップＳｄ１０で用いた拡大鍵Ｋ０が符号化の最初のステップＳｅ０で用いた鍵Ｋに一致するかの検証を行う。この検算は以下の意義を有する。例えばパルスレーザ照射などによる故障解析攻撃を受けて暗号化演算途中で拡大鍵の改ざん例えばＫ３がＫ３’に変化されたとすると、その影響は後続の拡大鍵に順次伝播し、ステップＳｅ１０では、拡大鍵Ｋ１０’によって暗号文Ｃ’が生成される。拡大鍵Ｋ１０’と暗号文Ｃ’を用いた復号逆演算が行われたとき、ステップ

Ｓｄ１０では平文Ｐが得られるが、最終的な拡大鍵はＫ０’となり当初のＫと不一致になる。暗号化において鍵が変化されると、それと同じ鍵を用いれば復号逆演算で得られる平文は暗号化対象の平文に一致する。したがって暗号文の逆演算だけでは検出できない異状を拡大鍵（ステップＳｅ０の鍵とステップＳｄ１０の鍵）の比較によって検出することが可能になる。これは、拡大鍵の符号化演算と復号演算の夫々が直前の拡大鍵の値を参照して行われるので、符号化演算の最初と、復号逆演算の最後との間で鍵の値が相違することによる。図１０は暗号化に際しての暗号化演算のタイミングチャートを例示し、図１１は暗号化に際して行われる復号逆演算のタイミングチャートを例示する。

【００４２】

復号においては、鍵の初期値Ｋから、復号の最初に用いる拡大鍵Ｋ１０を生成する拡大鍵演算だけを先ず予備的に行い、それによって得られた拡大鍵Ｋ１０を用いて図７のＤｅｃで示される復号演算が行われる。この復号演算においても、最終的にステップＳｄ１０で用いた拡大鍵Ｋ０が最初の鍵Ｋに一致するかの検証を行うことによって、予備的な拡大鍵演算ステップで故障解析攻撃を受けることにより途中で拡大鍵の改ざんが行われたか否かを検出する。この鍵の照合で復号時に故障解析攻撃を受けて拡大鍵が改ざんされたか否かを検出する事ができるので、復号においては、鍵の復号結果を暗号化する逆演算を行う必要はない。図１２は復号に際してのタイミングチャートを例示し、図１３は復号における予備的な拡大鍵演算処理で故障解析攻撃による拡大鍵の改ざんを受けたときの動作タイミングチャートを例示する。図１３において、誤った平文を出力させ、かつ例外処理信号をアクティブにさせないためには、最後の拡大鍵演算終了段階でK0になるよう再度故障をいれることが必要になる。しかしながら、１２８ビット全ビットをそろえるのは困難である。再度の故障注入が一番容易そうな２回目のK8’算出時への攻撃も、一回目の算出がラウンド7からの計算に対し、２回目の計算はラウンド９からの算出になり、異なる回路動作に対して故障を注入させてK8に戻すような故障解析攻撃を行うことは実質的に困難である。すなわち、本発明による故障解析攻撃対策は強固であると考えられる。

【００４３】

図１に戻ってＡＥＳコプロセッサ７の具体的な構成について説明を加える。図１において前記レジスタ２１～２４はＣＰＵ３によってアクセス可能にされるレジスタであり、鍵レジスタ２４には鍵Ｋが初期設定され、平文／暗号文レジスタ２２には暗号化すべき平文又は復号すべき暗号文がＣＰＵ３によりセットされ、また、ＡＥＳ演算部２０で演算された暗号文又は平文がセットされる。制御レジスタ２３には暗号化又は復号の演算開始指示ビット、ECBモード、CBC(Cipher Block Chaining)モードなどの暗号利用モードを選択する暗号利用モードビットなどから構成される。これらのビットはＣＰＵ３からバス１０介してリード/ライトが出来、ＣＰＵ命令により各モードの設定や、演算をしているかどうかを確認できる。初期値レジスタ２１はCBCモードなどにおいてでてくる初期値などを格納するレジスタである。平文／暗号文レジスタ２２や初期値レジスタ２１はＡＥＳコプロセッサ７による演算終了後に更新される。

【００４４】

ＡＥＳ演算部２０は、インタフェース回路（Ｉ／Ｆ）３０、中間値レジスタ３１、ＡＥＳ演算回路３２、制御回路３３、拡大鍵計算回路３４、及び拡大鍵レジスタ３５を有する。制御回路３３は制御レジスタの各ビットの情報をもとに、AESコプロセッサ７内部のデータ移動の制御やＡＥＳ演算回路３２の演算種別の選択、現在演算しているラウンド数の管理などの演算制御を行う。また、制御回路３３は演算終了時には制御レジスタ２３のステータスビットを変化させる制御なども行う。中間値レジスタ３１は各ラウンドの途中の暗号／復号結果を保管する。拡大鍵計算器３４は各ラウンドの拡大鍵の計算を順次行う。拡大鍵レジスタ３５は算出された拡大鍵を順次保管し、次の拡大鍵演算のために拡大鍵を拡大鍵計算器３４に渡す。最初の鍵Ｋは鍵レジスタ２４から拡大鍵レジスタ３５に転送され、拡大鍵レジスタ３５と鍵レジスタ２４との間では相互に鍵の入れ替えが行われる。すなわち、例えば、拡大鍵計算器３４で計算されて拡大鍵レジスタ３５に与えられた拡大鍵が鍵レジスタ２４に転送され、その前に鍵レジスタ２４が保有する拡大鍵が拡大鍵レジス

タ３５に転送される。比較用鍵レジスタ２５は最初の鍵Ｋを保管する。インタフェース回路３０は暗号利用モードによりデータ入力種類を選択したり、必要に応じてEOR（排他的論理輪）演算を実施する。ＡＳＥ演算回路３２は拡大鍵レジスタ３５及び中間値レジスタ３１の値を用いて、各ラウンドの暗号化、復号演算を行う。その演算には例えば第１文献及び第２文献に記載の具体的な処理を適用すればよく、ここではその詳細な説明は省略する。比較器２６は鍵レジスタ２４の値と比較用鍵レジスタ２５の値が所用の比較タイミングにおいて一致するかを確認するために用いられる。例えば所定タイミングとは、暗号化処理では図１１に例示されるように復号逆演算にて最後のＫ０の値が確定するタイミング、復号処理では図１２に例示されるように予備的演算に後の復号演算の最後のＫ０の値が確定するタイミングである。不一致の場合は例外処理信号により、リセット状態に遷移するなどの、故障対策がシステムコントロールロジック４に搭載されている。

【００４５】

図１４には図１の回路において、鍵長が１２８ビット、暗号利用モードがＥＣＢモードの場合における暗号演算のフローチャートが例示される。Ｅ０～Ｅ１０の処理は図７のステップＳｅ０～Ｓｅ１０の処理に対応され、その演算処理アルゴリズムは夫々等しくされる。図１５には図１４の処理の後の復号逆演算のフローチャートが例示される。ステップＳＴＰ１の判定結果が不一致であれば、故障解析攻撃を受けて暗号化演算途中で拡大鍵の改ざんが行われていることが明らかとなり、例外処理に移行される。図１７には上記暗号化処理において暗号化対象にされた平文と暗号化された暗号文の復号逆演算によって得られた平文との一致を検算する処理も行うようにしたときのフローチャートが例示される。

【００４６】

図１６には図１の回路において、鍵長が１２８ビット、暗号利用モードがＥＣＢモードの場合における復号演算のフローチャートが例示される。Ｄ０～Ｄ１０の処理は図７のステップＳｄ０～Ｓｄ１０の処理に対応され、その演算処理アルゴリズムは夫々等しくされる。図１８には上記復号処理において復号された平文の暗号化逆演算を行い、当初の暗号文と暗号化逆演算された暗号文との一致を検算する処理も行うようにしたときのフローチャートが例示される。

【００４７】

上記各タイミングチャートにおいて、AES_CSビットは制御レジスタのビットで、AESの暗号化の開始/終了の制御、ならびに暗号化終了時にAESコプロがAES_CSビットを変化させることで暗号化の演算状況を知らせるステータスビットとされる。

【００４８】

上記ＡＥＳコプロセッサによれば、暗号文から復号演算を実施する際、もしくは平文から暗号化計算後の復号逆演算を実施する際、暗号鍵から計算して求めてストアした拡大鍵に対して、故障攻撃されていないか確認するため、拡大鍵の逆算を実施し、その結果と保管されている拡大鍵もしくは暗号鍵と比較し、不一致の場合は例外処理信号により、リセット状態に遷移するなどの故障対策を講ずることができる。

【００４９】

図１９には図１のＡＥＳコプロセッサ７の第１の変形例が示される。図１９のＡＥＳコプロセッサ７Ａは比較用鍵レジスタ２５及び比較器２６が廃止されたＡＥＳ演算部２０Ａを有し、ＡＥＳ演算部２０Ａはそれに応ずる制御回路３３Ａを備える。この構成においては、復号化演算の開始時に鍵レジスタ２４と拡大鍵レジスタ３５に復号化演算で使用する最初の鍵を書き込んで復号化演算を実施する。拡大鍵レジスタ３５は拡大鍵計算器３４とのやりとりの結果、最終的に生成された暗号鍵が格納される。鍵レジスタ２４と拡大鍵レジスタ３５との間の拡大鍵の順次入れ替え制御は図１と同じである。この例においては、復号演算で最後に得られる拡大鍵を鍵レジスタ２４からＣＰＵがリードしてその正規性を判定する事によって故障攻撃の有無を検証する。そのほかの構成は図１と同じであるからその詳細な説明は省略する。

【００５０】

図１９の回路において、鍵長が128ビット、暗号利用モードがECBモードの場合における

暗号化演算を実施したときのフローチャートが図２０に例示され、そのときのタイミングチャートが図２２に例示される。

【００５１】

図１９の回路において、鍵長が128ビット、暗号利用モードがECBモードの場合における復号化演算を実施したときのフローチャートが図２１に例示され、そのときのタイミングチャートが図２３に例示される。復号化演算の際、最終ラウンドの鍵K10が生成されていない場合、図１６のように生成後演算を実施する。

【００５２】

ＡＥＳ演算回路３２で復号演算終了後、拡大鍵レジスタ３５の値と鍵レジスタ２４の値を入れ替え、生成された暗号鍵をＣＰＵ３が鍵レジスタ２４にアクセスすることでリードできるようにすることにより、演算中に故障注入されていないかを比較器なしで評価することができる。

【００５３】

尚、図１９において、暗号化演算処理の最後に得られる拡大鍵を鍵レジスタ２４からＣＰＵがリードしてその正規性を判定する事によって、暗号化演算処理での故障攻撃の有無を検証も可能である。その前提としてＣＰＵ３は鍵ＫのほかにＫ１０をＡＥＳコプロセッサ７Ａの外部で管理している。

【００５４】

図２４には図１のＡＥＳコプロセッサ７の第２の変形例が示される。図２４のＡＥＳコプロセッサ７Ｂは比較用鍵レジスタ２５に代えて拡大鍵保管レジスタ４０を有し、拡大鍵保管レジスタ４０の値も比較対象とする比較器４１を備え、ＡＥＳ演算部２０Ｂはそれに応ずる制御回路３３Ｂを有する。この構成においては、ＣＰＵ３は鍵Ｋの生成時、暗号鍵Ｋ以外の拡大鍵Ｋ１～Ｋ１０も生成して拡大鍵保管レジスタ４０に予めセットするように配送できるシステムを構築し、セキュア演算中は鍵から生成した拡大鍵を常に比較することで、故障注入を検出できるようにしたものである。そのほかの構成は図１と同じであるからその詳細な説明は省略する。図２４の回路において、鍵長が128ビット、暗号利用モードがECBモードの場合における暗号演算のフローチャートが図２５に例示され、復号演算のフローチャートが図２６に例示される。図２５のステップＳＴＯ１０において拡大鍵保管レジスタのラウンドｉの拡大鍵と拡大鍵レジスタの値が一致するか否かが判定される。図２６ではステップＳＴＯ２０において拡大鍵保管レジスタのラウンドｉの拡大鍵と拡大鍵レジスタの値が一致するか否かが判定される。ラウンド変数ｉが０のときは拡大鍵保管レジスタ４０の代わりに鍵レジスタ２４の値を用いる。

【００５５】

尚、図２４において、故障対策非考慮時は、比較器４１による検出を止め、拡大鍵を拡大鍵保管レジスタ４０からロードして用いることも可能である。また、拡大鍵保管レジスタ４０に保管する拡大鍵を最終ラウンドのみなどの一部とし、検算は保管した拡大鍵のみとし、拡大鍵計算結果をＡＥＳの演算回路の入力にすることも可能である。

【００５６】

図１９及び図２４の変形例に代表されるように、計算で求めた拡大鍵(暗号鍵含む)とストアされた拡大鍵との比較のタイミングは、毎ラウンド、計算の最後のみなど様々な方法が考えられる。最初の暗号化演算の際は必ずしも拡大鍵が計算されているわけではないが、既に拡大鍵が計算されレジスタにストアされている場合では、上記復号化演算と同様の検証が可能である。

【００５７】

《ＯＦＢモードにおける故障攻撃対策》

ＯＦＢモードにおける暗号化復号処理では図６に示されるように、暗号化も復号もＥで示される“暗号処理”のアルゴリズムを共通に用いる。図８にはこの点が更に詳細に示される、Ｅｎｃの符号化もＤｅｃの復号も、入力ＩＶに対して鍵Ｋを用いた暗号化処理（Ｅ）を共通に行い、暗号化ではその暗号化処理（Ｅ）結果Ｒに対して平文Ｐ０との排他的論理和を採って暗号文Ｃ０を生成し、復号ではその暗号化処理（Ｅ）結果Ｒに対して暗号文

Ｃ０との排他的論理和を採って平文Ｐ０を生成する。したがって暗号化に際して復号の逆演算を単純に行っても暗号化処理（Ｅ）が共通化されるので故障攻撃に対する検出の実効性は低い。暗号化処理（Ｅ）は例えば今までで説明したＥＣＢモードにおけるＥの暗号化処理と同じである。

【００５８】

図９にはＯＦＢモードの暗号化における検算処理を概念的に示している。値Ｉｉに対して演算処理（Ｅ）を行ってその結果Ｉｊに対して平文Ｐｉとの排他的論理輪を採って暗号文Ｃｉを生成し、Ｉｊ対して同じく演算処理（Ｅ）を行ってその結果Ｉｋに対して平文Ｐｊとの排他的論理輪を採って暗号文Ｃｊを生成するという処理を順次行うとき、Ｉｊに対する暗号化処理（Ｅ）に並行して、Ｉｊに逆演算すなわち復号処理（Ｄ）を行う。復号処理（Ｄ）は今までで説明したＥＣＢモードにおけるＤの復号処理と同じである。例えばＩｊに対する暗号化処理（Ｅ）において鍵はＫ０～Ｋ１０に自律的に演算処理されたとすると、Ｉｊに逆演算すなわち復号処理（Ｄ）では、Ｋ１０～Ｋ０の鍵を自律的な演算処理によって生成する。したがって、この一連の暗号化処理（Ｅ）と逆演算復号処理（Ｄ）は図１で説明した検算処理と実質的に同じになり、それと同様に、故障攻撃の有無の検証が可能になる。このように、暗号化処理（Ｅ）とその前の暗号化処理（Ｅ）の逆演算である復号処理（Ｄ）を並列化すると、その処理は例えば図２７に例示されるように、処理Ｐ１に並行して逆演算処理Ｒ１を行い、処理Ｐ２に並行して逆演算処理Ｒ２を行う。

【００５９】

図２８には図２７に示されるようにＯＦＢモードにおける自動逆算機能を有するＡＥＳコプロセッサ７Ｃが例示される。図１とはＡＥＳ演算部２０Ｃの構成が相違され、暗号化と逆算を並列的に行うために、２個の中間値レジスタ３１ａ，３１ｂと２個の拡大鍵レジスタ３５ａ，３５ｂを有し、更に比較用レジスタ５０、２個の比較器５１，５２を備え、それらを制御する制御回路３３Ｃが設けられる点が相違される。図２８の構成では、平文／暗号文と拡大鍵の両方の逆算結果について検証し、夫々の結果についてＡＥＳコプロセッサ７Ｃの外部に例外処理信号として出力する。ＥＯＲは排他的論理和回路である。

【００６０】

図２９及び図３０には図２８の回路において鍵長が１２８ビット、暗号利用モードがＯＦＢモードの場合における暗号化演算を実施したときのフローチャートが例示される。復号時のフローチャートは特に図示はしないが、図２９、図３０において平文Ｐ０のかわりに暗号文Ｃ０が、暗号文Ｃ０のかわりに平文Ｐ０が入る。

【００６１】

図３１には暗号演算の第１回目の動作のタイミングチャートが示され図２７の処理Ｐ０の暗号化処理（Ｅ）に対応される。図３２には暗号演算の第２回目の動作のタイミングチャートが例示され図２７の処理Ｐ１の暗号化処理（Ｅ）とＲ１の逆演算復号処理に対応される。各タイミングチャートは各レジスタ以外に制御レジスタ内にあるコントロール／ステータス信号であるAES_CSビットと、過去暗号演算の実施の有無(すなわち検算の有無)を示す検算ビットの状態が示される。

【００６２】

図３１に従えば、バス１０経由でAES_CSビットに”1”をライトすると、ＡＥＳ演算部２０Ｃが動作を開始する。鍵設定直後の演算につき、逆算の対象となる暗号文、拡大鍵の結果がでていないため、中間値レジスタ２（３１ｂ）、比較用レジスタ（５０）、ならびに拡大鍵レジスタ2（３５ｂ）は不定となっている。暗号演算につき拡大鍵が逐次生成されて拡大鍵レジスタ1（３５ａ）の値が更新されていくとともに、各ラウンドでの演算結果が中間値レジスタ1（３１ａ）に更新されて格納される。最終ラウンド終了後、中間値レジスタ1（３１ａ）に初期値IVの暗号化の結果であるI1が格納される。逆算をするため拡大鍵レジスタ2（３５ｂ）に最終ラウンドの拡大鍵K10、比較用レジスタ（５０）に初期値IVが書き込まれる。その後、初期値レジスタ（２１）にI1を、鍵レジスタ（２４）の値K0を拡大鍵レジスタ1（３５ａ）に、I1と平文P0の値からC0を計算し、平文／暗号文レジスタにそれぞれ書き込み、制御レジスタ（２３）内の検算ビットにフラグを立てる。

【００６３】
図３２に従えば、図３１の演算終了後、ユーザは平文／暗号文レジスタからＣ０入手しＰ１を書き込む。バス１経由でAES_CSビットに”１“をライトすると、ＡＥＳ演算部２０Ｃが動作を開始する。２回目はP1の暗号演算とC0の逆演算である復号演算が平行して処理される。この処理はパイプライン処理とされる。ＡＥＳ演算回路３２と拡大鍵計算器３４を暗号化処理と逆演算復号処理に共用しているからである。パイプライン処理により、それぞれのラウンドの拡大鍵が逐次生成されて両拡大鍵レジスタ３５ａ，３５ｂの値を更新していくとともに、各ラウンドでの暗号／復号演算結果が両中間値レジスタ３１ａ，３１ｂに格納される。最終ラウンド終了後、中間値レジスタ1（３１ａ）にI1の暗号化の結果であるI2が、中間値レジスタ2（３１ｂ）にI1の復号化の結果であるIVが格納される。ここで、最初の暗号演算ならびに逆算中に攻撃されていないか比較用レジスタ（５０）と中間値レジスタ2（３１ｂ）との値を比較器１（５１）で較し、異常があれば例外処理信号1によりＡＥＳプロセッサ７Ｃの外部に知らせる。同様に拡大鍵演算における攻撃がないか、鍵レジスタ(２４)と拡大鍵レジスタ2(３５ｂ)との値を比較器２（５２）で比較し、異常があれば例外処理信号２によりＡＥＳプロセッサ７Ｃの外部に知らせる。例えばシステムコントロールロジック４が例外処理信号１，２を受け取ることによってＩＣカード用マイコンをリセット状態などに遷移させることによって外部との情報を遮断し、或いは割り込みを発生させて本ＡＥＳの暗号化結果が有効でないことを知らせる、ことなどが考えられる。

【００６４】
検証後、次の演算／逆算を実施のため、拡大鍵レジスタに最終ラウンドの拡大鍵K10、比較用レジスタ５０に初期値I1が書き込まれる。その後、初期値レジスタ２１にI2を、鍵レジスタ２４の値K0を拡大鍵レジスタ1（３５ａ）に、I2と平文P1の値からC1を計算し、平文／暗号文レジスタ２２にそれぞれ書き込む。

【００６５】
以後の暗号演算文C2=EOR(AES(I2), P2)…においても同様の処理が実施される。最後の暗号文Cn=EOR(In, Pn)のについては、暗号対象となる平文はないもののＡＥＳ演算を追加で1回実施することで逆算による検証が可能であり、また、逆算だけ実行するモードを専用に設けることもできる。

【００６６】
上記において、平文／暗号文レジスタ２２に平文P0, P1…のかわりに暗号文C0、C1…を書き込むことで、OFBモードでの復号演算においても、暗号化処理（Ｅ）とその逆演算復号処理（Ｄ）によって同様の検算を行うことができる。

【００６７】
図２８の例はハードウェアで暗号／復号同時演算まで実施する回路構成としたが、単に初期値レジスタを入力として復号演算を実施できるハードウェアを追加することで、図２７のＲ１，Ｒ２などの逆演算をサポートすることも可能である。CFBモード、OFBモード、CRTモードとも初期値をＡＥＳコプロセッサの入力としているが、復号演算のときもAESコプロセッサは暗号化処理を行うから、それらの動作モードにおける故障攻撃対策においては、暗号化と復号の双方ともに、入力値の暗号化演算に並行して当該入力値への復号演算を行うことが特徴となる。また、図２８において比較回路５２を廃止し、ＣＰＵ３が鍵レジスタ２４をリードアクセスして鍵の正規性を判定するようにしてもよい。

【００６８】
以上説明した本発明の実施携帯によれば以下の作用効果を得る。

【００６９】
（１）逆算を行って検算を行う場合に、逆算結果だけでなく、逆算において最終的に用いた暗号鍵が当初の演算で最初に用いた暗号鍵に一致すかの検証を行う。これにより、例えば図４のような鍵スケジュールで攻撃されていないか確認すること可能である。

【００７０】
（２）拡大鍵をオンザフライ(On the fly)で毎回生成していくとき、復号演算時には最

終ラウンドの拡大鍵を元に暗号鍵を計算することになるので、生成された暗号鍵と保管された暗号鍵を比較することにすることで、図４のような誤った拡大鍵が使用されていないかが検出可能となる。暗号化時には暗号鍵は生成されないが、検算として逆算を実施することで拡大鍵の検算も取り入れることができる。暗号化時に鍵スケジュールを故障解析攻撃して誤暗号文を生成できたとしても、逆算による検算の際に検出できなくするためには暗号鍵から復号鍵を計算するときと、復号演算時において復号鍵から暗号鍵を計算するときの故障解析攻撃が必要となり、１回の暗号演算で２回の攻撃が必要となる。２回の攻撃タイミングや攻撃場所が違うことによる故障注入の難易度が上がるため、故障解析攻撃に対して検算による検出性能を向上させることができる。

【００７１】

（３）暗号利用モードがOFBモードなどの場合においても図６のようにAES演算回路で暗号化と復号演算を単に実施するのではなく、暗号化と復号の双方ともに、入力値の暗号化演算（Ｅ）に並行して当該入力値への復号演算（Ｄ）を行うから、図４のように２回故障を注入しても同じ誤りをおこすことが困難になり、故障解析攻撃に対して検算による検出性能を向上させることができる。

【００７２】

以上本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。

【００７３】

例えば、本発明に係るデータ処理装置はＩＣカード用マイコンに限定されず、ＲＦＩＤチップ、汎用マイクロコンピュータ等であってもよい。更に、本発明に係るデータ処理装置は１チップの半導体集積回路に限定されない。また、データ処理装置は公開暗号プロセッサを搭載しなくてもよい。本発明において暗号化復号を行うプロセッサはＤＥＳやＡＥＳに準拠する処理を行なうコプロセッサに限定されず、適宜変更可能である。

【図面の簡単な説明】

【００７４】

【図１】図１は共通鍵暗号プロセッサの一例としてＡＥＳコプロセッサの実装例を示すブロックダイヤグラムである。

【図２】図２は本発明の一例に係るＩＣカード用マイコンのブロックダイヤグラムである。

【図３】図３は拡大鍵レジスタへ一回攻撃を受けてその値が変化される場合に同じ演算を何回繰り返しても同じ結果しか得られず故障攻撃に対処することが出来ない場合を例示する説明図である。

【図４】図４は複数回計算の各回の演算において同一タイミングで同一計算をしている場合に毎回同じ故障注入を与えて同じ誤った値を生成させることで故障攻撃に対する検算が無効化される可能性のあることを示す説明図である。

【図５】図５は暗号利用モードとしてＥＣＢモードを示す説明図である。

【図６】図６は暗号利用モードとしてＯＦＢモードを示す説明図である。

【図７】図７はＥＣＢモードにおいける暗号化と復号の演算手法を例示する説明図である。

【図８】図８はＯＦＢモードにおいて暗号化も復号もＥで示される“暗号処理”のアルゴリズムを共通に用いる点を詳細に示した説明図である。

【図９】図９にはＯＦＢモードの暗号化における検算処理を概念的に例示する説明図である。

【図１０】図１０はＥＣＢモードで暗号化に際して行われる暗号化演算のタイミングチャートである。

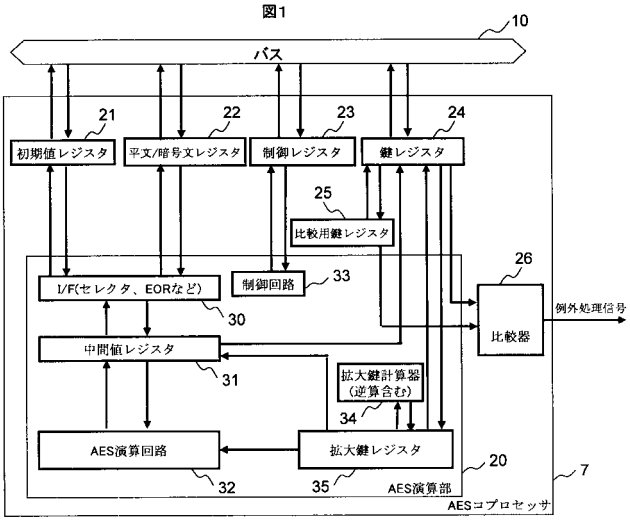
【図１１】図１１はＥＣＢモードで暗号化に際して行われる復号逆演算のタイミングチャートである。

【図１２】図１２はＥＣＢモードで復号に際しての動作タイミングチャートである。

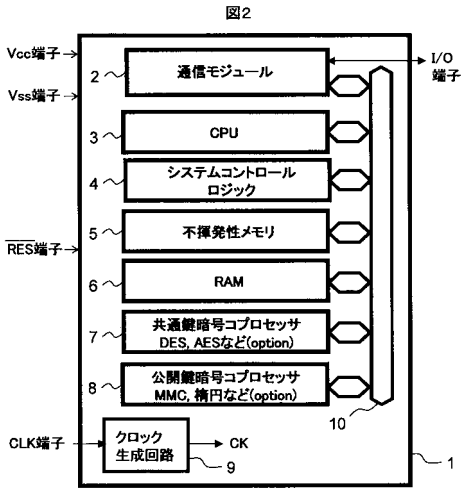
【図１３】図１３はＥＣＢモードの復号における予備的な拡大鍵演算処理で故障解析攻撃による拡大鍵の改ざんを受けたときの動作タイミングチャートである。
【図１４】図１４は図１の回路において、鍵長が１２８ビット、暗号利用モードがＥＣＢモードの場合における暗号演算のフローチャートである。
【図１５】図１５には図１４の処理の後の復号逆演算のフローチャートである。
【図１６】図１６には図１の回路において、鍵長が１２８ビット、暗号利用モードがＥＣＢモードの場合における復号演算のフローチャートである。
【図１７】図１７は暗号化処理において暗号化対象にされた平文と暗号化された暗号文の復号逆演算によって得られた平文との一致を検算する処理も行うようにしたときのフローチャートである。
【図１８】図１８は復号処理において復号された平文の暗号化逆演算を行い、当初の暗号文と暗号化逆演算された暗号文との一致を検算する処理も行うようにしたときのフローチャートである。
【図１９】図１９は図１のＡＥＳコプロセッサの第１の変形例を示すブロックダイヤグラムである。
【図２０】図２０は図１９の回路において、鍵長が128ビット、暗号利用モードがECBモードの場合における暗号化演算を実施したときのフローチャートである。
【図２１】図２１は図１９の回路において、鍵長が128ビット、暗号利用モードがECBモードの場合における復号化演算を実施したときのフローチャートである。
【図２２】図２２は図２０に対応されるタイミングチャートである。
【図２３】図２３は図２１に対応されるタイミングチャートである。
【図２４】図２４は図１のＡＥＳコプロセッサ７の第２の変形例を示すブロックダイヤグラムである。
【図２５】図２５は図２４の回路において、鍵長が128ビット、暗号利用モードがECBモードの場合における暗号演算のフローチャートである。
【図２６】図２６は図２５の復号演算のフローチャートである。
【図２７】図２７は暗号化処理（Ｅ）とその前の暗号化処理（Ｅ）の逆演算である復号処理（Ｄ）を並列化したとき処理Ｐ１に並行して逆演算処理Ｒ１を行い、処理Ｐ２に並行して逆演算処理Ｒ２を行う様子を例示する説明図である。
【図２８】図２８は図２７に示されるようにＯＦＢモードにおける自動逆算機能を有するＡＥＳコプロセッサ７Ｃを例示するブロックダイヤグラムである。
【図２９】図２９は図２８の回路において鍵長が１２８ビット、暗号利用モードがＯＦＢモードの場合における暗号化演算を実施したときのフローチャートである。
【図３０】図３０は図２９に続くフローチャートである。
【図３１】図３１は暗号演算の第１回目の動作のタイミングチャートであって図２７の処理Ｐ０の暗号化処理（Ｅ）に対応されるタイミングチャートである。
【図３２】図３２には暗号演算の第２回目の動作のタイミングチャートであって図２７の処理Ｐ１の暗号化処理（Ｅ）とＲ１の逆演算復号処理に対応されるタイミングチャートである。
【符号の説明】
【００７５】
１　ＩＣカード用マイコン
２　通信モジュール
３　中央処理装置（ＣＰＵ）
４　システムコントロールロジック
５　不揮発性メモリ
６　ＲＡＭ
７，７Ａ，７Ｂ，７Ｃ　共通鍵暗号プロセッサ
８　公開鍵暗号プロセッサ
９　クロック生成回路

１０　内部バス
２０，２０Ａ，２０Ｂ，２０Ｃ　ＡＥＳ演算部
２１　初期値レジスタ
２２　平文／暗号文レジスタ
２３　制御レジスタ
２４　鍵レジスタ
２５　比較用鍵レジスタ
２６，４１，５１，５２　比較器
３０　インタフェース回路（Ｉ／Ｆ）
３１，３１ａ，３１ｂ　中間値レジスタ
３２　ＡＥＳ演算回路
３３，３３Ａ，３３Ｂ，３３Ｃ　制御回路
３４　拡大鍵計算回路
３５，３５ａ，３５ｂ　拡大鍵レジスタ
４０　拡大鍵保管レジスタ
５０　比較用レジスタ

【図１】

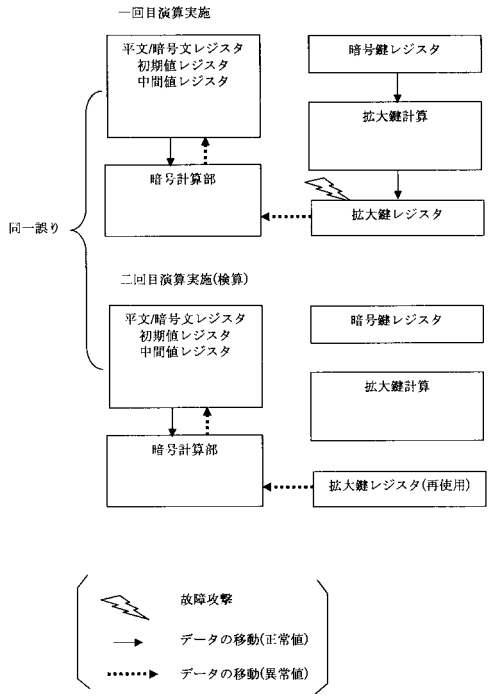


【図２】



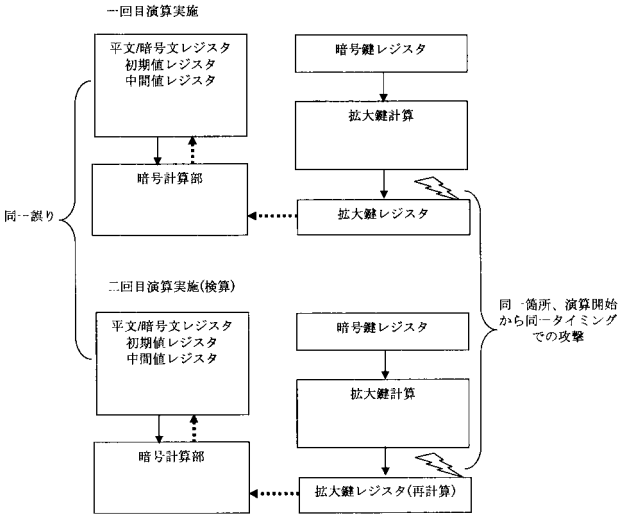
【図３】

図3



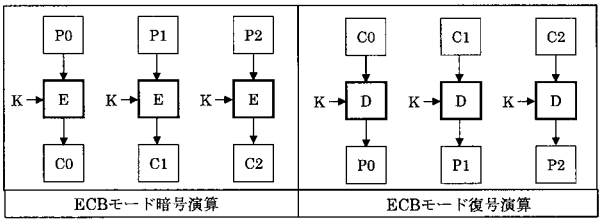
【図４】

図4



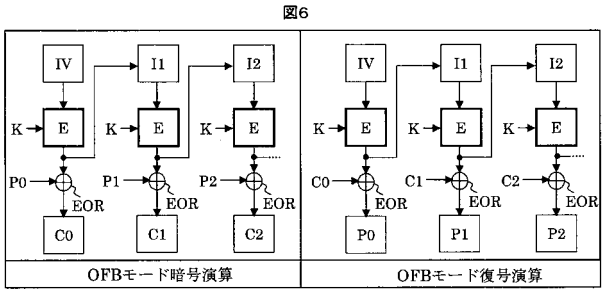
【図５】

図5



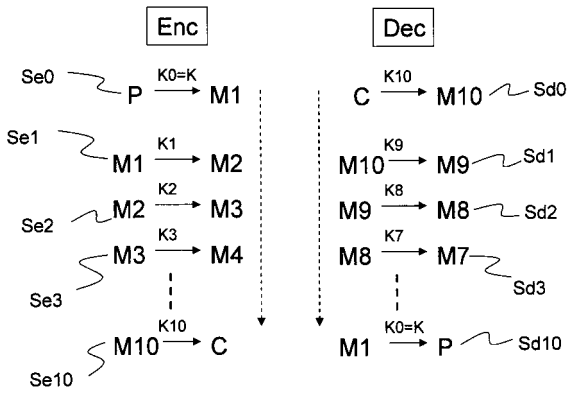
【図６】

図6



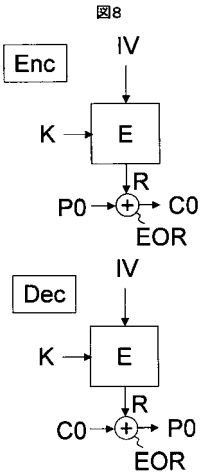
【図７】

図7



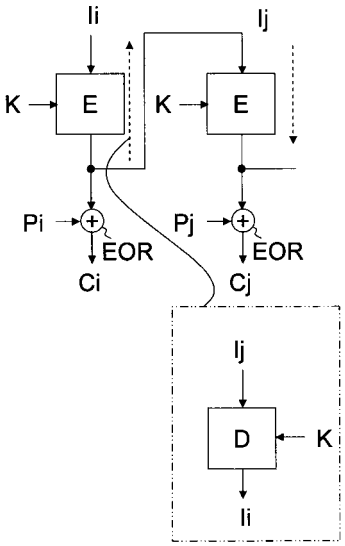
【図８】

図8



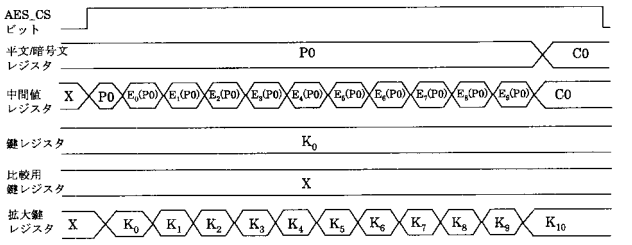
【図９】

図9



【図１０】

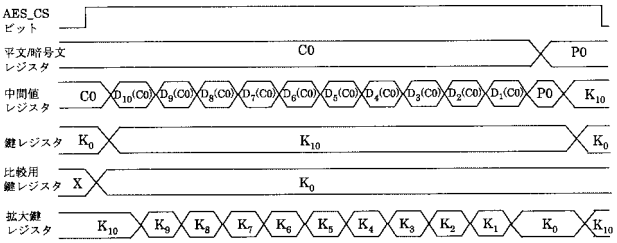
図10



K_n：暗号鍵Kに対するラウンドnの拡大鍵。K_0=K P0：ECBモードの平文 C0：ECBモードの暗号文
$E_n(P0)$：暗号鍵Kに対するラウンドn終了後の値。$E_{10}(P0)$ = C0 X：不定値(不定値から不定値の更新有)

【図１１】

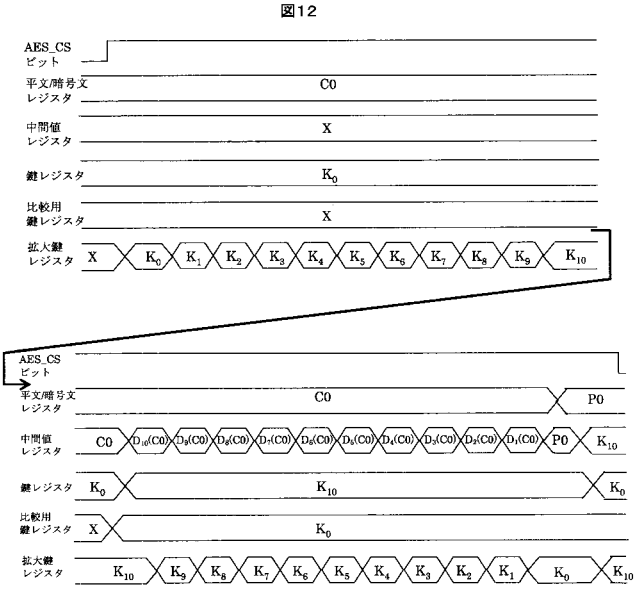
図11



K_n：暗号鍵Kに対するラウンドnの拡大鍵。K_0=K P0：ECBモードの平文 C0：ECBモードの暗号文
$D_n(C0)$：暗号鍵Kに対するラウンドn終了後の値。$D_0(C0)$ = P0

【図１２】

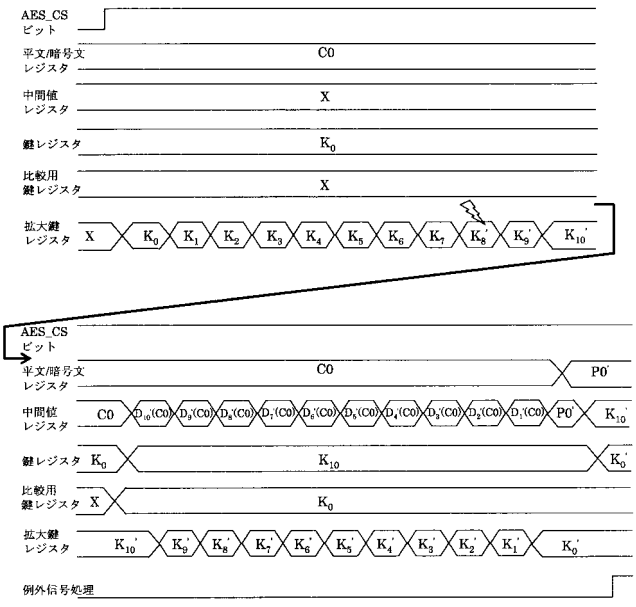
図12



K_n：暗号鍵Kに対するラウンドnの拡大鍵。K_0=K P0：ECBモードの平文 C0：ECBモードの暗号文
$D_n(C0)$：暗号鍵Kに対するラウンドn終了後の値。$D_0(C0)$ = P0
拡大鍵レジスタにK_{10}が保管されていない場合は、暗号鍵から計算後に復号演算

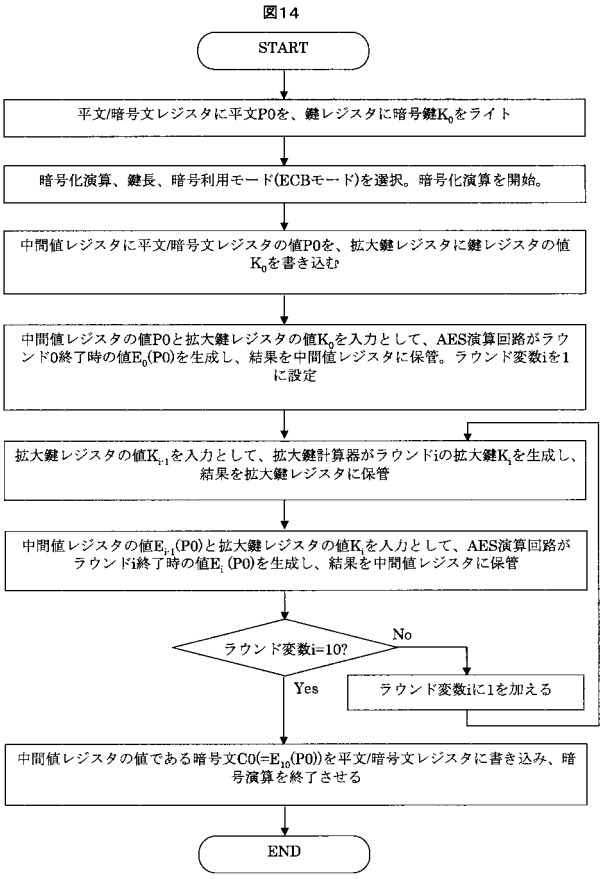
【図１３】

図13

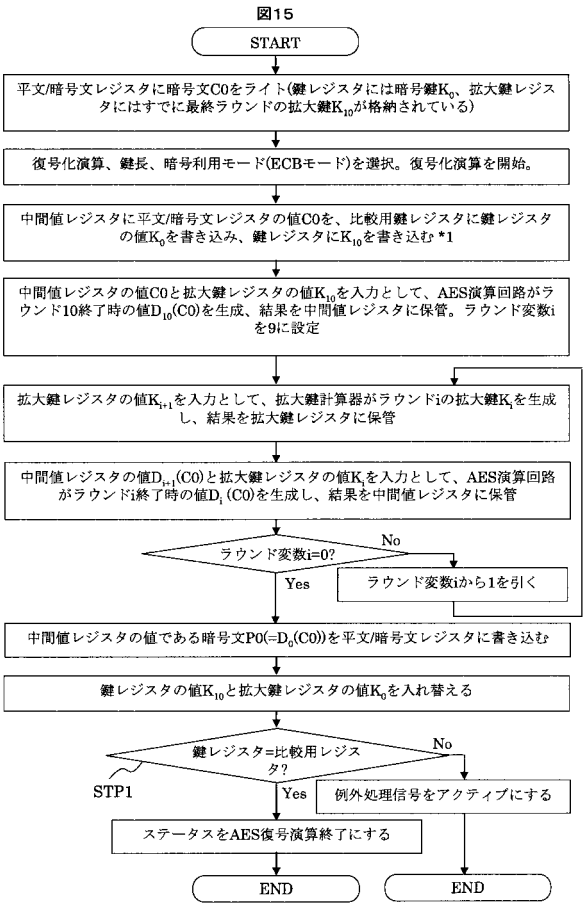


K_n：暗号鍵Kに対するラウンドnの拡大鍵。K_0=K
K_n'：故障注入により誤った値に変化したラウンドnの拡大鍵。$K_n' \neq K_n$
P0：ECBモードの平文 C0：ECBモードの暗号文
$D_n(C0)$：暗号鍵Kに対するラウンドn終了後の値。$D_0(C0)$ = P0
$D_n'(C0)$：誤ったラウンドnの拡大鍵K_{10}'～K_0'を用いたときの各ラウンド終了後の値。$D_0'(C0)$ = P0'

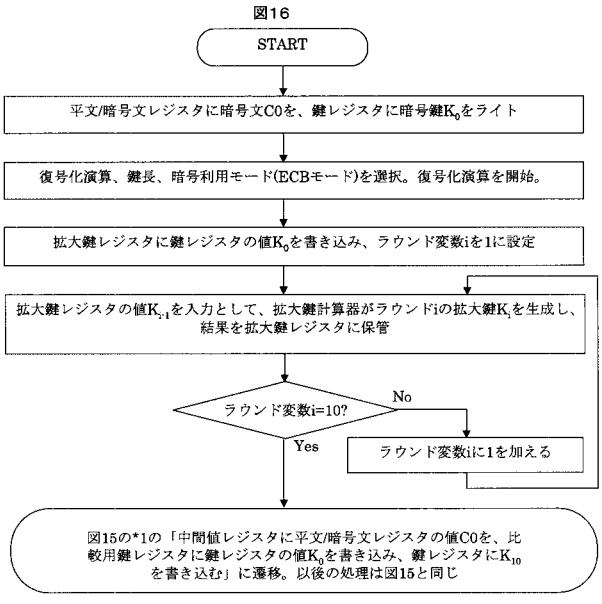
【図１４】



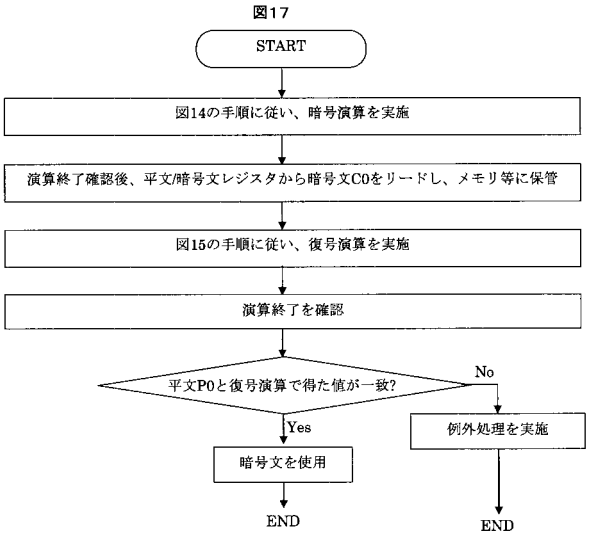
【図１５】



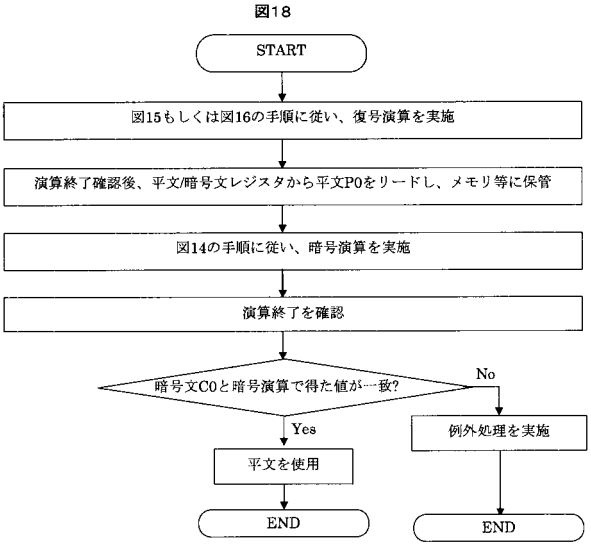
【図１６】



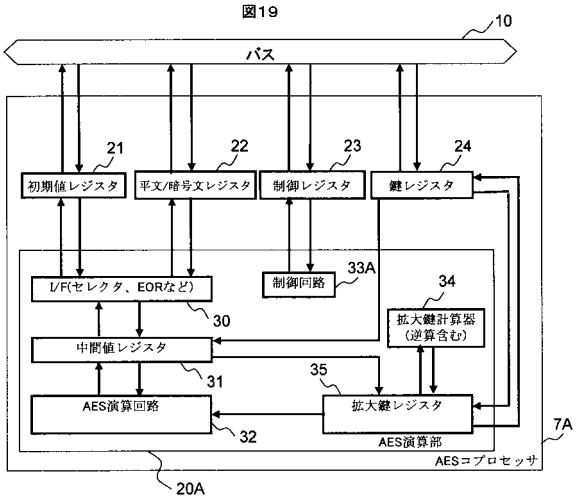
【図１７】



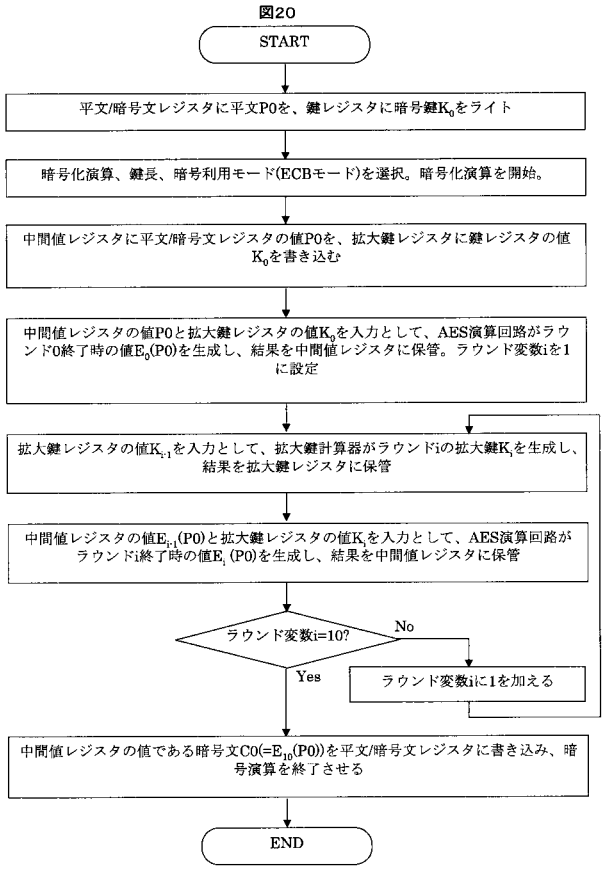
【図１８】



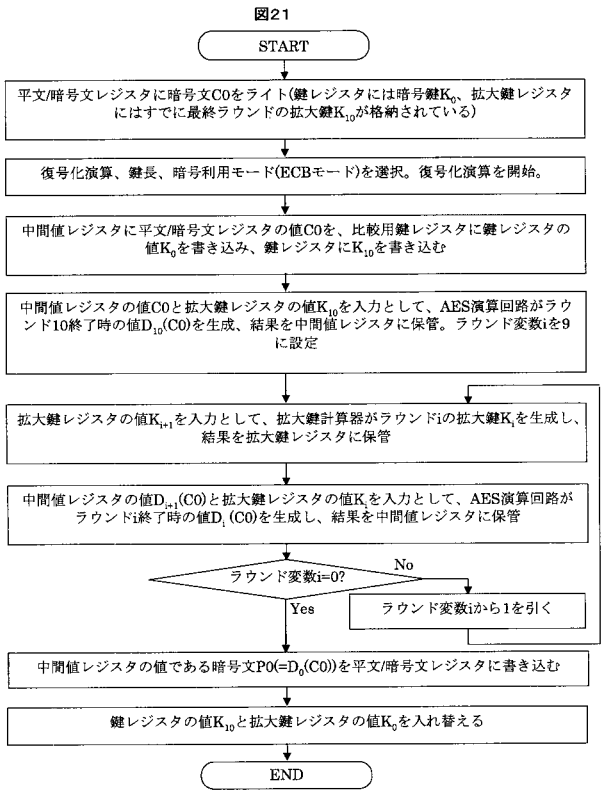
【図１９】



【図２０】

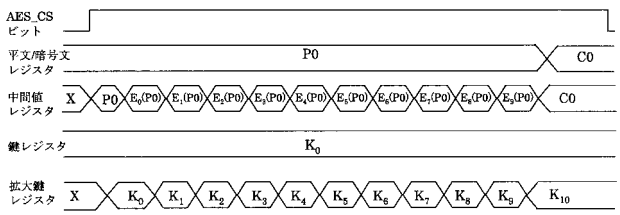


【図２１】



【図２２】

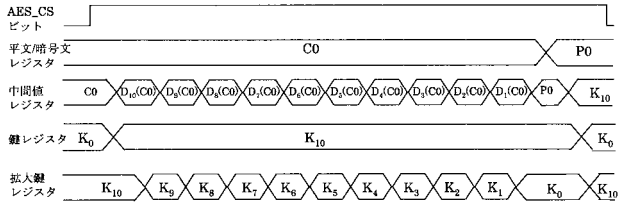
図22



K_n：暗号鍵Kに対するラウンドnの拡大鍵。K_0=K P0：ECBモードの平文 C0：ECBモードの暗号文
$E_n(P0)$：暗号鍵Kに対するラウンドn終了後の値。$E_{10}(P0)$ = C0 X：不定値(不定値から不定値の更新有)

【図２３】

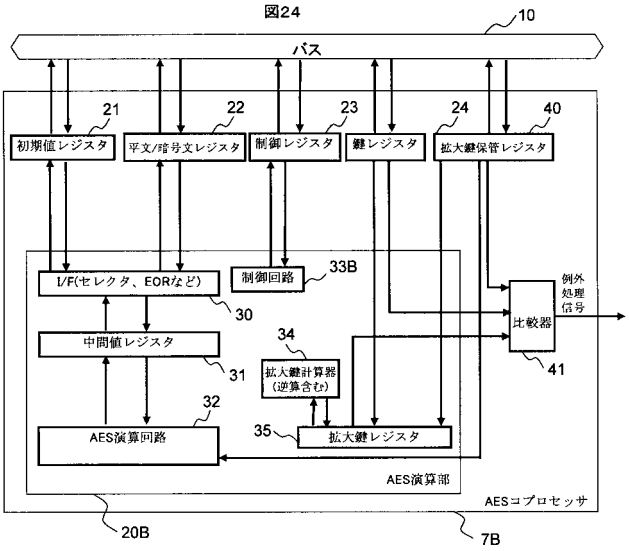
図23



K_n：暗号鍵Kに対するラウンドnの拡大鍵。K_0=K P0：ECBモードの平文 C0：ECBモードの暗号文
$D_n(C0)$：暗号鍵Kに対するラウンドn終了後の値。$D_0(C0)$ = P0
拡大鍵レジスタにK_{10}が保管されていない場合は、暗号鍵から計算後に復号演算

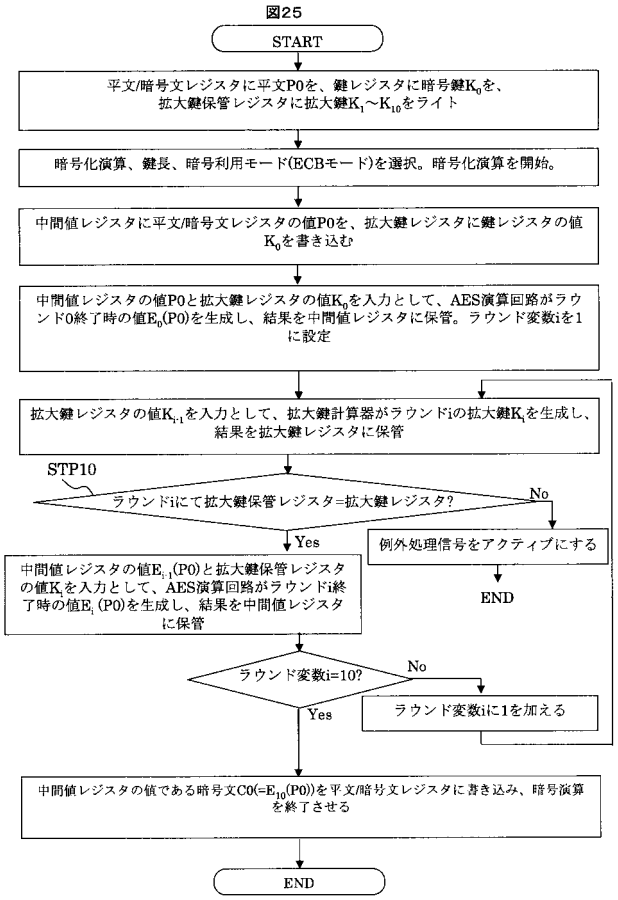
【図２４】

図24



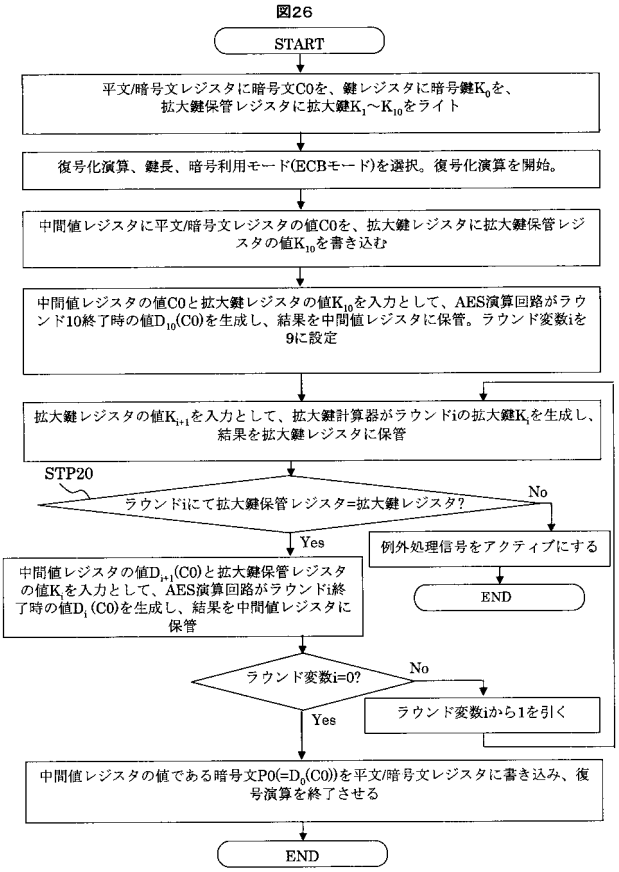
【図２５】

図25



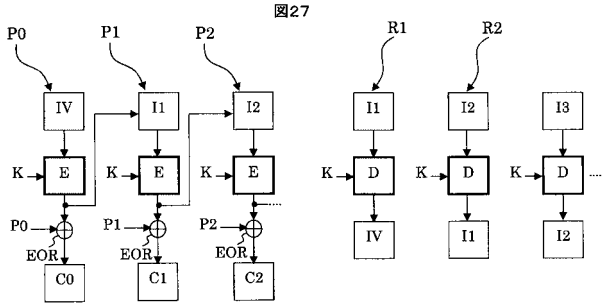
【図２６】

図26



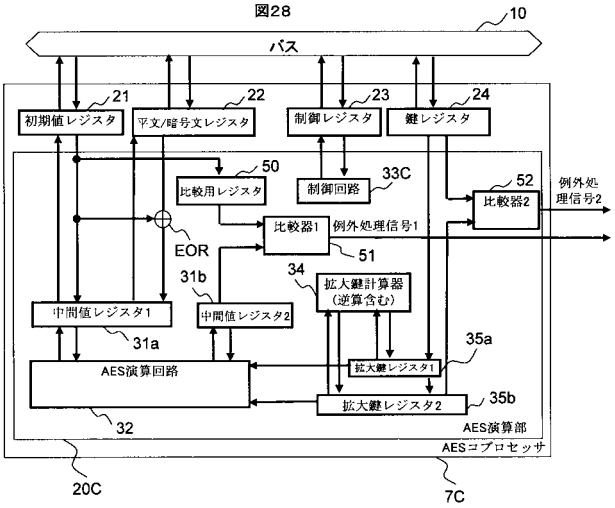
【図２７】

図27



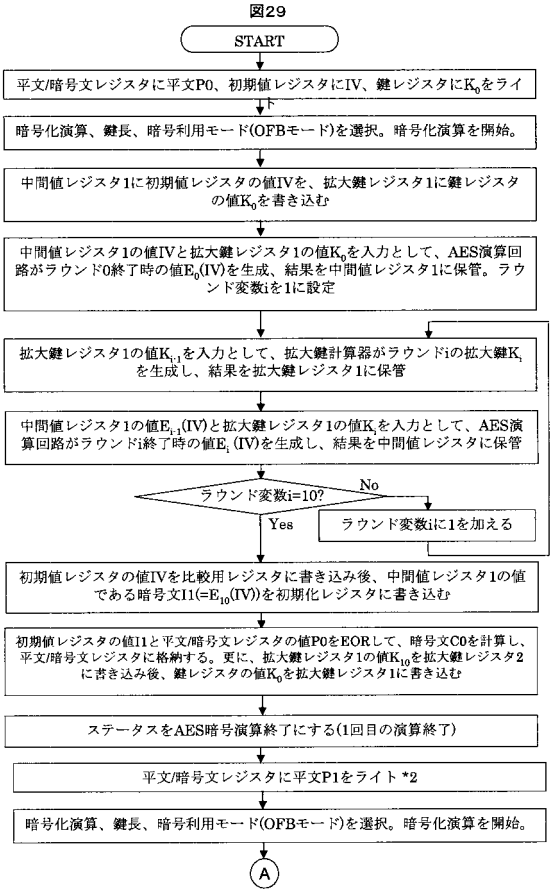
【図２８】

図28



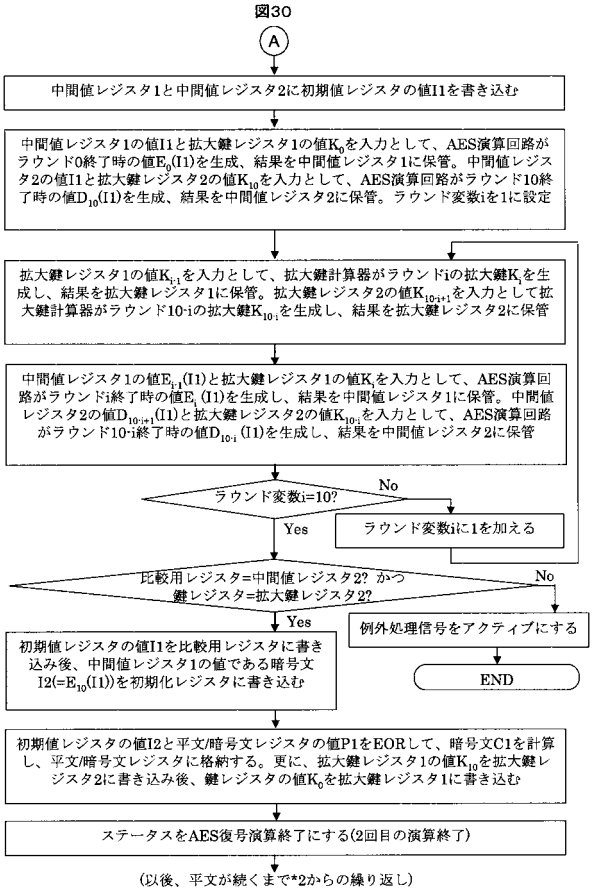
【図２９】

図29



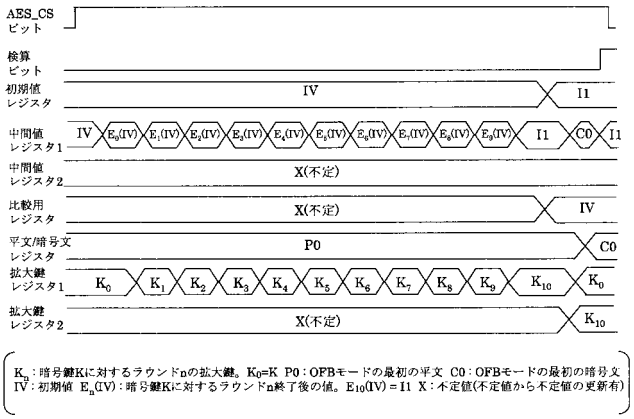
【図３０】

図30



【図３１】

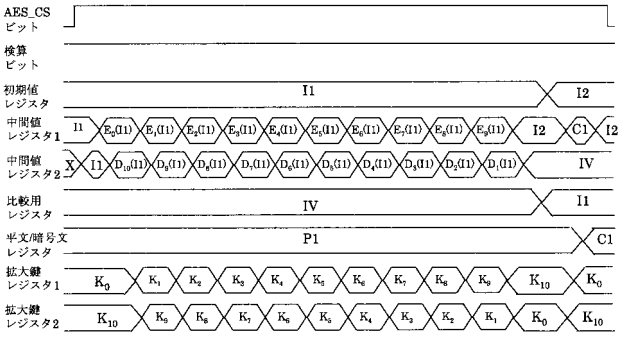
図31



K_n：暗号鍵Kに対するラウンドnの拡大鍵。K_0=K　P0：OFBモードの最初の平文　C0：OFBモードの最初の暗号文
IV：初期値　E_n(IV)：暗号鍵Kに対するラウンドn終了後の値。E_{10}(IV) = I1　X：不定値(不定値から不定値の更新有)

【図３２】

図32



P1：OFBモードの2番目の平文　C1：OFBモードの2番目の暗号文　X：不定値
E_n(I1)：暗号鍵Kに対する暗号演算のラウンドn終了後の値(AddRoundKeyの出力)。E_{10}(I1) = I2
D_n(I1)：暗号鍵Kに対する復号演算のラウンドn終了後の値(AddRoundKeyの出力)。D_0(I1) = IV