

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7656798号
(P7656798)

(45)発行日 令和7年4月4日(2025.4.4)

(24)登録日 令和7年3月27日(2025.3.27)

(51)国際特許分類

F I

G 0 9 C 1/00 (2006.01) G 0 9 C 1/00 6 2 0 Z

請求項の数 20 (全28頁)

(21)出願番号	特願2021-77713(P2021-77713)	(73)特許権者	390019839
(22)出願日	令和3年4月30日(2021.4.30)		三星電子株式会社
(65)公開番号	特開2021-177239(P2021-177239 A)		S a m s u n g E l e c t r o n i c s C o . , L t d .
(43)公開日	令和3年11月11日(2021.11.11)		大韓民国京畿道水原市靈通区三星路129
審査請求日	令和6年3月8日(2024.3.8)		129, S a m s u n g - r o , Y e o n g t o n g - g u , S u w o n - s i , G y e o n g g i - d o , R e p u b l i c o f K o r e a
(31)優先権主張番号	63/021,761		
(32)優先日	令和2年5月8日(2020.5.8)		
(33)優先権主張国・地域又は機関	米国(US)		
(31)優先権主張番号	10-2020-0139444	(73)特許権者	521081975
(32)優先日	令和2年10月26日(2020.10.26)		ソウル ナショナル ユニヴァーシティ
(33)優先権主張国・地域又は機関	韓国(KR)		アール アンド ディービー ファウンデーション
			大韓民国 08826 ソウル クアナッ
			最終頁に続く

(54)【発明の名称】 同型暗号を用いた暗号化方法及び装置

(57)【特許請求の範囲】

【請求項1】

プロセッサによって実行される同型暗号を用いた暗号化方法において、
データを暗号化することで暗号文を生成するステップと、
前記暗号文に対応するモジュラスに対して1つ以上のターゲット点の選択に基づいたモジュラーリダクションを行うことで、前記暗号文にブートストラップを行うステップと、
を含み、前記行うステップは、前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成するステップを行うことにより前記ブートストラップを行うステップを含み、
前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成するステップは、
前記ターゲット近似多項式の次数に基づいて1つ以上の基準点を決定するステップと、
前記1つ以上の基準点に基づいて任意の多項式を取得するステップと、
前記任意の多項式から選択された1つ以上の極点に基づいて決定される前記ターゲット点に基づき、前記ターゲット近似多項式を生成するステップと
を含む、暗号化方法。

【請求項2】

前記決定するステップは、前記1つ以上の基準点を通過する区分的連続関数を決定するステップと、
前記1つ以上の基準点で前記区分的連続関数との誤差の絶対値が予め決定された値にな

る多項式を生成することで前記任意の多項式を決定するステップと、
を含む、請求項 1 に記載の暗号化方法。

【請求項 3】

前記区分的連続関数との誤差の絶対値が、予め決定された値になる多項式を生成することで前記任意の多項式を決定するステップは、前記 1 つ以上の基準点に含まれた第 1 基準点における誤差と前記第 1 基準点と隣接する第 2 基準点の誤差の符号が異なり、絶対値が前記予め決定された値になる多項式を生成することで前記任意の多項式を決定するステップを含む、請求項 2 に記載の暗号化方法。

【請求項 4】

前記任意の多項式から選択された 1 つ以上の極点に基づいて前記ターゲット近似多項式を生成するステップは、

前記任意の多項式と前記 1 つ以上の基準点を通過する区分的連続関数間の誤差の極点のうち、絶対値が予め決定された値よりも大きいか同じ候補点を決定するステップと、

前記候補点のうち、前記度数に基づいた数のターゲット点を選択するステップと、

前記ターゲット点に基づいて前記ターゲット近似多項式を生成するステップと、

を含む、請求項 1 に記載の暗号化方法。

【請求項 5】

前記ターゲット点を選択するステップは、前記候補点のうち、極大と極小が交互に示されるように前記ターゲット点を選択するステップを含む、請求項 4 に記載の暗号化方法。

【請求項 6】

前記ターゲット点を選択するステップは、前記誤差の絶対値の和が最大になるよう前記ターゲット点を選択するステップを含む、請求項 4 に記載の暗号化方法。

【請求項 7】

前記任意の多項式から選択された 1 つ以上の極点に基づいて前記ターゲット近似多項式を生成するステップは、前記 1 つ以上の極点の絶対値の最大値と最小値の相対誤差が閾値よりも小さい場合の多項式を前記ターゲット近似多項式で生成するステップを含む、請求項 1 に記載の暗号化方法。

【請求項 8】

前記ターゲット近似多項式の基底は、チェビシェフ多項式の基底である、請求項 1 に記載の暗号化方法。

【請求項 9】

請求項 1 ～請求項 8 のいずれか一項に記載の暗号化方法を前記プロセッサに実行させるコンピュータプログラム。

【請求項 10】

同型暗号を用いた暗号化装置において、

データを暗号化することで暗号文を生成し、前記暗号文に対応するモジュラスに対して 1 つ以上のターゲット点の選択に基づいたモジュラーリダクションを行うことで、前記暗号文にブートストラップを行う 1 つ以上のプロセッサを含み、

前記 1 つ以上のプロセッサは、前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成することで前記ブートストラップを行い、

前記 1 つ以上のプロセッサは、前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成する場合において、

前記ターゲット近似多項式の度数に基づいて 1 つ以上の基準点を決定し、

前記 1 つ以上の基準点に基づいて任意の多項式を取得し、

前記任意の多項式から選択された 1 つ以上の極点に基づいて決定される前記ターゲット点に基づき、前記ターゲット近似多項式を生成する、暗号化装置。

【請求項 11】

前記 1 つ以上のプロセッサによって実行される命令を格納するメモリをさらに含み、

前記 1 つ以上のプロセッサは、前記命令を実行して前記データを暗号化することで暗号文を生成し、前記暗号文に対応するモジュラスに対して 1 つ以上のターゲット点の選択に

10

20

30

40

50

基づいたモジュラーリダクションを行うことで、前記暗号文にブートストラップを行う、請求項 10 に記載の暗号化装置。

【請求項 12】

前記 1 つ以上のプロセッサは、

前記 1 つ以上の基準点を通過する区分的連続関数を決定し、

前記 1 つ以上の基準点で前記区分的連続関数との誤差の絶対値が予め決定された値になる多項式を生成することで前記任意の多項式を決定する、請求項 10 に記載の暗号化装置。

【請求項 13】

前記 1 つ以上のプロセッサは、前記 1 つ以上の基準点に含まれた第 1 基準点における誤差と前記第 1 基準点と隣接する第 2 基準点の誤差の符号が異なり、絶対値が前記予め決定された値になる多項式を生成することで前記任意の多項式を決定する、請求項 12 に記載の暗号化装置。

10

【請求項 14】

前記 1 つ以上のプロセッサは、

前記任意の多項式と前記 1 つ以上の基準点を通過する区分的連続関数間の誤差の極点のうち、絶対値が予め決定された値よりも大きいか同じ候補点を決定し、

前記候補点のうち、前記度数に基づいた数のターゲット点を選択し、

前記ターゲット点に基づいて前記ターゲット近似多項式を生成する、請求項 10 に記載の暗号化装置。

【請求項 15】

前記 1 つ以上のプロセッサは、前記候補点のうち、極大と極小が交互に示されるように前記ターゲット点を選択する、請求項 14 に記載の暗号化装置。

20

【請求項 16】

前記 1 つ以上のプロセッサは、前記誤差の絶対値の和が最大になるように前記ターゲット点を選択する、請求項 14 に記載の暗号化装置。

【請求項 17】

前記 1 つ以上のプロセッサは、前記 1 つ以上の極点の絶対値の最大値と最小値の相対誤差が閾値よりも小さい場合の多項式を前記ターゲット近似多項式で生成する、請求項 10 に記載の暗号化装置。

【請求項 18】

前記ターゲット近似多項式の基底は、チェビシェフ多項式の基底である、請求項 10 に記載の暗号化装置。

30

【請求項 19】

同型暗号化を行う装置において、

暗号化されたデータの暗号文に対応するモジュラスに対して、1 つ以上のターゲット点の選択に基づいたモジュラーリダクションに対応する関数の近似を行うことで前記暗号化されたデータの暗号文をブートストラップし、

前記モジュラーリダクションに対応する前記関数を近似するターゲット近似多項式を生成する 1 つ以上のプロセッサを含み、

前記 1 つ以上のプロセッサは、前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成することで前記ブートストラップを行い、

40

前記 1 つ以上のプロセッサは、前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成する場合において、

前記ターゲット近似多項式の度数に基づいて 1 つ以上の基準点を決定し、

前記 1 つ以上の基準点に基づいて任意の多項式を決定し、

前記任意の多項式から選択された 1 つ以上の極点に基づいて決定される前記ターゲット点に基づき、前記ターゲット近似多項式を生成する、装置。

【請求項 20】

前記ターゲット近似多項式は、チェビシェフ交互定理 (Chebyshev alternation theorem) を用いて生成される、請求項 19 に記載の装置。

50

【発明の詳細な説明】

【技術分野】

【0001】

以下の実施形態は、同型暗号を用いた暗号化方法及び装置に関する。

【背景技術】

【0002】

完全同型暗号化 (fully homomorphic encryption) は、暗号化されたデータを用いた任意の論理的演算又は数学的演算を可能にする暗号化方式である。完全同型暗号化方式は、データ処理でセキュリティを保持可能にする。

【0003】

しかし、従来における暗号化方式は、暗号化されたデータの処理が難しいため、顧客のプライバシーの保護が不十分であった。

【0004】

完全同型暗号化は、顧客がプライバシーを守りながら多くのサービスを受けることを可能にする。

【発明の概要】

【発明が解決しようとする課題】

【0005】

本発明の目的は、同型暗号を用いた暗号化方法及び装置を提供することにある。

【課題を解決するための手段】

【0006】

プロセッサによって実行される同型暗号 (homomorphic encryption) を用いた暗号化方法において、一実施形態に係る暗号化方法は、データを暗号化することで暗号文を生成するステップと、前記暗号文に対応するモジュラスに対して1つ以上のターゲット点の選択に基づいたモジュラーリダクションを行うことで、前記暗号文にブートストラップを行うステップとを含む。

【0007】

前記行うステップは、前記モジュラーリダクションに対応する関数を近似することにより前記ブートストラップを行うステップを含むことができる。

【0008】

前記モジュラーリダクションに対応する関数を近似することで前記ブートストラップを行うステップは、前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成するステップを含むことができる。

【0009】

前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成するステップは、前記ターゲット近似多項式の次数に基づいて1つ以上の基準点を決定するステップと、前記ターゲット近似多項式の次数に基づいて1つ以上の基準点を決定するステップと、前記任意の多項式から選択された1つ以上の極点に基づいて前記ターゲット近似多項式を生成するステップとを含むことができる。

【0010】

前記決定するステップは、前記1つ以上の基準点を通過する区分的連続関数を決定するステップと、前記1つ以上の基準点で前記区分的連続関数との誤差の絶対値が予め決定された値になる多項式を生成することで前記任意の多項式を決定するステップとを含むことができる。

【0011】

前記区分的連続関数との誤差の絶対値が、予め決定された値になる多項式を生成することで前記任意の多項式を決定するステップは、前記1つ以上の基準点に含まれた第1基準点における誤差と前記第1基準点と隣接する第2基準点の誤差の符号が異なり、絶対値が前記予め決定された値になる多項式を生成することで前記任意の多項式を決定するステップを含むことができる。

10

20

30

40

50

【0012】

前記任意の多項式から選択された1つ以上の極点に基づいて前記ターゲット近似多項式を生成するステップは、前記任意の多項式と前記1つ以上の基準点を通過する区分的連続関数間の誤差の極点のうち、絶対値が予め決定された値よりも大きいか同じ候補点を決定するステップと、前記候補点のうち、前記次数に基づいた数のターゲット点を選択するステップと、前記ターゲット点に基づいて前記ターゲット近似多項式を生成するステップとを含むことができる。

【0013】

前記ターゲット点を選択するステップは、前記候補点のうち、極大と極小が交互に示されるように前記ターゲット点を選択するステップを含むことができる。

10

【0014】

前記ターゲット点を選択するステップは、前記誤差の絶対値の和が最大になるよう前記ターゲット点を選択するステップを含むことができる。

【0015】

前記任意の多項式から選択された1つ以上の極点に基づいて前記ターゲット近似多項式を生成するステップは、前記1つ以上の極点の絶対値の最大値と最小値の相対誤差が閾値よりも小さい場合の多項式を前記ターゲット近似多項式で生成するステップを含むことができる。

【0016】

前記ターゲット近似多項式の基底は、チェビシェフ多項式の基底であってもよい。

20

【0017】

同型暗号を用いた暗号化装置において、データを暗号化することで暗号文を生成し、前記暗号文に対応するモジュラスに対して1つ以上のターゲット点の選択に基づいたモジュラーリダクションを行うことで、前記暗号文にブートストラップを行う1つ以上のプロセッサを含む。

【0018】

前記暗号化装置は、前記1つ以上のプロセッサによって実行される命令を格納するメモリをさらに含み、前記1つ以上のプロセッサは、前記命令を実行して前記データを暗号化することで暗号文を生成し、前記暗号文に対応するモジュラスに対して1つ以上のターゲット点の選択に基づいたモジュラーリダクションを行うことで前記暗号文にブートストラップを行うことができる。

30

【0019】

前記1つ以上のプロセッサは、前記モジュラーリダクションに対応する関数を近似することで前記ブートストラップを行うことができる。

【0020】

前記1つ以上のプロセッサは、前記モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成することができる。

【0021】

前記1つ以上のプロセッサは、前記ターゲット近似多項式の次数に基づいて1つ以上の基準点を決定し、前記1つ以上の基準点に基づいて任意の多項式を取得し、前記任意の多項式から選択された1つ以上の極点に基づいて前記ターゲット近似多項式を生成することができる。

40

【0022】

前記1つ以上のプロセッサは、前記1つ以上の基準点を通過する区分的連続関数を決定し、前記1つ以上の基準点で前記区分的連続関数との誤差の絶対値が、予め決定された値になる多項式を生成することで前記任意の多項式を決定することができる。

【0023】

前記1つ以上のプロセッサは、前記1つ以上の基準点に含まれた第1基準点における誤差と前記第1基準点と隣接する第2基準点の誤差の符号が異なり、絶対値が前記予め決定された値になる多項式を生成することで前記任意の多項式を決定することができる。

50

【0024】

前記1つ以上のプロセッサは、前記任意の多項式と前記1つ以上の基準点を通過する区分的連続関数間の誤差の極点のうち、絶対値が予め決定された値よりも大きいか同じ候補点を決定し、前記候補点のうち、前記次数に基づいた数のターゲット点を選択し、前記ターゲット点に基づいて前記ターゲット近似多項式を生成することができる。

【0025】

前記1つ以上のプロセッサは、前記候補点のうち、極大と極小が交互に示されるように前記ターゲット点を選択することができる。

【0026】

前記1つ以上のプロセッサは、前記誤差の絶対値の和が最大になるように前記ターゲット点を選択することができる。

10

【0027】

前記1つ以上のプロセッサは、前記1つ以上の極点の絶対値の最大値と最小値の相対誤差が閾値よりも小さい場合の多項式を前記ターゲット近似多項式で生成することができる。

【0028】

前記ターゲット近似多項式の基底は、チェビシェフ多項式の基底であってもよい。

【0029】

同型暗号化を行う装置において、暗号化されたデータの暗号文に対応するモジュラスに対して、1つ以上のターゲット点の選択に基づいたモジュラーリダクションに対応する関数の近似を行うことで前記暗号化されたデータの暗号文をブートストラップし、前記モジュラーリダクションに対応する前記関数を近似するターゲット近似多項式を生成する1つ以上のプロセッサとを含む。

20

【0030】

前記近似多項式は、チェビシェフ交互定理 (Chebyshev alternation theorem) を用いて生成されることができる。

【0031】

前記1つ以上のプロセッサは、前記ターゲット近似多項式の次数に基づいて1つ以上の基準点を決定し、前記1つ以上の基準点に基づいて任意の多項式を決定し、前記任意の多項式から選択された1つ以上の極点に基づいてターゲット近似多項式を生成することができる。

30

【発明の効果】

【0032】

本発明によると、同型暗号を用いた暗号化方法及び装置を提供することができる。

【図面の簡単な説明】

【0033】

【図1】一実施形態に係る暗号化装置の概略的なブロック図を示す。

【図2A】図1に示された暗号化装置がターゲット近似多項式を生成するアルゴリズムの一例を示す。

【図2B】図1に示された暗号化装置がターゲット近似多項式を生成するアルゴリズムの他の例を示す。

40

【図2C】図1に示された暗号化装置がターゲット近似多項式を生成する動作のフローチャートを示す。

【図3】図1に示された暗号化装置が極点を探索する動作のフローチャートを示す。

【図4A】図1に示された暗号化装置がターゲット点を選択するアルゴリズムの例を示す。

【図4B】図1に示された暗号化装置がターゲット点を選択する動作のフローチャートを示す。

【図5】図1に示された暗号化装置の全体動作の順序を示す。

【発明を実施するための形態】

【0034】

以下、添付の図面を参照して実施形態を詳細に説明する。しかし、実施形態には多様な

50

変更が加えられることができ、特許出願の権利範囲がこの実施形態により制限されたり限定されることはない。実施形態に対するすべての変更、均等物ないし代替物が権利範囲に含まれるものとして理解されなければならない。

【0035】

本明細書で用いる用語は、単に特定の実施形態を説明するために用いられるものであって、本発明を限定しようとする意図はない。単数の表現は、文脈上、明白に異なる意味をもたない限り複数の表現を含む。本明細書において、「含む」又は「有する」等の用語は明細書上に記載した特徴、数字、ステップ、動作、構成要素、部品、又はこれらを組み合わせたものが存在することを示すものであって、一つ又はそれ以上の他の特徴や数字、ステップ、動作、構成要素、部品、又はこれらを組み合わせたものなどの存在又は付加の可能性を予め排除しないものとして理解しなければならない。

10

【0036】

異なるように定義されない限り、技術的であるか又は科学的な用語を含むここで用いる全ての用語は、本実施形態が属する技術分野で通常の知識を有する者によって一般的に理解されるものと同じ意味を有する。一般的に用いられる予め定義された用語は、関連技術の文脈上で有する意味と一致する意味を有するものと解釈すべきであって、本明細書で明白に定義しない限り、理想的又は過度に形式的な意味として解釈されることはない。

【0037】

また、図面を参照して説明する際に、図面符号に拘わらず同じ構成要素は同じ参照符号を付与し、これに対する重複する説明は省略する。実施形態の説明において関連する公知技術に対する具体的な説明が本発明の要旨を不要に曖昧にすると判断される場合、その詳細な説明は省略する。

20

【0038】

また、実施例の構成要素の説明において、第1、第2、A、B、(a)、(b)などの用語を使用することができる。このような用語は、その構成要素を他の構成要素と区別するためのものであって、その用語によって該当の構成要素の本質や順番又は順序などが限定されることはない。いずれかの構成要素が他の構成要素に「連結」、「結合」又は「接続」されたと記載される場合、その構成要素はその他の構成要素に直接的に連結又は接続され得るが、各構成要素の間に更なる構成要素が「連結」、「結合」又は「接続」され得ると理解されなければならないのであろう。

30

【0039】

いずれかの1つの実施形態に含まれている構成要素と、共通的な機能を含む構成要素は、他の実施形態において同じ名称を用いて説明することにする。反対となる記載がない以上、いずれか1つの実施形態に記載した説明は、他の実施形態にも適用されてもよく、重複する範囲で具体的な説明は省略することにする。

【0040】

図1は、一実施形態に係る暗号化装置の概略的なブロック図を示す。

【0041】

図1を参照すると、暗号化(encryption)装置10は、データの暗号化を行う。暗号化装置10は、データの暗号化を介して暗号化されたデータを生成する。以下において、暗号化されたデータは、暗号文(ciphertext)と称される。

40

【0042】

暗号化装置10は、同型暗号を用いて暗号化されたデータを復号化することなく、演算できる暗号技術を提供することができる。例えば、暗号化装置10は、同型暗号を用いて暗号化された状態で演算した結果を復号化することで、平文(plaintext)状態のデータを演算した結果と同じ結果が導き出される。暗号化装置10は、任意の実数又は複素数に対する同型暗号演算を提供することができる。

【0043】

暗号化装置10は、同型暗号(homomorphic encryption)に必要なブートストラップ(bootstrapping)を行ってもよい。暗号化装置10

50

は、同型暗号に求められるモジュラーリダクション (modular reduction) に対応する関数を近似するためにターゲット近似多項式を生成する。

【0044】

暗号化装置10は、最適な最小最大近似多項式 (optimal minimax approximate polynomial) の各次数に対して、最小最大近似誤差 (minimax approximation error) を算出することができる。

【0045】

暗号化装置10は、最適近似多項式をターゲット近似多項式により探すことで、同型暗号方式の最小最大近似誤差の観点で優れた性能を提供することができる。

【0046】

暗号化装置10は、モジュラーリダクション関数を近似するための近似領域情報を用いて、モジュラーリダクション関数を近似するターゲット近似多項式を生成する。

【0047】

暗号化装置10は、プロセッサ100及びメモリ200を含む。

【0048】

プロセッサ100は、メモリ200に格納されたデータを処理する。プロセッサ100は、メモリ200に格納されたコンピュータで読み出し可能なコード (例えば、ソフトウェア) 及びプロセッサ100によって誘発された命令 (instruction) を実行する。

【0049】

「プロセッサ100」は、目的とする動作を実行させるための物理的な構造を有する回路を有するハードウェアで具現されたデータ処理装置である。例えば、目的とする動作は、プログラムに含まれているコード又は命令を含んでもよい。

【0050】

例えば、ハードウェアで具現されたデータ処理装置は、マイクロプロセッサ、中央処理装置、プロセッサコア、マルチコアプロセッサ、マルチプロセッサ、ASIC (Application-Specific Integrated Circuit)、FPGA (Field Programmable Gate Array) を含んでもよい。

【0051】

プロセッサ100は、データを暗号化することで暗号文を生成する。プロセッサ100は、生成した暗号文に対応するモジュラスに対して、モジュラーリダクション (modular reduction) を行うことで暗号文にブートストラップを行ってもよい。

【0052】

プロセッサ100は、モジュラーリダクションに対応する関数を近似することによりブートストラップを行う。プロセッサ100は、モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成する。

【0053】

プロセッサ100は、ターゲット近似多項式の次数に基づいて1つ以上の基準点を決定することができる。

【0054】

プロセッサ100は、決定された1つ以上の基準点に基づいて任意の多項式を決定する。プロセッサ100は、1つ以上の基準点を通過する区分的連続 (piecewise continuous) 関数を決定する。プロセッサ100は、1つ以上の基準点で区分的連続関数との誤差の絶対値が、予め決定された値になる多項式を生成することで、任意の多項式を決定することができる。

【0055】

プロセッサ100は、1つ以上の基準点に含まれた第1基準点における誤差と、第1基準点に隣接する第2基準点の誤差の符号が異なり、絶対値が予め決定された値になる多項式を生成することで、任意の多項式を決定することができる。

【0056】

10

20

30

40

50

プロセッサ100は、任意の多項式から選択された1つ以上の極点（`extremum point`）に基づいてターゲット近似多項式を生成する。具体的に、プロセッサ100は、任意の多項式と1つ以上の基準点を通過する区分的連続関数間の誤差の極点のうち、絶対値が予め決定された値よりも大きいか同じ候補点を決定することができる。

【0057】

プロセッサ100は、決定した候補点のうち、ターゲット近似多項式の次数に基づいた数のターゲット点を選択する。プロセッサ100は、候補点のうち、極大と極小が交互に示されるようにターゲット点を選択してもよい。プロセッサ100は、任意の多項式と1つ以上の基準点を通過する区分的連続関数間の誤差の絶対値の和が最大になるようターゲット点を選択する。

【0058】

プロセッサ100は、選択したターゲット点に基づいてターゲット近似多項式を生成する。プロセッサ100は、1つ以上の極点の絶対値の最大値と最小値の相対誤差が閾値よりも小さい場合の多項式を、ターゲット近似多項式で生成することができる。

【0059】

ここで、ターゲット近似多項式の基底は、チェビシェフ（`Chebyshev`）多項式の基底であってもよい。

【0060】

メモリ200は、プロセッサによって実行可能な命令（又は、プログラム）を格納する。例えば、命令は、プロセッサの動作及び／又はプロセッサの各構成の動作を実行するための命令を含んでもよい。

【0061】

メモリ200は、揮発性メモリ装置又は不揮発性メモリ装置に具現されてもよい。

【0062】

揮発性メモリ装置は、DRAM（`dynamic random access memory`）、SRAM（`static random access memory`）、T-RAM（`thyristor RAM`）、Z-RAM（`zero capacitor RAM`）、又はTTRAM（`Twin Transistor RAM`）で具現される。

【0063】

不揮発性メモリ装置は、EEPROM（`Electrically Erasable Programmable Read-Only Memory`）、フラッシュ（`flash`）メモリ、MRAM（`Magnetic RAM`）、スピン伝達トルクMRAM（`Spin-Transfer Torque (STT)-MRAM`）、Conductive Bridging RAM（`CBRAM`）、FeRAM（`Ferroelectric RAM`）、PRAM（`Phase change RAM`）、抵抗メモリ（`Resistive RAM (RRAM)`）、ナノチューブRRAM（`Nanotube RRAM`）、ポリマーRAM（`Polymer RAM (PoRAM)`）、ナノ浮遊ゲートメモリ（`Nano Floating Gate Memory (NFGM)`）、ホログラフィックメモリ（`holographic memory`）、分子電子メモリ素子（`Molecular Electronic Memory Device`）、又は、絶縁抵抗変化メモリ（`Insulator Resistance Change Memory`）に具現することができる。

【0064】

以下において、図2A～図2Cを参照して暗号化装置10が暗号化及びブートストラップを行う過程について詳細に説明する。以下、暗号化装置10が行う暗号化動作の説明後も、ブートストラップ過程についてより詳細に説明する。

【0065】

図2A及び図2Bは、図1に示された暗号化装置がターゲット近似多項式を生成するアルゴリズムの例を示し、図2Cは、図1に示された暗号化装置がターゲット近似多項式を生成する動作のフローチャートを示す。

10

20

30

40

50

【0066】

図2A～図2Cを参照すると、プロセッサ100は、データに対する暗号化を行ってもよい。

【0067】

以下、プロセッサ100の暗号化動作を説明するための表記について説明する。

【0068】

【数1】

$\mathbb{Z}$ 、 $\mathbb{Q}$ 、 $\mathbb{R}$ 及び $\mathbb{C}$ はそれぞれ整数、有理数、実数及び複素数の集合を示す。

$C[D]$ は、ドメイン D 上の連続関数の集合を意味する。

$[d]$ は、 d 以下の正の整数の集合を意味する。例えば、 $[d]$ は $\{1, 2, \dots, d\}$ を意味する。

$\text{round}(x)$ は、 x に最も近い整数を出力する関数を意味する。

2の累乗である M に対して、 $\Phi_M(X) = X^N + 1$ は M 番目の円分多項式(cyclotomic polynomial)を意味し、 $M = 2N$ であってもよい。

$\mathcal{R}$ 及び $\mathcal{R}_q$ は、それぞれ $\mathcal{R} = \mathbb{Z}[X]/\langle\Phi_M(X)\rangle$ 及び $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ を意味する。

$\mathbb{Q}[X]/\langle\Phi_M(X)\rangle$ は、 M 番目の円分体(cyclotomic field)を意味する。

正の実数 α に対して、 $\mathcal{DG}(\alpha)$ は $\mathbb{Z}^N$ における分布として定義され、そのエントリ(entries)は、分散 $\alpha/2$ の離散ガウス分布(discrete gaussian distribution)から独立にサンプリングされる。

$\text{HWT}(h)$ は、ハミング荷重値(Hamming weight) h を有する $\{0, \pm 1\}^N$ の部分集合を意味する。

$\mathcal{ZO}(\rho)$ は、 $\{0, \pm 1\}^N$ における分布を意味し、そのエントリは、それぞれの ± 1 に対して $\rho/2$ の確率を有し、0になる確率として $1 - \rho$ を有し、独立にサンプリングされることができる。

チェビシエフ多項式 $T_n(x)$ は、 $\cos n\theta = T_n(\cos \theta)$ に定義され、以下で記載されるログ(logarithm)のベース(base)は2である。

以下、プロセッサが行う暗号化動作について説明する。

【0069】

プロセッサ100は、実数又は複素数の暗号化されたデータに対する数個の演算を支援する。暗号化装置10は、主に実数を扱っているため、暗号化方式のセキュリティを保障するノイズがデータの重要な要素の外部に受容されされてもよい。

【0070】

数個の独立的なメッセージが暗号化前に基本埋め込み(canonical embedding)によって1つの多項式で符号化される。

【0071】

【数2】

10

20

30

40

50

基本埋め込み σ は、 $a \in \mathbb{Q}[X]/\langle \Phi_M(X) \rangle$ を $\mathbb{C}^N$ の元素に格納し、 $\mathbb{C}^N$ の元素は、 $\Phi_M(X)$ の個別根で求められた a の値を意味する。

$\Phi_M(X)$ の根は、M 累乗根 (M-th root of unity) の奇数の累乗 (power of odd integer) であり、 $\mathbb{Z}_M^* = \langle -1, 5 \rangle$ であってもよい。

$\mathbb{H}$ は $\mathbb{H} = \{(z_j)_{j \in \mathbb{Z}_M^*} : z_j = \overline{z_{-j}}\}$ であり、

π は $\mathbb{H}$ から $\mathbb{C}^{N/2}$ に対する自然投影 (natural projection) を意味する。

σ の範囲は $\mathbb{H}$ であってもよい。

$N/2$ 複素数メッセージは $\mathbb{C}^{N/2}$ で元素を構成し、それぞれの座標はスロット (slot) と称される。符号化及び復号化の過程は、下記のように与えられる。

ベクトル $z \in \mathbb{C}^{N/2}$ に対して、符号化 $\text{Ecd}(z; \Delta)$ は数式 (1) の値を返す (return) ことができる。

$$m(X) = \sigma^{-1} \left(\lfloor \Delta \cdot \pi^{-1}(z) \rfloor_{\sigma(\mathcal{R})} \right) \in \mathcal{R} \quad (1) \quad 20$$

ここで、 Δ はスケール因子 (scaling factor) を示し、 $\lfloor \pi^{-1}(z) \rfloor_{\sigma(\mathcal{R})}$ は $\pi^{-1}(z)$ を $\sigma(\mathcal{R})$ の元素で離散化 (discretization) 又は丸め処理 (rounding) する演算を示す。

多項式 $m(X) \in \mathcal{R}$ に対して、復号化 $\text{Dcd}(m; \Delta)$ はベクトル $z \in \mathbb{C}^{N/2}$ を返すことができ、ベクトルのインデックスのエントリ j は $j \in \{0, 1, \dots, N/2 - 1\}$ に対して $z_j = \lfloor \Delta^{-1} \cdot m(\zeta_M^{5^j}) \rfloor$ を満足し、 ζ_M は M 自乗根 (M-th root of unity) を示す。

30

暗号化装置 10 はキーを生成することができる。与えられたセキュリティパラメータ λ に対して、暗号化装置 10 は M の累乗、整数 h 、整数 P 、正の実数 α 、フレッシュ暗号文モジュラス (fresh ciphertext modulus) q_L 及び最大暗号文モジュラスとなるビッグ暗号文モジュラス (big ciphertext modulus) Q を選択する。

【0072】

フレッシュ暗号文は、演算が実行されていない暗号化されたデータ又は最初に暗号化されたデータを意味する。

【0073】

プロセッサ 100 は、パブリックキー (public key、 pk) 及びシークレットキー (secret key、 sk) を数式 (2) のように設定することができる。

【0074】

【数3】

$$sk := (1, s), pk := (-as + e, a) \in \mathcal{R}_{q_L}^2 \quad (2)$$

ここで、 s 、 a 及び e はそれぞれ $s \leftarrow \text{HWT}(h)$ 、 $a \leftarrow \mathcal{R}_{q_L}$ 及び $e \leftarrow \mathcal{DG}(\alpha^2)$ を意味する。

40

50

プロセッサ 100 は、計算キー (evaluation key) を数式 (3) のように設定することができる。

【0075】

【数4】

$$\text{evk} := (-a's + e' + Ps^2, a') \in \mathcal{R}_{Pq_L}^2 \quad (3)$$

a' 及び e' はそれぞれ $a' \leftarrow \mathcal{R}_{Pq_L}$ 、 $e' \leftarrow \mathcal{DG}(\alpha^2)$ を意味する。

プロセッサ 100 は、符号化 $\text{Enc}_{pk}(m \in \mathcal{R})$ を行うことで、数式 (4) のような変換を行うことができる。

$$c = v \cdot pk + (m + e_0, e_1) \mod q_L \quad (4)$$

ここで、 v 、 e_0 及び e_1 はそれぞれ $v \leftarrow \mathcal{ZO}(0.5)$ 、 $e_0, e_1 \leftarrow \mathcal{DG}(\alpha^2)$ のようにサンプリングされる。

プロセッサ 100 は、復号化 $\text{Dec}_{sk}(c \in \mathcal{R}_{q_L}^2)$ を行って $\bar{m} = \langle c, sk \rangle \mod q_L$ を返す。

【0076】

【数5】

プロセッサ 100 は、2つの暗号文に対する加算 $\text{Add}(c_1, c_2 \in \mathcal{R}_{q_L}^2)$ を行うことで、数式 (5) の結果を返すことができる。

$$c_{\text{add}} = c_1 + c_2 \mod q_L \quad (5)$$

プロセッサ 100 は、2つの暗号文 $c_1 = (b_1, a_1)$ 、 $c_2 = (b_2, a_2)$ に対する乗算 $\text{Mult}_{\text{evk}}(c_1, c_2 \in \mathcal{R}_{q_L}^2)$ を行って数式 (6) の結果を返すことができる。

【0077】

【数6】

$$c_{\text{mult}} = (d_0, d_1) + \lfloor P^{-1} \cdot d_2 \cdot \text{evk} \rfloor \mod q_L \quad (6)$$

ここで、 $(d_0, d_1, d_2) := (b_1 b_2, a_1 b_2 + a_2 b_1, a_1 a_2) \mod q_L$ であってもよい。

また、プロセッサ 100 は、 $\text{RS}_{l \rightarrow l'}(c \in \mathcal{R}_{q_L}^2)$ を行って数式 (7) の結果を返すことができる。

【0078】

【数7】

$$c' = \left\lfloor \frac{q_{l'}}{q_L} c \right\rfloor \mod q_{l'} \quad (7)$$

それぞれの暗号文は、ブートストラップなしに乗算の最大回数を示すレベル l (エル) を有する。レベル l の各暗号文に対するモジュラス q_l はスケーリング因子が p であり、ベースモジュラスが q_0 であるとき、 $p^l q_0$ の値を有することができる。

10

20

30

40

50

【0079】

さらに、プロセッサ100は、ブートストラップにおける同型線形変換に使用される複素共役 (complex conjugation) 演算及び回転 (rotation) 演算を行ってもよい。

【0080】

以下、暗号化装置10が行うブートストラップの動作について説明する。

【0081】

ブートストラップの目的は、乗算がこれ以上実行できないレベル0の暗号文をリフレッシュし、同じメッセージを有するレベルLの暗号文を生成することにある。

【0082】

暗号文は、暗号化されたデータを意味する。暗号文のレベルは、ブートストラップなしに可能な乗算の数を意味する。

【0083】

ブートストラップは、4つの動作を含んでもよい。最初に、ブートストラップはモジュラス増加動作を含み、第2に、同型線形変換動作を含む。第3に、同型モジュラーリダクション動作を含み、第4に、同型線形変換動作を含む。

【0084】

プロセッサ100は、モジュラス増加を行ってもよい。モジュラス増加動作を説明するために、 $R^2_{q_0}$ の代わりに R^2_Q の元素としてレベル0の暗号文が使用されてもよい。

【0085】

レベル0の暗号文は $\langle ct, sk \rangle_{m \bmod q_0}$ の状態であってもよい。ここで、 ct は暗号文を意味し、 sk はシークレットキーを意味する。プロセッサ100が暗号文を復号化 (decrypt) するときには、いかなる $I \in R$ に対して $\langle ct, sk \rangle_{m + q_0 I \bmod Q}$ の形態を有してもよい。

【0086】

ここで、 sk の係数が小さい数を有しているため、 I の係数の絶対値は小さい数を有する。例えば、 I の係数の絶対値は12よりも小さい値を有する。

【0087】

プロセッサ100は、レベル0の暗号文にブートストラップを行うことで $\langle ct', sk \rangle_{m \bmod q_L}$ を満たす ct' を作ることができる。そのため、プロセッサ100は、同型線形変換及びモジュラーリダクション関数の同型計算を行うことができる。

【0088】

以下では、プロセッサ100の同型線形変換動作について説明する。モジュラス増加が実行された後、暗号文 ct は $m + q_0 I$ を暗号化する暗号文として見なし得る。プロセッサ100は、メッセージ多項式の係数にモジュラーリダクションを同型的に行う。

【0089】

演算はスロットのためのものであり、メッセージ多項式の係数に対するものではないため、係数に意味のある演算を行うために、プロセッサ100は、 ct を $m + q_0 I$ の係数をそのスロット (slot) に暗号化する暗号文に変換することができる。

【0090】

プロセッサ100は、同型モジュラーリダクション関数の計算以後に変換された暗号文を以前の暗号文のスロットをそのメッセージの係数で暗号化する他の暗号文 ct' に逆変換する。以下では、このような変換及び逆変換演算をそれぞれCOEFFT o SLOT及びSLOT o COEFFと称する。

【0091】

上述した2つの変換演算は、メッセージの符号化及び復号化の同型計算として見なし、 $\Phi_M(x)$ の根に対するヴァンデルモンド (Vandermonde) の行列の一部の変数による線形変換であってもよい。また、変換演算は、一般的な同型行列乗算又はFFT-類似 (fast Fourier transform-like) 演算によって実行されてもよい。

10

20

30

40

50

【0092】

プロセッサ100は、同型モジュラーリダクション（又は、モジュラーリダクション）動作を行ってもよい。具体的に、プロセッサ100は、同型モジュラーリダクション関数を用いてモジュラーリダクション動作を行ってもよい。

【0093】

C O E F F T o S L O T変換が行われた後、プロセッサ100は、モジュラス q_0 にて各スロット上にモジュラーリダクションを同型的に行うことができる。以下では、このような過程はE V A L M O Dと称する。

【0094】

プロセッサ100はメッセージの範囲を、 m/q_0 を十分に小さくするよう制限することで、近似領域を q_0 の倍数の近所に制限する。このような範囲の制限により、プロセッサ100は、モジュラーリダクションをより効率よく行うことができる。

【0095】

以下、図2Aに示すアルゴリズムについて詳細に説明する。

【0096】

プロセッサ100は、図2Aに示すアルゴリズム1を用いて区間 $[a, b]$ 上の任意の連続関数について最小最大近似多項式を探すことで、ターゲット近似多項式を生成することができる。プロセッサ100は、チェビシェフ交互定理（C h e b y s h e v a l t e r n a t i o n t h e o r e m）を用いて、e q u i o s c i l l a t i o n条件が満たされるターゲット近似多項式を生成することができる。

【0097】

プロセッサ100は、ハール条件（H a a r c o n d i t i o n）を満たす基底関数 $\{g_1, \dots, g_n\}$ を有するターゲット近似多項式を生成する。プロセッサ100は、次数が d であるターゲット近似多項式を生成するため、基底関数 $\{g_1, \dots, g_n\}$ をパワー基底（p o w e r b a s i s） $\{1, x, \dots, x^d\}$ として選択する。ここで、 $n = d + 1$ であってもよい。

【0098】

プロセッサ100は、最小最大近似多項式の極点に収斂する基準点の集合を初期化することができる。プロセッサ100は、基準点の集合に対して最小最大多項式を取得する。基準点の集合は $[a, b]$ で有限な点の集合であるため、 $[a, b]$ の閉部分集合であってもよく、基準点に集合に対してチェビシェフ交互定理が満たされる。

【0099】

$f(x)$ が $[a, b]$ 上の連続関数であれば、基準点の集合上の最小最大近似多項式は、基底 $\{g_1, \dots, g_n\}$ を有する一般化された多項式 $p(x)$ であり、任意の E 値に対して数式（8）の条件を満たす。

【0100】

【数8】

$$p(x_i) - f(x_i) = (-1)^i E \quad i = 1, \dots, d+2 \quad (8)$$

プロセッサ100は、数式（8）を用いて任意の多項式 $p(x)$ を取得する。数式（8）により、 $p(x)$ 及び E の n 係数の $n+1$ の変数及び $n+1$ の方程式を有する線形方程式のシステムが形成され、線形方程式は、ハール条件により特異的でないため、プロセッサ100は、数式（8）の条件を満たす多項式 $p(x)$ を決定することができる。

【0101】

プロセッサ100は、 $z_0 = a$ 、 $z_{n+1} = b$ 、 $i = 1, 2, \dots, n$ 、であるとき、 x_i と x_{i+1} との間で $p(x) - f(x)$ の n 個の零点（z e r o s） z_i を取得し、それぞれの $[z_{i-1}, z_i]$ で $p(x) - f(x)$ の $n+1$ 個の極点である $y_1, \dots, y_{n+1}$ を取得することができる。

【0102】

プロセッサ100は、 $p(x_i) - f(x_i) < 0$ であるとき、 $[z_{i-1}, z_i]$ で $p(x) - f(x)$ の最小点を選択し、 $p(x_i) - f(x_i) > 0$ であるとき、 $[z_{i-1}, z_i]$ で $p(x) - f(x)$ の最大点を選択することができる。

【0103】

これを通じて、プロセッサ100は、新しい極点の集合 $y_1, \dots, y_{n+1}$ を候補点として選択することができる。このような候補点がequioscillation条件を満たせば、プロセッサ100は、チェビシェフ交互定理から最小最大近似多項式を返還することで、ターゲット近似多項式を生成することができる。

【0104】

また、プロセッサ100は、基準点の集合を上述した過程を介して取得した新しい極点の集合である $y_1, \dots, y_{n+1}$ に代替し、上述した多項式の生成過程を繰り返して行う。

【0105】

図2Aに示されたアルゴリズム1は、1つの区間の多重部分区間(multiple sub-intervals)に拡張されてもよい。多重部分区間で拡張して適用される場合、図2Aのステップ3とステップ4が変更される。

【0106】

それぞれの繰り返して、プロセッサ100は、誤差関数 $p - f$ の全ての地域極点を取得し、このような地域極点の絶対誤差値は、現在の基準点における絶対誤差値よりも大きくてもよい。

【0107】

その後、プロセッサ100は、取得した全ての地域極点のうち次の2つの条件が満たされる $n+1$ 個の新しい極点を選択することができる。

【0108】

1. 誤差値は符号が交互に表示される。

【0109】

2. 新しい極点の集合は、全域極点(global extreme point)を含む。

【0110】

上述した2つの条件は、最小最大一般化多項式の収斂を保障する。

【0111】

図2B及び図2Cは、図2Aに示すアルゴリズムを修正した多項式生成方法を示す。プロセッサ100は、全ての地域極点のうち新しい極点を選択する方法を変更してもよい。

【0112】

図2Bに示すアルゴリズムをフローチャートに示せば、図2Cの通りである。プロセッサ100は、近似領域で $d+2$ 個の点を設定する(S210)。 $d+2$ 個の点は上記で説明した1つ以上の基準点を意味する。

【0113】

プロセッサ100は、 $d+2$ 個の基準点に基づいて任意の多項式を取得することができる。例えば、プロセッサ100は、数式(8)を満たす多項式 $p(x)$ とE値を探す(S220)。

【0114】

プロセッサ100は、 $p(x) - f(x)$ の局所極大点と局所極小点のうち、絶対値がEよりも大きいか同じ点を取得する(S230)。ここで、E値は、上述した予め決定された値を意味し、取得された局所極大点と局所極小点は上記で説明した候補点を意味する。

【0115】

プロセッサ100は、取得した点のうち、極大と極小が交互に示されるよう $d+2$ 個の点を選択するが、 $p(x) - f(x)$ の絶対値の和が最大になるよう、 $d+2$ 個の点を選択する(S240)。絶対値の和が最大になる $d+2$ 個の点は上記で説明したターゲット

10

20

30

40

50

点を意味する。絶対値の和が最大になるように $d + 2$ 個の点を選択する過程については、図 4 A、図 4 B を参照して詳しく説明する。

【0116】

プロセッサ 100 は、選択した $d + 2$ 個のターゲット点のうち、絶対値の最大値と最小値の相対誤差が δ よりも小さいかを判別する (S250)。 δ は、上記で説明した閾値を意味する。

【0117】

プロセッサ 100 は、ターゲット点に対応する絶対値の最大値及び最小値の相対誤差が δ よりも小さい場合、任意の多項式 $p(x)$ をターゲット近似多項式で出力する (S260)。そうではない場合、プロセッサ 100 は、S220 ないし S250 の過程を繰り返す。

10

【0118】

以下では、図 2 B 及び図 2 C の動作をより詳しく説明する。

【0119】

プロセッサ 100 が近似しようとする関数は、数式 (9) に表現される有限な数字の整数近所でのみ定義される正規化されたモジュラーリダクション (normalized modular reduction) 関数である。

【0120】

【数 9】

$$\text{normod}(x) = x - \text{round}(x), \quad x \in \bigcup_{i=-(K-1)}^{K-1} [i - \epsilon, i + \epsilon] \quad (9)$$

20

数式 (9) は、ドメイン及び範囲に対してスケーリングされたモジュラーリダクション関数を示す。

【0121】

プロセッサ 100 は、同型計算 (homomorphic evaluation) を効率的に行うために、 $\text{normod}(x)$ を近似するためコサイン関数を倍角の公式 (double-angle formula) を用いて近似することができる。

30

【0122】

倍角の公式を l (エル) 回使用すれば、数式 (10) のコサイン関数が近似されなければならない。

【0123】

【数 10】

$$\cos\left(\frac{2\pi}{2^\ell}\left(x - \frac{1}{4}\right)\right), \quad x \in \bigcup_{i=-(K-1)}^{K-1} [i - \epsilon, i + \epsilon] \quad (10)$$

40

数式 (9)、(10) の関数を含む区分的連続関数を近似するために、プロセッサ 100 は、数式 (11) のように与えられる有限な複数の閉じた区間の和集合 (union) 上に定義される一般的な区分的連続関数を仮定する。

【0124】

【数 11】

50

$$D = \bigcup_{i=1}^t [a_i, b_i] \subset [a, b] \subset \mathbb{R}$$

(1 1)

ここで、全ての $i = 1, \dots, t-1$ に対して、 $a_i < b_i < a_{i+1} < b_{i+1}$ であってもよい。

【 0 1 2 5 】

プロセッサ 1 0 0 は、 d 以下の次数を有する多項式で与えられた区分的連続関数を数式 (1 1) の D 上に近似するため、複数の極点から新しい $d+2$ 個の基準点を選択するための基準を設定することができる。

【 0 1 2 6 】

プロセッサ 1 0 0 は $[a, b]$ 上でハール条件が満たされる $\{g_1, \dots, g_n\}$ を多項式の基底にして、ターゲット近似多項式を生成することができる。プロセッサ 1 0 0 は、各繰り返しで基準点の集合による最小最大近似多項式を取得することができ、次の繰り返しで新しい基準点の集合を選択することができる。

【 0 1 2 7 】

プロセッサ 1 0 0 が基準点の集合を用いて取得した任意の多項式を用いて算出した誤差関数の極点のうち、 $n+1$ 個の点を選択する場合の数は極めて多いこともある。プロセッサ 1 0 0 が行う暗号化過程において、多くの区間が考慮され得るため、候補極点の数が極めて多くてもよい。

【 0 1 2 8 】

プロセッサ 1 0 0 は、繰り返しの数を最大に減らすために、各繰り返しで複数の候補点のうち、 $n+1$ 個のターゲット点を選択してもよい。そのため、プロセッサ 1 0 0 は、各繰り返しで生成する近似多項式を収斂させて最小最大近似多項式を生成することができる。

【 0 1 2 9 】

$n+1$ 個のターゲット点を選択する基準を設定するため、プロセッサ 1 0 0 は数式 (1 2) の関数を定義する。

【 0 1 3 0 】

【数 1 2】

$$\mu_{p,f}(x) = \begin{cases} 1 & p(x) - f(x) \text{ is a local maximum value at } x \text{ on } D \\ -1 & p(x) - f(x) \text{ is a local minimum value at } x \text{ on } D \\ 0 & \text{otherwise,} \end{cases}$$

(1 2)

ここで、 $p(x)$ は各繰り返しで取得した任意の多項式を意味し、 $f(x)$ は近似される区分的連続関数を意味する。以下で便宜のために $\mu_{p,f}$ は μ と称する。

【 0 1 3 1 】

プロセッサ 1 0 0 は、 $p(x) - f(x)$ の全ての極点を取得して集合 B を生成する。 B は有限集合として、 $B = \{x_1, x_2, \dots, x_m\}$ のように示すことができる。プロセッサ 1 0 0 は、 B に属する 1 つの区間で一点を選択してもよい。

【 0 1 3 2 】

B が $x_1 < x_2 < \dots < x_m$ のように昇べきの順で整列されていると仮定すれば、 μ の値は 1 又は -1 であってもよい。極点の数は $m \geq n+1$ を満たす。

【 0 1 3 3 】

プロセッサ 1 0 0 は、関数の集合 S を数式 (1 3) のように定義する。

【 0 1 3 4 】

10

20

30

40

50

【数 1 3】

$$S = \{\sigma : [n+1] \rightarrow [m] \mid \sigma(i) < \sigma(i+1) \text{ for all } i = 1, \dots, n\} \quad (13)$$

ここで、 $n+1=m$ である場合、集合 S は恒等関数 (identity function) のみを含んでもよい。

【0 1 3 5】

プロセッサ 1 0 0 は、 $n+1$ 個の極点を選択するために 3 種類の基準を設定することができる。

【0 1 3 6】

最初の条件として、プロセッサ 1 0 0 は、地域極値 (local extreme value) 条件を設定する。もし、 E が設定された基準点における絶対誤差であれば、数式 (1 4) の条件が設定されてもよい。

【0 1 3 7】

【数 1 4】

$$\min_i \mu(x_{\sigma(i)}) (p(x_{\sigma(i)}) - f(x_{\sigma(i)})) \geq E \quad (14)$$

プロセッサ 1 0 0 は、地域極値の条件を満たすために $p(x) - f(x)$ の地域最大値が負数であるか、 $p(x) - f(x)$ の地域最小値が正数である場合にその極点を捨てることができる。

【0 1 3 8】

第 2 に、プロセッサ 1 0 0 は、交互条件を設定する。言い換えれば、数式 (1 5) の条件が設定されてもよい。言い換えれば、隣接する 2 つの極点のうち 1 つの極点地域最大値であれば、他の 1 つの極点は地域最小値であってもよい。

【0 1 3 9】

【数 1 5】

$$\mu(x_{\sigma(i)}) \cdot \mu(x_{\sigma(i+1)}) = -1 \text{ for } i = 1, \dots, n \quad (15)$$

第 3 に、プロセッサ 1 0 0 は、最大絶対和の条件 (maximum absolute sum condition) を設定する。プロセッサ 1 0 0 は、地域極値の条件及び交互条件が満たされる σ のうち、数式 (1 6) の値を最大化する σ を選択することができる。

【0 1 4 0】

【数 1 6】

$$\sum_{i=1}^{n+1} |p(x_{\sigma(i)}) - f(x_{\sigma(i)})| \quad (16)$$

現在の基準点である $x_1, \dots, x_{n+1}$ における絶対誤差値は小最大近似誤差よりも小さく、繰り返し回数が増加するほど、最小最大近似誤差として収斂されることができる。

【0141】

また、現在の基準点の絶対誤差値は、 $x_1, \dots, x_{n+1}$ で以前の繰り返しにおける近似多項式の絶対誤差値の加重平均であってもよい。

【0142】

プロセッサ100は、最大絶対和の条件を用いて現在の基準点の絶対誤差値が最小最大近似誤差として早く収束するようにする。

【0143】

地域極値の条件及び交互条件は、図2A及び図2Bに示すアルゴリズムの全てに適用され、最大絶対和の条件は、図2Bに示すアルゴリズム2に適用される。プロセッサ100は、最大絶対和の条件を適用させることで、最小最大近似多項式の収束を加速化することができる。

10

【0144】

【数17】

集合 S は、地域極値の条件及び交互条件が満たされる少なくとも1つの元素 σ_0 を常に含み、ある i_0 に対して、 $|p(x_{\sigma_0(i_0)}) - f(x_{\sigma_0(i_0)})| = \|p - f\|_\infty$ を満たす $\sigma_0(i_0)$ を有する。

20

プロセッサ100は、図2Bに示すアルゴリズム2のステップ2、3及び4を下記のように効率よく行うことができる。プロセッサ100は、連続関数 $f(x)$ に対する現在の基準点におけるパワー基底を有する近似多項式の係数を探ることができる。

【0145】

言い換えれば、プロセッサ100は、数式(17)の係数 c_j 値を取得することでターゲット近似多項式を生成し得る。

【0146】

【数18】

$$\sum_{j=0}^d c_j x_i^j - f(x_i) = (-1)^i E \quad (17)$$

30

ここで、 E は、線形方程式の任意の未知数であってもよい。近似多項式の基底の次数が増加するほど係数は減少する。プロセッサ100は、高い次数の基底の係数に対してより高い正確度を設定しなければならない。

【0147】

従って、プロセッサ100は、ターゲット近似多項式の基底をチェビシェフ多項式の基底を用いることで正確度の問題を効率よく解決することができる。チェビシェフ基底を用いた多項式の係数は、ほとんど同じ次数 (order) を有するため、プロセッサ100は、パワー基底の代わりにチェビシェフ基底を用いてターゲット近似多項式を生成することができる。

40

【0148】

チェビシェフ多項式は、上述したハール条件を満たし、プロセッサ100は、 $d+2$ 個の基準点を用いて数式(18)の $d+2$ 個の線形方程式のシステムを解いて c_j 及び E を算出することで、ターゲット近似多項式を取得することができる。

【0149】

【数19】

50

$$\sum_{j=0}^d c_j T_j(x_i) - f(x_i) = (-1)^i E \quad (18)$$

図3は、図1に示された暗号化装置が極点を探索する動作のフローチャートを示す。

【0150】

図3を参照すると、プロセッサ100は、基準点に基づいて任意の多項式を取得し、任意の多項式と基準点を通過する区分的連続関数間の誤差の極点を探索することができる。プロセッサ100は、任意の多項式と基準点を通過する区分的連続関数間の誤差の極点のうち、絶対値が予め決定された値よりも大きい、又は同じ候補点を取得することができる。以下、プロセッサ100が極点を探索して候補点を取得する過程について説明する。

【0151】

プロセッサ100は、任意の多項式と区分的連続関数間の誤差である $p(x) - f(x)$ を小さいステップ (step) でスキャンし、増加と減少が変わる極点を取得することができる。

【0152】

一般に、小さいステップは、極点探索の正確度を増加させるが、スキャン時間が増加してもよい。具体的に、 l (エル) ビットの正確度で極点を取得するためには 2^l に比例する時間が費やされる。

【0153】

しかし、プロセッサ100は、後述する探索動作によって 2^l 時間の代わりに l の線形時間 (linear time) 内に極点を探索することができる。

【0154】

プロセッサ100は、バイナリサーチ (binary search) 方式を用いて増加と減少が変わる極点の探索時間を低減することができる。

【0155】

以下では、任意の多項式と区分的連続関数の誤差を $r(x) = p(x) - f(x)$ のように示し、 sc はスキャンステップを示す。

【0156】

【数20】

プロセッサ100は、

$$\mu(x_0)r(x_0) \geq |E| \text{ 及び}$$

$$(r(x_0) - r(x_0 - sc))(r(x_0 + sc) - r(x_0)) \leq 0 \text{ を満たす } x_0 \text{ を探索し、}$$

数式 (19) の過程を連続して ℓ 回行うことで、 i 番目の極点を取得することができる。

$$x_{i,k} = \arg \max_{x \in \{x_{i,k-1} - sc/2^k, x_{i,k-1}, x_{i,k-1} + sc/2^k\}} |r(x)|, k = 1, 2, \dots, \ell. \quad (19)$$

数式 (19) の過程を介して、プロセッサ100は $O(\log(sc) + l)$ ビットの正確度 (precision) で極点を取得することができる。

【0157】

以下、上述した極点探索を通した候補点の取得過程を詳しく説明する。プロセッサ100は、近似領域のうち最も小さい点 x を取得する (S310)。プロセッサ100は、 x

が極大値である場合、 $r(x)$ が E の絶対値よりも大きい同一であるか、 x が極小値である場合、 $r(x)$ が E の絶対値に -1 を乗算した値よりも小さいかを判断する (S 3 2 0)。

【0 1 5 8】

ステップ S 3 2 0 の条件が満たされる場合、プロセッサ 1 0 0 は、 x_0 を配列 B に追加する (S 3 2 1)。ステップ S 3 2 0 の条件が満たされない場合、プロセッサ 1 0 0 は、 x に $x + sc$ を代入する (S 3 3 0)。

【0 1 5 9】

その後、プロセッサ 1 0 0 は、 x が近似領域に含まれているかを判断する (S 3 4 0)。X が近似領域に含まれている場合、プロセッサ 1 0 0 は、 $r(x) - r(x - sc)$ と $r(x + sc) - r(x)$ の符号が異なるかを判断する (S 3 5 0)。

10

【0 1 6 0】

x が近似領域に含まれていない場合、プロセッサ 1 0 0 は、 x に該当区間の最も大きい値を代入する (S 3 4 1)。ここで、プロセッサ 1 0 0 は、 x が極大値である場合、 $r(x)$ が E の絶対値よりも大きい同一であるか、又は、 x が極小値である場合、 $r(x)$ が E の絶対値に -1 を乗算した値よりも小さいかを判断する (S 3 4 2)。

【0 1 6 1】

プロセッサ 1 0 0 は、ステップ S 3 4 2 の条件が満たされる場合、 x_0 を配列 B に追加し (S 3 4 3)、ステップ S 3 4 2 の条件が満たされない場合、 x が近似領域の最大値であるかを判断する (S 3 4 4)。ここで、 x が近似領域の最大値である場合、プロセッサ 1 0 0 は動作を終了し、 x が近似領域の最大値でない場合、 x を次の区間の最も小さい値に代入する (S 3 4 5)。

20

【0 1 6 2】

プロセッサ 1 0 0 は、 $r(x) - r(x - sc)$ と $r(x + sc) - r(x)$ の符号が同一である場合、ステップ S 3 3 0 の動作を再び実行し、 $r(x) - r(x - sc)$ と $r(x + sc) - r(x)$ の符号が異なる場合、 l (エル) に 0 を代入し、 t に $sc / 2$ を代入する (S 3 6 0)。

【0 1 6 3】

プロセッサ 1 0 0 は、 $r(x) - r(x - sc)$ の値が 0 よりも大きいかを判断する (S 3 7 0)。プロセッサ 1 0 0 は、ステップ S 3 7 0 の条件が満たされる場合、 $x - t$ 、 x 、 $x + t$ のうち、 $r(x)$ 値が大きいものを選択して x に代入する (S 3 7 1)。その後、プロセッサ 1 0 0 は、 l に $l + 1$ を代入し、 t に $t / 2$ を代入する (S 3 7 2)。

30

【0 1 6 4】

プロセッサ 1 0 0 は、 l が正確度値であるかを比較する (S 3 7 3, S 3 7 4)。プロセッサ 1 0 0 は、 l が正確度値でない場合、ステップ S 3 7 1 の過程を再び行ってもよい。プロセッサ 1 0 0 は、 l が正確度値である場合、 $x - t$ 、 x 、 $x + t$ のうち、 $r(x)$ 値が小さいものを選択して x に代入する (S 3 7 5)。その後、プロセッサ 1 0 0 は、 l に $l + 1$ を代入し、 t に $t / 2$ を代入する (S 3 7 6)。

【0 1 6 5】

プロセッサ 1 0 0 は、S 3 7 3、S 3 7 4 の条件が満たされる場合、ステップ S 3 2 0 の動作を再び行ってもよい。最終的に、プロセッサ 1 0 0 は、配列 B に属する極点を候補点として取得することができる。

40

【0 1 6 6】

sc の値が十分に小さければ、 $|r(x)|$ は $a > 0$ であり、 x^* の近くの b に対して $a(x - x^*)^2 + b$ と類似して動作することができる。このような、動作を通じてプロセッサ 1 0 0 は、 x^* の近所で $|x_1 - x^*| < |x_2 - x^*|$ である場合、 $|r(x_1)| > |r(x_2)|$ を保障することができ、その逆も保障できる。

【0 1 6 7】

上述した極点探索を通した候補点の取得動作により、プロセッサ 1 0 0 は、 l のビット正確度を有する極点を 2^l 時間の代わりに、 l の線形時間 (linear time) 内

50

に探索して候補点を取得することができる。

【0168】

図4Aは、図1に示された暗号化装置がターゲット点を選択するアルゴリズムの例を示し、図4Bは、図1に示された暗号化装置がターゲット点を選択する動作のフローチャートを示す。

【0169】

図4A及び図4Bを参照すると、プロセッサ100は、図3に示す探索動作によって探した候補点のうち、ターゲット近似多項式の次数に基づいた数のターゲット点を選択する。

【0170】

プロセッサ100は、候補点のうち極大と極小が交互に示されるようにターゲット点を選択し、誤差の絶対値の和が最大になるようターゲット点を選択する。ターゲット点は、次の繰り返し (iteration) 動作における新しい基準点になり得る。

【0171】

以下では、ターゲット点を取得する過程について詳細に説明する。プロセッサ100は、ターゲット点 (又は、新しい基準点) を探すために上記で説明した地域極値の条件、交互条件、及び最大絶対和の条件が満たされる点をサイーブ (naive) な方式で選択することができる。

【0172】

ナイーブな選択方式は、交互条件が満たされる全ての $n+1$ 個の点に対して絶対和を算出し、最大絶対和を有する $n+1$ 個の点を選択してもよい。ナイーブな選択方式である場合、地域極点の数が m である場合、 (m_{n+1}) 個の点を全て調査しなければならない。

【0173】

プロセッサ100は、図4A及び図4Bに示す動作により、ナイーブな方式に比べてターゲット点を選択する時間を低減することができる。以下では、ターゲット点を効率よく選択する動作について説明する。

【0174】

プロセッサ100は、繰り返しの動作ごとに候補点のうち、一部の元素を除去することで、最終的に $n+1$ 個のターゲット点を取得することができる。 $m > n+1$ である場合、ターゲット点に含まれていない点が少なくとも1つ存在し得る状態であってもよい。

【0175】

図4Aに示すアルゴリズムを介して、プロセッサ100は、 $O(m \log m)$ 時間内にターゲット点を選択する。言い換えれば、プロセッサ100は、準線型 (quasi-linear) 時間内にターゲット点を選択することができる。

【0176】

順に整列した集合 B から元素が除去されるごとに、残っている元素は整列され、インデックスである昇順で再びラベル (relabelled) されてもよい。

【0177】

プロセッサ100は、アルゴリズムの4aで一部の極点を除去するために値を比較するとき、比較される値が同一であるか最も小さい元素が1つ以上の場合、このような元素をランダムに除去することができる。

【0178】

図4Bに示すフローチャートは、図4Aに示すアルゴリズムの動作の順序を示す。図4A及び図4Bの動作により、プロセッサ100は、ターゲット点を元素として有する配列 B を取得することができる。

【0179】

プロセッサ100は、 i に1を代入する (S410)。プロセッサ100は、 x_i と x_{i+1} の両方が極大であるか極小であるかを判断する (S420)。

【0180】

ステップS420の条件を満たす場合、プロセッサ100は、 x_i と x_{i+1} のうち $|r(x)|$ 値が小さい値は配列から除去し、配列の元素を再配列する (S421)。 $|r(x)|$

10

20

30

40

50

x) | 値は、上述した任意の多項式と区分的連続関数の誤差値を意味する。ステップ S 4 2 0 の条件が満たされない場合、プロセッサ 1 0 0 は、 i に $i + 1$ を代入する (S 4 2 2)。

【0181】

再配列した後、プロセッサ 1 0 0 は、 x_i が配列 B で最も大きい点であるかを判断する (S 4 3 0)。 x_i が最も大きい点ではない場合、プロセッサ 1 0 0 は 4 2 0 の動作を再び行う。

【0182】

ステップ S 4 1 0 ~ ステップ S 4 3 0 の動作は、図 4 A に示すアルゴリズム 3 のステップ 1 ~ 7 の動作に対応する。

【0183】

x_i が最も大きい点である場合、プロセッサ 1 0 0 は、B の元素の個数が $d + 2$ 個であるかを判断する (S 4 4 0)。 B の元素の個数が $d + 2$ 個である場合、プロセッサ 1 0 0 は、ターゲット点の選択動作を終了する。

【0184】

B の元素の個数が $d + 2$ 個でない場合、プロセッサ 1 0 0 は、隣接する 2 点の $|r(x)|$ 値を加えた値の全てを配列 T に入れて整列する (S 4 5 0)。言い換えれば、 $|r(x_1)| + |r(x_2)|$, $|r(x_2)| + |r(x_3)|$, $|r(x_3)| + |r(x_4)|$, $\dots$ を T に入れて整列することができる。

【0185】

ステップ S 4 4 0 ~ ステップ S 4 5 0 の動作は、図 4 A に示すルゴリズムのステップ 9 ~ ステップ S 1 0 の動作に対応する。

【0186】

プロセッサ 1 0 0 は、B の元素の個数が $d + 3$ 個であるかを判断する (S 4 6 0)。 B の元素の個数が $d + 3$ 個である場合、プロセッサ 1 0 0 は x_1 、 x_{d+3} のうち $|r(x)|$ 値が小さい値を配列から捨て、配列の再配列を行って動作を終了する (S 4 6 1)。

【0187】

プロセッサ 1 0 0 は、B の元素の個数が $d + 4$ 個であるかを判断する (S 4 7 0)。 B の元素の個数が $d + 4$ 個である場合、プロセッサ 1 0 0 は T に $|r_{x_1}| + |r_{x_{d+4}}|$ を追加して再整列する (S 4 7 1)。その後、プロセッサ 1 0 0 は、T で最も小さい値に該当する 2 点を B から除去して再配列した後、動作を終了する (S 4 7 2)。

【0188】

プロセッサ 1 0 0 は、B の元素の個数が $d + 4$ でない場合、T で最も小さい値に該当する 2 点のうち、両端の 2 個の点のうち 1 つが含まれているかを判断する (S 4 8 0)。ステップ S 4 8 0 の条件が満たされる場合、プロセッサ 1 0 0 は、両端 2 個の点のうちの 1 つを B から捨てて再配列する (S 4 8 1)。ステップ S 4 8 0 の条件を満たされない場合、プロセッサ 1 0 0 は、T で最も小さい値に該当する 2 点を全て B から捨てて再配列する (S 4 8 2)。

【0189】

その後、プロセッサ 1 0 0 は、T から捨てられた元素が含まれている値は捨て、新しく隣接になった 2 点の $|r(x)|$ 値を加えた値を追加して再整列した後、ステップ S 4 6 0 の動作を再び行う (S 4 9 0)。ステップ S 4 6 0 ~ ステップ S 4 9 0 の動作については、図 4 A に示すステップ S 1 1 ~ ステップ S 2 3 の動作に対応する。

【0190】

図 4 A に示すアルゴリズムの最後の部分で極点 X 2 が除去される場合について説明すると、

$T = \{|r(x_1)| + |r(x_2)|, |r(x_2)| + |r(x_3)|, |r(x_3)| + |r(x_4)|, \dots\}$

は、

$T = \{|r(x_1)| + |r(x_3)|, |r(x_3)| + |r(x_4)|, \dots\}$

10

20

30

40

50

のように変更される。

【0191】

図4A及び図4Bに示すターゲット点を選択する動作により、プロセッサ100は準線型時間内に候補点からターゲット点を選択することができる。

【0192】

プロセッサ100は、選択したターゲット点に基づいてモジュラーリダクション関数を最適に近似するターゲット近似多項式を生成することができる。言い換えれば、プロセッサ100は、選択したターゲット点を経由するd次多項式をターゲット近似多項式により生成することができる。

【0193】

図5は、図1に示された暗号化装置の全体動作の順序を示す。

【0194】

プロセッサ100は、同型暗号(homomorphic encryption)を用いてデータに対する暗号化を行う。プロセッサ100は、データを暗号化することで暗号文を生成する(S510)。

【0195】

プロセッサ100は、生成した暗号文に対応するモジュラスに対して1つ以上のターゲット点の選択に基づいたモジュラーリダクションを行うことで、暗号文にブートストラップを行う(S530)。

【0196】

プロセッサ100は、モジュラーリダクションに対応する関数を近似することで、ブートストラップを行ってもよい。プロセッサ100は、モジュラーリダクションに対応する関数を近似するターゲット近似多項式を生成する。

【0197】

プロセッサ100は、ターゲット近似多項式の次数に基づいて1つ以上の基準点を決定する。プロセッサ100は、決定された1つ以上の基準点に基づいて任意の多項式を取得する。

【0198】

具体的に、プロセッサ100は、1つ以上の基準点を通過する区分的連続関数を取得し、1つ以上の基準点で区分的連続関数との誤差の絶対値が予め決定された値になる多項式を生成することで、任意の多項式を取得することができる。

【0199】

プロセッサ100は、1つ以上の基準点に含まれた第1基準点における誤差と、第1基準点と隣接する第2基準点の誤差の符号が異なり、絶対値が予め決定された値になる多項式を生成することで、任意の多項式を取得することができる。

【0200】

プロセッサ100は、取得した任意の多項式から選択された1つ以上の極点に基づいてターゲット近似多項式を生成する。プロセッサ100は、任意の多項式と1つ以上の基準点を通過する区分的連続関数間の誤差の極点のうち、絶対値が予め決定された値よりも大きい場合と同じ候補点を取得することができる。

【0201】

プロセッサ100は、取得した候補点のうち、ターゲット近似多項式の次数に基づいた数のターゲット点を選択する。具体的に、プロセッサ100は、候補点のうち、極大と極小が交互に示されるようにターゲット点を選択してもよい。プロセッサ100は、区分的連続関数の任意の多項式の誤差の絶対値の和が最大になるようにターゲット点を選択してもよい。

【0202】

プロセッサ100は、選択したターゲット点に基づいてターゲット近似多項式を生成する。プロセッサ100は、1つ以上の極点の絶対値の最大値と最小値の相対誤差が、閾値よりも小さい場合の多項式をターゲット近似多項式により生成することができる。

10

20

30

40

50

【0203】

ここで、ターゲット近似多項式の基底 (basis) は、チェビシェフ多項式の基底が使用され得る。

【0204】

実施形態に係る方法は、様々なコンピュータ手段を介して実施されるプログラム命令の形態で具現され、コンピュータ読み取り可能な記録媒体に記録される。記録媒体は、プログラム命令、データファイル、データ構造などを単独又は組み合わせて含む。記録媒体及びプログラム命令は、本発明の目的のために特別に設計して構成されたものでもよく、コンピュータソフトウェア分野の技術を有する当業者にとって公知のものであり使用可能なものであってもよい。コンピュータ読み取り可能な記録媒体の例として、ハードディスク、フロッピー（登録商標）ディスク及び磁気テープのような磁気媒体、CD-ROM、DVDのような光記録媒体、プロブティカルディスクのような磁気-光媒体、及びROM、RAM、フラッシュメモリなどのようなプログラム命令を保存して実行するように特別に構成されたハードウェア装置を含む。プログラム命令の例としては、コンパイラによって生成されるような機械語コードだけでなく、インタプリタなどを用いてコンピュータによって実行される高級言語コードを含む。ハードウェア装置は、本発明に示す動作を実行するために1つ以上のソフトウェアモジュールとして作動するように構成してもよく、その逆も同様である。

10

【0205】

ソフトウェアは、コンピュータプログラム、コード、命令、又はそのうちの一つ以上の組合せを含み、希望の通りに動作するように処理装置を構成したり、独立的又は結合的に処理装置を命令することができる。ソフトウェア及び／又はデータは、処理装置によって解釈されたり処理装置に命令又はデータを提供するために、いずれかの種類の機械、構成要素、物理的装置、仮想装置、コンピュータ格納媒体又は装置、又は送信される信号波に永久的又は一時的に具体化することができる。ソフトウェアはネットワークに連結されたコンピュータシステム上に分散され、分散した方法で格納されたり実行され得る。ソフトウェア及びデータは一つ以上のコンピュータで読出し可能な記録媒体に格納され得る。

20

【0206】

上述したように実施形態をたとえ限定された図面によって説明したが、当技術分野で通常の知識を有する者であれば、上記の説明に基づいて様々な技術的な修正及び変形を適用することができる。例えば、説明された技術が説明された方法と異なる順で実行されるし、及び／又は説明されたシステム、構造、装置、回路などの構成要素が説明された方法と異なる形態で結合又は組み合わせられてもよいし、他の構成要素又は均等物によって置き換え又は置換されたとしても適切な結果を達成することができる。

30

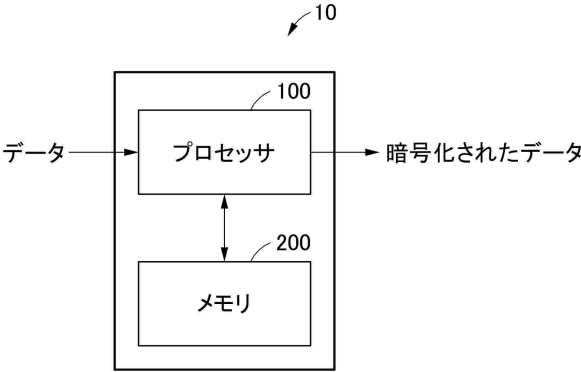
【0207】

したがって、他の具現、他の実施形態、及び特許請求の範囲と均等なものも後述する請求範囲の範囲に属する。

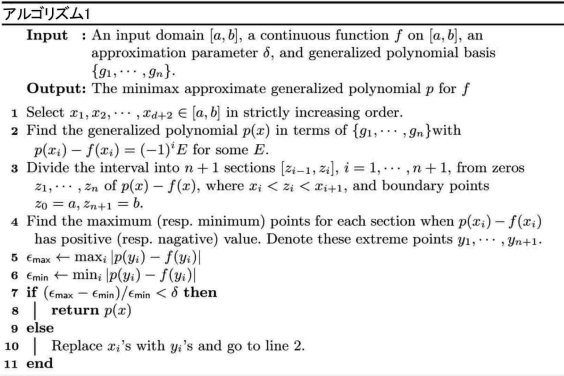
40

50

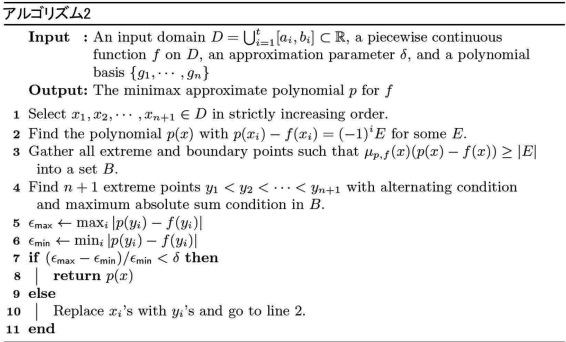
【図面】
【図 1】



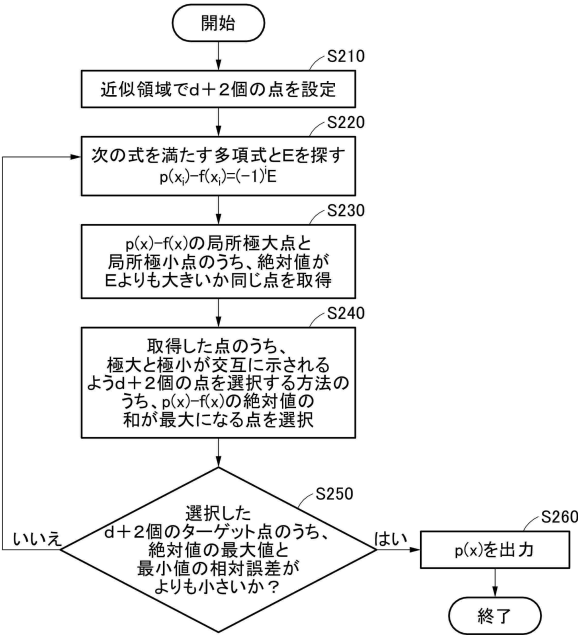
【図 2 A】



【図 2 B】



【図 2 C】



10

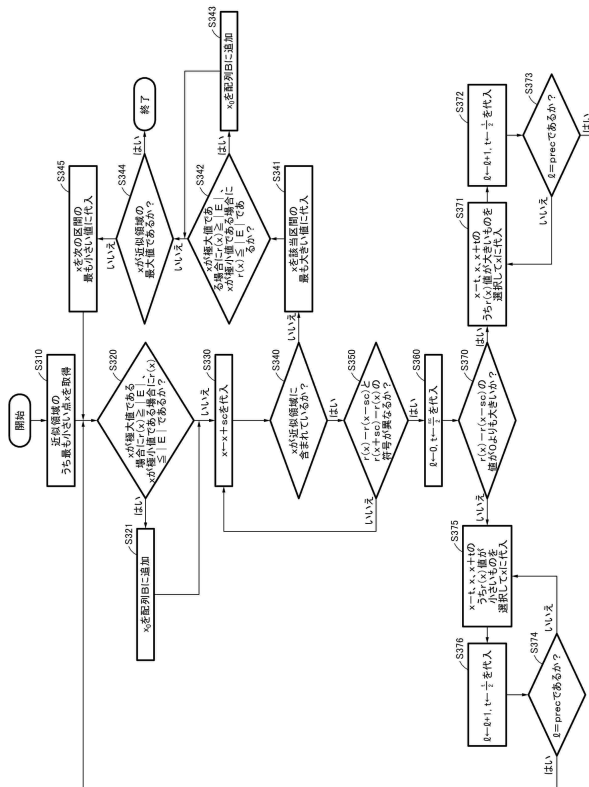
20

30

40

50

【図 3】



【図 4 A】

アルゴリズム3

Input : An increasing ordered set of extreme points $B = \{t_1, t_2, \dots, t_m\}$ with $m \geq n + 1$, and number of basis n .

Output: $n + 1$ points in B satisfying alternating condition and maximum absolute sum condition.

```

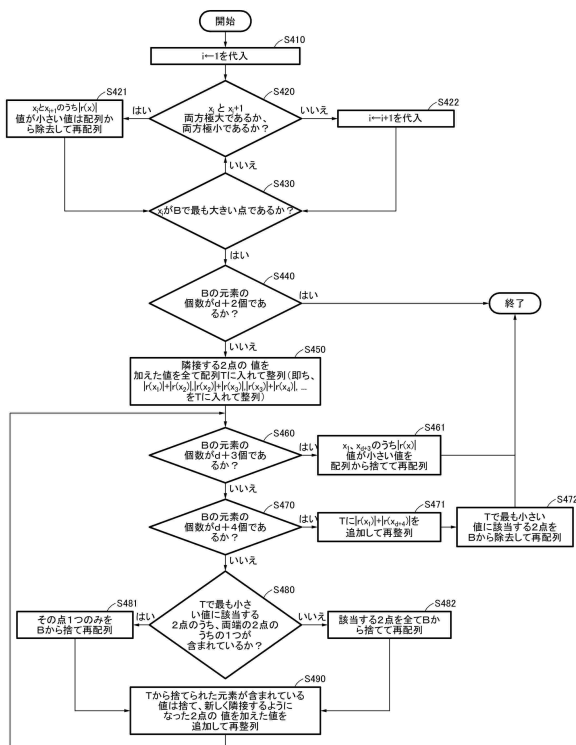
1   $i \leftarrow 1$ 
2  while  $t_i$  is not the last element of  $B$  do
3    if  $\mu(t_i)\mu(t_{i+1}) = 1$  then
4      Remove from  $B$  one of two points  $t_i, t_{i+1}$  having less value among
        $\{|r(t_i)|, |r(t_{i+1})|\}$ .
5    else
6       $i \leftarrow i + 1$ 
7    end
8  end
9  if  $|B| > n + 2$  then
10   Calculate all  $|r(t_i)| + |r(t_{i+1})|$  for  $i = 1, \dots, |B| - 1$  and sort and store this
       values into the array  $T$ .
11  while  $|B| > n + 1$  do
12   if  $|B| = n + 2$  then
13     Remove from  $B$  one of two points  $t_1, t_{|B|}$  having less value among
        $\{|r(t_1)|, |r(t_{|B|})|\}$ .
14   else if  $|B| = n + 3$  then
15     Insert  $|r(t_1)| + |r(t_{|B|})|$  into  $T$  and sort  $T$ . Remove from  $B$  the two
       element having the smallest value in  $T$ .
16   else
17     if  $t_1$  or  $t_{|B|}$  is included in the smallest element in  $T$  then
18       Remove from  $B$  only  $t_1$  or  $t_{|B|}$ .
19     else
20       Remove from  $B$  the two elements having the smallest element in  $T$ .
21   end
22   Remove from  $T$  all elements related to the removed extreme points,
       and insert into  $T$  the sum of absolute error values of the two newly
       adjacent extreme points.
23 end
24 end

```

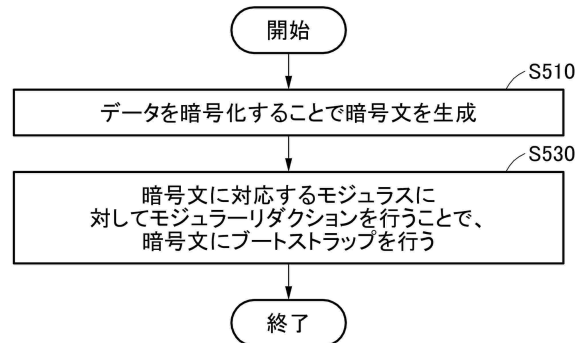
10

20

【図 4 B】



【図 5】



30

40

50

フロントページの続き

クーク クァナツクーロ 1

(73)特許権者 521157546

朝鮮大學校産學協力團

大韓民国光州廣域市東区畢門大路 3 0 9 朝鮮大學校 本館

(74)代理人 100107766

弁理士 伊東 忠重

(74)代理人 100070150

弁理士 伊東 忠彦

(74)代理人 100135079

弁理士 宮崎 修

(72)発明者 盧 宗善

大韓民国ソウル特別市冠岳区冠岳路 1

(72)発明者 李 俊雨

大韓民国ソウル特別市冠岳区冠岳路 1

(72)発明者 金 榮植

大韓民国光州廣域市東区畢門大路 3 0 9 朝鮮大學校 本館

(72)発明者 李 勇雨

大韓民国ソウル特別市冠岳区冠岳路 1

(72)発明者 李 恩相

大韓民国ソウル特別市冠岳区冠岳路 1

審査官 平井 誠

(56)参考文献

Yongwoo Lee et al., Near-optimal Polynomial for Modulus Reduction Using L2-norm for Approximate Homomorphic Encryption, ePrint Archive, 2020年04月28日, pages 1-9, <URL:https://eprint.iacr.org/2020/488>

(58)調査した分野 (Int.Cl., D B 名)

G 0 9 C 1 / 0 0 - 5 / 0 0

H 0 4 L 9 / 0 0 - 4 0