(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2020-46558
(P2020-46558A)
(43)公開日　令和2年3月26日(2020.3.26)

(51)Int.Cl.		F I			テーマコード（参考）
G09C 1/00	(2006.01)	G09C	1/00	650Z	5J104
G06F 21/60	(2013.01)	G06F	21/60	320	

審査請求 未請求　請求項の数 14　ＯＬ　（全 27 頁）

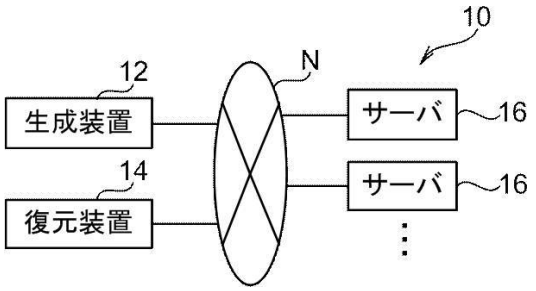
(21)出願番号　特願2018-175393 (P2018-175393)
(22)出願日　平成30年9月19日 (2018.9.19)

(71)出願人　000125370
学校法人東京理科大学
東京都新宿区神楽坂一丁目３番地
(74)代理人　100079049
弁理士　中島　淳
(74)代理人　100084995
弁理士　加藤　和詳
(74)代理人　100099025
弁理士　福田　浩志
(72)発明者　岩村　恵市
東京都新宿区神楽坂一丁目３番地　学校法人東京理科大学内

Ｆターム(参考)　5J104 AA16　EA02　EA13　PA07　PA14

(54)【発明の名称】生成装置、復元装置、送信装置、受信装置、生成プログラム、復元プログラム、送信プログラム、及び受信プログラム

(57)【要約】　（修正有）
【課題】秘密情報のオーナが知らない間に、秘密情報が漏洩することを抑制することができる秘密分散システムを提供する。
【解決手段】ｎを２以上の整数、ｋを２以上ｎ以下の整数とし、秘密情報をｎ個の分散値に分散し、ｋ個の分散値によって秘密情報を復元でき、ｋ個未満では秘密情報を復元できないシステムにおいて、生成装置１２は、秘密情報を識別する識別情報を、１つの秘密鍵を用いて変換することによって、少なくともｋ－１個以下の分散値の各々に対応する値を生成する。
【選択図】図１



【特許請求の範囲】
【請求項１】
ｎを２以上の整数、ｋを２以上ｎ以下の整数とし、秘密情報をｎ個の分散値に分散し、ｋ個の前記分散値によって前記秘密情報を復元でき、ｋ個未満では前記秘密情報を復元できないシステムにおいて、
前記秘密情報を識別する識別情報を、１つの秘密鍵を用いて変換することによって、少なくともｋ－１個以下の前記分散値の各々に対応する値を生成する生成部
を備えた生成装置。
【請求項２】
ｎを２以上の整数、ｋを２以上ｎ以下の整数とし、秘密情報をｎ個の分散値に分散し、ｋ個の前記分散値によって前記秘密情報を復元でき、ｋ個未満では前記秘密情報を復元できないシステムにおいて、
前記秘密情報を識別する識別情報を、１つの秘密鍵を用いて変換することによって、ｋ－１個以下の前記分散値を生成する生成部
を備えた生成装置。
【請求項３】
前記生成部は、生成した値及び前記秘密情報を用いて前記ｋ－１個以下の前記分散値以外の分散値を求めるための係数を算出する
請求項２に記載の生成装置。
【請求項４】
請求項１又は請求項２に記載の生成装置により生成された値を受信する受信部と、
前記受信部により受信された値を用いて前記秘密情報を復元する復元部と、
を備えた復元装置。
【請求項５】
請求項３に記載の生成装置により生成された係数を用いて同一の秘密情報から生成された異なる分散値の組み合わせによって２回以上前記秘密情報を復元する復元部
を備えた復元装置。
【請求項６】
秘密情報に対して、第１の乱数及び第２の乱数を作用させて第１の送信データを生成し、
前記秘密情報に対して、前記第１の乱数に代えた第３の乱数及び前記第２の乱数に代えた第４の乱数を同様に作用させて第２の送信データを生成する生成部と、
前記第１の送信データ及び前記第２の送信データを同一の受信装置に送信する送信部と、
を備えた送信装置。
【請求項７】
請求項６に記載の送信装置により送信された第１の送信データ及び第２の送信データを受信する受信部と、
前記第１の送信データから前記第２の乱数を排除した結果と、前記第２の送信データから前記第４の乱数を排除した結果との差分が、前記第１の乱数と前記第３の乱数との差分に等しいか否かを検証する検証部と、
を備えた受信装置。
【請求項８】
ｎを２以上の整数、ｋを２以上ｎ以下の整数とし、秘密情報をｎ個の分散値に分散し、ｋ個の前記分散値によって前記秘密情報を復元でき、ｋ個未満では前記秘密情報を復元できないシステムにおいて、
前記秘密情報を識別する識別情報を、１つの秘密鍵を用いて変換することによって、少なくともｋ－１個以下の前記分散値の各々に対応する値を生成する
処理をコンピュータに実行させるための生成プログラム。
【請求項９】

ｎを２以上の整数、ｋを２以上ｎ以下の整数とし、秘密情報をｎ個の分散値に分散し、ｋ個の前記分散値によって前記秘密情報を復元でき、ｋ個未満では前記秘密情報を復元できないシステムにおいて、
前記秘密情報を識別する識別情報を、１つの秘密鍵を用いて変換することによって、ｋ－１個以下の前記分散値を生成する
処理をコンピュータに実行させるための生成プログラム。
【請求項１０】
生成した値及び前記秘密情報を用いて前記ｋ－１個以下の前記分散値以外の分散値を求めるための係数を算出する
処理を更に前記コンピュータに実行させるための請求項９に記載の生成プログラム。
【請求項１１】
請求項８又は請求項９に記載の生成プログラムが実行されることにより生成された値を受信し、
受信した値を用いて前記秘密情報を復元する
処理をコンピュータに実行させるための復元プログラム。
【請求項１２】
請求項１０に記載の生成プログラムが実行されることにより生成された係数を用いて同一の秘密情報から生成された異なる分散値の組み合わせによって２回以上前記秘密情報を復元する
処理をコンピュータに実行させるための復元プログラム。
【請求項１３】
秘密情報に対して、第１の乱数及び第２の乱数を作用させて第１の送信データを生成し、
前記秘密情報に対して、前記第１の乱数に代えた第３の乱数及び前記第２の乱数に代えた第４の乱数を同様に作用させて第２の送信データを生成し、
前記第１の送信データ及び前記第２の送信データを同一の受信装置に送信する
処理をコンピュータに実行させるための送信プログラム。
【請求項１４】
請求項１３に記載の送信プログラムが実行されることにより送信された第１の送信データ及び第２の送信データを受信し、
前記第１の送信データから前記第２の乱数を排除した結果と、前記第２の送信データから前記第４の乱数を排除した結果との差分が、前記第１の乱数と前記第３の乱数との差分に等しいか否かを検証する
処理をコンピュータに実行させるための受信プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本開示は、生成装置、復元装置、送信装置、受信装置、生成プログラム、復元プログラム、送信プログラム、及び受信プログラムに関する。
【背景技術】
【０００２】
近年、新たなネットワーク技術としてクラウドコンピューティングが注目されている。クラウドコンピューティングとは、ユーザの持つデータをクラウドと呼ばれるネットワーク上の複数のサーバにより構成される仮想の大容量ストレージに分散して保管する技術である。クラウドコンピューティングでは、分散して保管されたデータに対し、ユーザがどこからでもネットワーク経由で必要に応じてアクセスすることを可能にする。さらに、データを暗号化して秘匿計算を実現することで単にデータを保管するだけでなく、クラウド上に分散して保管されたデータを用いて、個々のデータを秘匿しながら任意の処理を行うことが求められている。
【０００３】

このような秘匿計算を実現するために秘密分散法の利用が注目されている。秘密分散法とは１個の秘密情報をｎ個に分散し、ｎ個に分散した分散値のうち、ｋ個（ｋ≦ｎ）の分散値を集めることで元の秘密情報が復元できるという技術である。また、秘密分散法では、ｎ個に分散した分散値のうち、ｋ個未満の分散値からは秘密情報に関する情報を得ることができない。この秘密分散法として、Ｓｈａｍｉｒによる（ｋ、ｎ）閾値秘密分散法（以下、「Ｓｈａｍｉｒ法」ともいう）が知られている。

【０００４】

また、分散値全体のデータサイズの小型化を実現するためのランプ型秘密分散法も知られている。また、非特許文献１では、非対称秘密分散法と呼ばれる秘密分散法が提案されている。

【０００５】

また、特許文献１には、秘密分散のデータ復元処理の計算量を抑制する秘密分散システムが開示されている。

【先行技術文献】

【特許文献】

【０００６】

【特許文献１】特開２０１３－２４３４４１号公報

【非特許文献】

【０００７】

【非特許文献１】高橋慧、岩村恵市、“クラウドコンピューティングに適した計算量的安全性を持つ秘密分散法”、ＣＳＳ２０１２（２０１２）

【発明の概要】

【発明が解決しようとする課題】

【０００８】

しかしながら、Ｓｈａｍｉｒ法をはじめとする秘密分散法では、正当な復元者ではない攻撃者が、分散値が保管されているサーバを攻撃し、ｋ個以上の分散値を得ることができれば、そのｋ個以上の分散値から秘密情報を復元できてしまう。また、非対称秘密分散法でも、攻撃者が鍵サーバ及びデータサーバにアクセスできる状態にあり、鍵を用いて生成した分散値を含めｋ個の分散値を得ることができれば、秘密情報を復元することができてしまう。

【０００９】

すなわち、前述した各種の秘密分散法では、秘密情報のオーナが知らない間に、秘密情報が漏洩してしまう場合があった。

【００１０】

本開示は、以上の事情を鑑みて成されたものであり、秘密情報のオーナが知らない間に、秘密情報が漏洩することを抑制することができる生成装置、復元装置、送信装置、受信装置、生成プログラム、復元プログラム、送信プログラム、及び受信プログラムを提供することを目的とする。

【課題を解決するための手段】

【００１１】

開示の技術は、一つの態様として、ｎを２以上の整数、ｋを２以上ｎ以下の整数とし、秘密情報をｎ個の分散値に分散し、ｋ個の前記分散値によって前記秘密情報を復元でき、ｋ個未満では前記秘密情報を復元できないシステムにおいて、生成装置が、前記秘密情報を識別する識別情報を、１つの秘密鍵を用いて変換することによって、少なくともｋ－１個以下の前記分散値の各々に対応する値を生成する生成部を備える。

【発明の効果】

【００１２】

本開示によれば、秘密情報のオーナが知らない間に、秘密情報が漏洩することを抑制することができる。

【図面の簡単な説明】

【００１３】
【図１】第１～第４実施形態に係る秘密分散システムの構成の一例を示すブロック図である。
【図２】第１～第４実施形態に係る生成装置のハードウェア構成の一例を示すブロック図である。
【図３】第１～第４実施形態に係る復元装置のハードウェア構成の一例を示すブロック図である。
【図４】第１実施形態に係る分散処理の一例を示すフローチャートである。
【図５】第１実施形態に係る復元処理の一例を示すフローチャートである。
【図６】第２実施形態に係る分散処理の一例を示すフローチャートである。
【図７】第２実施形態に係る復元処理の一例を示すフローチャートである。
【図８】第３実施形態に係る分散処理の一例を示すフローチャートである。
【図９】第３実施形態に係る復元処理の一例を示すフローチャートである。
【図１０】第４実施形態に係る分散処理の一例を示すフローチャートである。
【図１１】第４実施形態に係る復元処理の一例を示すフローチャートである。
【図１２】第５実施形態に係る通信システムの構成の一例を示すブロック図である。
【図１３】第５実施形態に係る送信装置のハードウェア構成の一例を示すブロック図である。
【図１４】第５実施形態に係る受信装置のハードウェア構成の一例を示すブロック図である。
【図１５】第５実施形態に係る送信処理の一例を示すフローチャートである。
【図１６】第５実施形態に係る受信処理の一例を示すフローチャートである。
【発明を実施するための形態】
【００１４】
以下、図面を参照して、本開示の技術を実施するための形態例を詳細に説明する。
【００１５】
まず、実施形態の詳細を説明する前に、Ｓｈａｍｉｒ法及び非対称秘密分散法の詳細と問題点とを説明する。
【００１６】
＜Ｓｈａｍｉｒ法＞
Ｓｈａｍｉｒ法による秘密分散システムでは、ｎ台のサーバに秘密情報が分散される。秘密情報のオーナは、秘密分散時にのみ存在するディーラに秘密情報の分散を依頼し、ディーラは、秘密分散演算を行うことによりｎ個の分散値を計算し、分散値をｎ台のサーバに分散して保管する。もしくは、秘密情報のオーナは、秘密分散時に自らディーラとなって秘密情報を分散する。一方、復元者は、ｎ台のサーバに分散されたｎ個の分散値のうち、ｋ個の分散値を集めることによって、秘密情報を復元することができる。Ｓｈａｍｉｒ法での秘密情報の分散及び復元は、以下に示すように行われる。
【００１７】
＜分散＞
１．ディーラは、ｓ＜ｐで、かつｎ＜ｐである任意の素数ｐを選ぶ。なお、ｓは秘密情報を表す。
２．ディーラは、Ｚ／ｐＺから、異なるｎ個のx_i（ｉ＝１、２、…、ｎ）を選び、選んだx_iをｎ台のサーバそれぞれのサーバＩＤとする。
３．ディーラは、Ｚ／ｐＺから、ｋ－１個の乱数a_l（ｌ＝１、２、…、ｋ－１）を選び、以下の（１）式（以下、「分散式」ともいう）を生成する。
【００１８】
【数１】

$$W_i = s + a_1 x_i + a_2 x_i^2 + \cdots + a_{k-1} x_i^{k-1} \pmod{p} \cdots (1)$$

【００１９】
４．ディーラは、上記（１）式のx_iに各サーバＩＤを代入して分散値W_iを計算し、各サーバに（x_i、W_i）を配布する。
【００２０】
＜復元＞
１．復元に用いる分散値をW_i（ｉ＝１、２、…、ｋ）とする。また、その分散値に対応するサーバＩＤをx_i（ｉ＝１、２、…、ｋ）とする。
２．復元者は、ｋ個の（x_i、W_i）を集め、集めたｋ個の（x_i、W_i）を上記（１）式に代入し、ｋ個の連立方程式を解くことによって秘密情報であるｓを復元する。なお、ｓを復元する際には、例えば、Ｌａｇｒａｎｇｅの補間公式を用いることができる。
【００２１】
＜非対称秘密分散法＞
クラウドシステムを構成するｎ台のサーバからｔ台(ｔ＜ｋ)のサーバを選択し、選択したｔ台のサーバを鍵サーバとする。鍵サーバは、分散値を保管せず、擬似乱数を生成するための鍵情報を保管する。ｎ台のサーバのうち、鍵サーバ以外のサーバをデータサーバと呼び、データサーバは、分散値を生成するユーザから送られた分散値を保管する。また、秘密情報を管理する各ユーザｙには、ユーザを識別する情報であるＩＤ［ｙ］（ｙ＝１、…、ｒ）が割り当てられているものとする。また、それぞれのユーザｙが管理するｍ個の秘密情報s_{1j}、…、s_{mj}（ｊ＝１、…、ｒ）にも、それぞれ秘密情報を識別する情報であるｄＩＤ［s_{ij}］（ｉ＝１、…、ｍ）が割り当てられているものとする。また、以下では、Ｅｎｃ（ａ、ｂ）は、ａをｂという鍵を用いて暗号化する処理を表すものとする。
【００２２】
＜分散＞
ここでは、ユーザｙがディーラとなる場合を説明する。また、ここでは、説明を分かり易くするために、ｒ＝１とし、s_{ij}をs_iと表す。
１．ユーザｙは、自身のＩＤ［ｙ］を鍵サーバx_1、…、x_tに送る。
２．ＩＤ［ｙ］を受け取った鍵サーバは、自身が有する暗号装置と鍵key_j（ｊ＝１、…、ｔ）とを用いて、以下の（２）式により、Ｅｉｄ（ｙ、ｊ）を生成し、生成したＥｉｄ（ｙ、ｊ）をユーザｙに送る。

$$Eid(y,j)=Enc(ID[y],key_j)\quad (j=1,\cdots,t)\quad \cdots(2)$$

３．Ｅｉｄ（ｙ、ｊ）を受け取ったユーザｙは、自身が管理する秘密情報s_iを識別するｄＩＤ［s_i］（ｉ＝１、…、ｍ）を用いて、以下の（３）式により、擬似乱数q_{ij}を生成する。

$$q_{ij}=Enc(dID[s_i],Eid(y,j))\quad \cdots(3)$$

４．まず、ユーザｙは、式（１）のｋ－１次の分散式の係数ベクトルＡ（ｉ）＝［s_i、a_{i1}、…、a_{ik-1}］^{T}におけるｋ－１－ｔ次の部分ベクトルA'_{k-1-t}＝［a_{it+1}、…、a_{ik-1}］^{T}を、真性乱数を用いてｉごとに定める。次に、ユーザｙは、（３）式により生成したｔ次の擬似乱数系列Ｑ＝［q_{i1}、…、q_{it}］^{T}及び鍵サーバのＩＤ系列を用いて、以下の（４）式により、残りの部分ベクトルA'_t（ｉ）＝［a_{i1}、…、a_{it}］^{T}をｉごとに算出する。
【００２３】

【数２】

$$\begin{bmatrix} q_{i1} \\ \vdots \\ q_{it} \end{bmatrix} = \begin{bmatrix} S_i \\ \vdots \\ S_i \end{bmatrix} + \begin{bmatrix} x_1 & \cdots & x_1^{k-1} \\ \vdots & \ddots & \vdots \\ x_t & \cdots & x_t^{k-1} \end{bmatrix} + \begin{bmatrix} a_{i1} \\ \vdots \\ a_{ik-1} \end{bmatrix} \cdots (4)$$

【００２４】

ここで、上記（４）式においてＳ＝［ｓ$_{i}$、…、ｓ$_{i}$］^{T}とすると、以下の（５）式で表される。

$A(i)_{k-1}=X'^{-1}(Q-S)$　…(5)

これにより、ユーザｙは、ｋ－１次の分散式の係数ベクトルＡ（ｉ）＝［ｓ$_{i}$、ａ$_{i1}$、…、ａ$_{ik-1}$］^{T}におけるｋ－１次の部分ベクトルＡ（ｉ）$_{k-1}$＝［ａ$_{i1}$、…、ａ$_{ik-1}$］^{T}を全て求めることができる。

５．また、ユーザｙは、データサーバｘ$_{t+1}$、…、ｘ$_{n}$に関する分散値Ｗ$_{it+1}$、…、Ｗ$_{in}$を、４．の手順で生成した係数行列を用いて、（ｋ、ｎ）閾値秘密分散法と同様の手順により算出する。

６．ユーザｙは、各データサーバに、生成した分散値Ｗ$_{1j}$、…、Ｗ$_{mj}$（ｊ＝ｔ＋１、…、ｎ）を送る。

【００２５】

＜復元＞

１．復元者は、ｎ個のサーバｘ$_{1}$、…、ｘ$_{n}$から任意のｋ個のサーバを選択し、選択したサーバに対して、ユーザｙのＩＤ［ｙ］及び秘密情報ｓ$_{i}$のｄＩＤ［ｓ$_{i}$］を送る。

２．鍵サーバのうち、（ＩＤ［ｙ］、ｄＩＤ［ｓ$_{i}$］）を受け取ったサーバは、自身の持つ鍵ｋｅｙ$_{j}$及び暗号装置を利用して、（２）式によりＥｉｄ（ｙ、ｊ）を生成し、（３）式により擬似乱数ｑ$_{ij}$を生成する。そして、サーバは、生成した擬似乱数ｑ$_{ij}$を復元者に送る。

３．データサーバのうち、（ＩＤ［ｙ］、ｄＩＤ［ｓ$_{i}$］）を受け取ったサーバは、自身のサーバＩＤに対応する分散値Ｗ$_{ij}$を復元者に送る。

４．復元者は、受け取った擬似乱数ｑ$_{ij}$及び分散値Ｗ$_{ij}$を用いて、（ｋ、ｎ）閾値秘密分散法と同様の手順により、秘密情報ｓ$_{i}$を復元する。

【００２６】

しかしながら、Ｓｈａｍｉｒ法をはじめとする秘密分散法では、正当な復元者ではない攻撃者が、分散値が保管されているサーバを攻撃し、ｋ個以上の分散値を得ることができれば、そのｋ個以上の分散値から秘密情報を復元できてしまう。また、非対称秘密分散法でも、攻撃者が鍵サーバ及びデータサーバにアクセスできる状態にあり、鍵を用いて生成した分散値を含め、ｋ個の分散値を得ることができれば、秘密情報を復元することができてしまう。従って、Ｓｈａｍｉｒ法及び非対称秘密分散法を含む秘密分散法では、秘密情報のオーナが知らない間に、秘密情報が漏洩してしまう場合がある。

【００２７】

また、Ｓｈａｍｉｒ法及び非対称秘密分散法を含む秘密分散法では、攻撃者がサーバを乗っ取り、復元時に偽の分散値を出力した場合、復元者は、正しい秘密情報を復元できず、更に、復元した秘密情報の正当性を検証することもできない。

【００２８】

そこで、以下の実施形態では、以下の３つの問題点を解決する手法を説明する。

（Ａ）秘密情報のオーナが知らない間に、秘密情報が漏洩する。

（Ｂ）偽の分散値が出力されても、それを検証することができない。

一方、一般に暗号技術では暗号文に相当するものは１つであるため、ｎ＝ｋ＝１と解釈

できる。また、鍵を用いて暗号化しているために、ｋ＝１個の分散値に相当する暗号文が漏洩しても、鍵が安全に管理されていれば秘密情報は漏洩しない。従って、暗号技術では(Ａ)は解決されているが、偽の暗号文が出力されると正しく復号できず、復号結果が乱数等である場合、それが正しいか検証できないのが一般的である。よって、暗号技術では(Ｂ）は解決されていない。一般に、秘密分散法及び暗号技術では、秘密情報、分散値、及び暗号文に対してＭＡＣ（Message Authentication Code）等の仕組みの異なる技術を用いて復号結果の正当性を検証できるが、秘密分散法又は暗号技術のみでは復号結果の正当性を検証することはできない。よって、（Ｂ）と同様の原理によって以下の（Ｃ）も解決する。

（Ｃ）暗号技術において偽の暗号文が出力されても、それを検証することができない。

【００２９】

［第１実施形態］

まず、図１を参照して、本実施形態に係る秘密分散システム１０の構成を説明する。図１に示すように、秘密分散システム１０は、生成装置１２、復元装置１４、及び複数台のサーバ１６を含む。生成装置１２、復元装置１４、及び複数台のサーバ１６は、ネットワークＮに接続され、互いに通信可能とされる。本実施形態に係る秘密分散システム１０は、ｎを２以上の整数、ｋを２以上ｎ以下の整数とし、秘密情報をｎ個の分散値に分散し、ｋ個の分散値によって秘密情報を復元でき、ｋ個未満では秘密情報を復元できないシステムである。

【００３０】

生成装置１２は、秘密情報のオーナ（以下、単に「オーナ」という）によって操作される装置であり、生成装置１２の例としては、パーソナルコンピュータ等の情報処理装置が挙げられる。復元装置１４は、秘密情報を復元する復元者（以下、単に「復元者」という）によって操作される装置であり、復元装置１４の例としては、パーソナルコンピュータ等の情報処理装置が挙げられる。

【００３１】

前述したＳｈａｍｉｒ法において（１）式を用いて分散値W_iを生成する場合、（１）式にサーバＩＤであるx_iを代入し、x_iに対応する分散値W_iを得る。従って、x_iは、分散値W_iを識別する値であると言える。一般に、x_iが分からなければ、分散値W_iをｋ個集めても秘密情報を復元することはできない。そこで、本実施形態では、以下に示す処理によって、攻撃者がｋ個の分散値W_iを集めたとしても、秘密情報を復元できないようにする。

【００３２】

以下では、生成装置１２は、１つの秘密鍵key_0、及びｍ（ｍは１以上の整数）個の秘密情報s_1、…、s_mを保管し、各秘密情報s_j（ｊ＝１、…、ｍ）には、秘密情報s_jを識別する識別情報の一例としてのＤＩＤ［s_j］が割り振られているものとする。また、前述したx_iを秘密情報毎に区別するためにx_{ij}（ｉ＝１、…、ｎ、ｊ＝１、…、ｍ）と表記する。また、分散値を保管するサーバ１６（以下、「データサーバ１６」ともいう）はｎ台準備されている。また、２≦ｔ≦ｋ－１とし、ｎ－ｔ（＜ｋ）台のサーバ１６のx_{ij}は、予め定められているものとする。また、以下では、Ｅｎｃ（ａ、ｂ）は、ａをｂという鍵を用いて暗号化することを意味し、ａ｜ｂは、ａの後にｂを連結することを意味する。

【００３３】

次に、図２を参照して、本実施形態に係る生成装置１２のハードウェア構成を説明する。図２に示すように、生成装置１２は、ＣＰＵ（Central Processing Unit）２０、一時記憶領域としてのメモリ２１、不揮発性の記憶部２２を含む。また、生成装置１２は、液晶ディスプレイ等の表示装置２３、キーボード等の入力装置２４、及びネットワークＮに接続されるネットワークＩ／Ｆ２５（InterFace）を含む。ＣＰＵ２０、メモリ２１、記憶部２２、表示装置２３、入力装置２４、及びネットワークＩ／Ｆ２５は、バス２６に接続される。

【００３４】
記憶部２２は、ＨＤＤ（Hard Disk Drive）、ＳＳＤ（Solid State Drive）、及びフラッシュメモリ等によって実現される。記憶媒体としての記憶部２２には、生成プログラム２８が記憶される。ＣＰＵ２０は、記憶部２２から生成プログラム２８を読み出してからメモリ２１に展開し、展開した生成プログラム２８を実行する。ＣＰＵ２０が生成プログラム２８を実行することで、開示の技術の生成部として動作する。また、記憶部２２には、前述した秘密鍵key_0及び秘密情報s_jが記憶される。
【００３５】
次に、図３を参照して、本実施形態に係る復元装置１４のハードウェア構成を説明する。図３に示すように、復元装置１４は、ＣＰＵ３０、一時記憶領域としてのメモリ３１、不揮発性の記憶部３２を含む。また、復元装置１４は、液晶ディスプレイ等の表示装置３３、キーボード等の入力装置３４、及びネットワークＮに接続されるネットワークＩ／Ｆ３５を含む。ＣＰＵ３０、メモリ３１、記憶部３２、表示装置３３、入力装置３４、及びネットワークＩ／Ｆ３５は、バス３６に接続される。
【００３６】
記憶部３２は、ＨＤＤ、ＳＳＤ、及びフラッシュメモリ等によって実現される。記憶媒体としての記憶部３２には、復元プログラム３８が記憶される。ＣＰＵ３０は、記憶部３２から復元プログラム３８を読み出してからメモリ３１に展開し、展開した復元プログラム３８を実行する。ＣＰＵ３０が復元プログラム３８を実行することで、開示の技術の受信部及び復元部として動作する。
【００３７】
次に、図４及び図５を参照して、本実施形態に係る秘密分散システム１０の作用を説明する。まず、図４を参照して、秘密情報を分散する分散処理を説明する。生成装置１２のＣＰＵ２０が生成プログラム２８を実行することで、図４に示す分散処理を実行する。
【００３８】
図４のステップＳ１０で、ＣＰＵ２０は、以下に示す（６）式に従って、秘密情報s_jを識別する識別情報ｄＩＤ［s_j］を、１つの秘密鍵key_0を用いて変換することによって、ｔ台のサーバ１６に対応するｔ個のx_{ij}を生成する。このx_{ij}が分散値W_{ij}の各々に対応する値の一例である。

$$x_{ij}=Enc(dID[s_j]|i,key_0)\quad(i=1,\cdots,t,j=1,\cdots,m)\quad\cdots(6)$$

【００３９】
ステップＳ１２で、ＣＰＵ２０は、ｋ－１個の乱数a_l（ｌ＝１、…、ｋ－１）を生成し、分散式（上記（１）式）を決定する。
【００４０】
ステップＳ１４で、ＣＰＵ２０は、ステップＳ１０で生成したｔ個のx_{ij}と、予め定められたｎ－ｔ個のx_{ij}とを用いて、ステップＳ１２で決定した分散式に従って、各x_{ij}に対応する分散値W_{ij}を算出する。ステップＳ１６で、ＣＰＵ２０は、ステップＳ１４で算出した分散値W_{ij}を、対応するサーバ１６に送信する。各サーバ１６は、受信した分散値W_{ij}を保管する。ステップＳ１６の処理が終了すると、本分散処理が終了する。
【００４１】
次に、図５を参照して、図４に示す分散処理によって分散された分散値から、秘密情報を復元する復元処理を説明する。復元装置１４のＣＰＵ３０が復元プログラム３８を実行することで、図５に示す復元処理を実行する。
【００４２】
図５のステップＳ２０で、ＣＰＵ３０は、ｄＩＤ［s_j］に対応する分散値W_{ij}をｋ台のサーバ１６から取得する。ステップＳ２２で、ＣＰＵ３０は、x_{ij}が予め定められているｎ－ｔ台のサーバ１６以外のサーバ１６に対応するx_{ij}の送信を要求する要求情報を生成装置１２に送信する。
【００４３】

オーナは、生成装置１２が要求情報を受信した後、秘密情報の復元を許可する場合は、入力装置２４を介して復元を許可する操作を行い、秘密情報の復元を許可しない場合は、入力装置２４を介して復元を禁止する操作を行う。生成装置１２のＣＰＵ２０は、オーナにより復元を許可する操作が行われた場合に、ステップＳ１０と同様の処理によりx_{ij}を生成し、生成したx_{ij}を復元装置１４に送信する。また、ＣＰＵ２０は、オーナにより復元を禁止する操作が行われた場合は、x_{ij}を生成しない。すなわち、この場合、ＣＰＵ２０は、x_{ij}を復元装置１４に送信しない。ただし、この送信処理はオーナによる入力装置２４を介した操作がなくても、オーナが予め許可する条件を設定しておき、要求情報がその条件を満たす場合に自動的に送信を行い、満たさない場合に送信しないよう予めプログラム等により設定しておいても良い。これは、以降も同様である。

【００４４】

ステップＳ２４で、ＣＰＵ３０は、ステップＳ２２での要求に対応して生成装置１２から送信されたx_{ij}を受信したか否かを判定する。この判定が肯定判定となった場合は、処理はステップＳ２６に移行する。

【００４５】

ステップＳ２６で、ＣＰＵ３０は、ステップＳ２０で取得した分散値W_{ij}及びステップＳ２４で受信したx_{ij}を用いて、（ｋ、ｎ）閾値秘密分散法と同様の手順により、秘密情報s_jを復元する。ステップＳ２６の処理が終了すると、本復元処理が終了する。

【００４６】

一方、例えば、オーナが秘密情報の復元を禁止する操作を行ったことにより、ステップＳ２２でx_{ij}の送信を要求してから所定期間を経過してもx_{ij}が受信されなかった場合、または拒絶信号が送付された場合、ステップＳ２４の判定が否定判定となる。ステップＳ２４の判定が否定判定となった場合は、ステップＳ２６の処理は実行されずに本復元処理が終了する。

【００４７】

ところで、Ｌａｇｒａｎｇｅの補間公式では、ｘ座標が異なるｋ個の点（x_1、y_1）、…、（x_k、y_k）から、それらの点を通るｋ－１次以下の多項式Ｗ（ｘ）を求めることができる。前述した非対称秘密分散法の分散の４．の処理は、Ｌａｇｒａｎｇｅの補間公式によって多項式を求めることを意味するが、ｘとｙとの関係は入れ替えてもよい。このため、ｙ座標が異なるｋ個の点（x_1、y_1）、…、（x_k、y_k）から、それらの点を通るｋ－１次以下の多項式Ｆ（ｙ）を求めることもでき、求めた曲線はＷ（ｘ）＝Ｆ（ｙ）となる。すなわち、本実施形態は、ｘ座標を公開し、ｙ座標を分散値として生成・秘匿していた従来の秘密分散法に対して、生成した分散値を公開し、その基となるｘ座標を生成・秘匿することに対応する。また、オーナは、ステップＳ２２で送信された要求情報に対して復元を許可しない場合、復元を禁止する操作を行うことによってx_{ij}を復元装置１４に送信しないことにより、秘密情報の復元を制御することができる。この結果、秘密情報のオーナが知らない間に、秘密情報が漏洩することを抑制することができる。また、非対称秘密分散法と同等以上の安全性を確保することができる。

【００４８】

なお、本実施形態では、分散処理のステップＳ１０で、ｄＩＤ［s_j］とｉとを連結して得られた情報を、key_0を用いて暗号化することによってx_{ij}を生成する場合について説明したが、これに限定されない。例えば、ｉを、key_0を用いて暗号化することによってkey_iを生成し、ｄＩＤ［s_j］を、key_iを用いて暗号化することによってx_{ij}を生成してもよい。また、例えば、ｄＩＤ［s_j］を、key_0を用いて暗号化することによってkey_jを生成し、ｉを、key_jを用いて暗号化することによってx_{ij}を生成してもよい。

【００４９】

また、本実施形態では、ｔ個のx_{ij}を生成し、残りのｎ－ｔ個のx_{ij}が予め定められている場合について説明したが、これに限定されない。分散処理のステップＳ１０で、ｎ個全てのx_{ij}（ｉ＝１、…、ｎ）を生成してもよい。この場合、復元処理のステップ

Ｓ２２で、復元に必要な全てのx_{ij}の送信を要求する要求情報を生成装置１２に送信する形態が例示される。

【００５０】

［第２実施形態］

開示の技術の第２実施形態を説明する。なお、秘密分散システム１０の構成（図１参照）、生成装置１２のハードウェア構成（図２参照）、及び復元装置１４のハードウェア構成（図３参照）は、第１実施形態と同様であるため説明を省略する。

【００５１】

前述した非対称秘密分散法は、復元時に秘密情報のオーナによる復元の許諾処理を導入していないため、鍵サーバを含めてｋ個の分散値を得ることができたユーザは、攻撃者であっても秘密情報を復元できる。また、ユーザが複数人いることを想定した場合、特定のオーナが鍵サーバを管理することができない。また、ユーザを１人とし、そのユーザ（すなわち、秘密情報のオーナ）が全ての鍵サーバを管理する、すなわち、非対称秘密分散法の＜分散＞の１．及び２．の処理をオーナが行えば、ｋ－１個までの分散値をオーナが生成できるが、＜復元＞においてオーナによる復元の許諾処理を含んでいない。また、非対称秘密分散法では、鍵サーバがあることを前提とするため、鍵サーバが保管する鍵を用いて分散値を生成する手順に固定されている。本実施形態では、より汎用的な形とした手法を説明する。この手法により、上記（Ａ）の問題点が汎用的に解決される。

【００５２】

なお、本実施形態では、全てのサーバ１６のサーバＩＤ（x_{ij}）は予め定められており、公開されているものとする。また、本実施形態では、サーバ１６のうち、鍵サーバとして機能するサーバ１６を「鍵サーバ１６」といい、データサーバとして機能するサーバ１６を「データサーバ１６」という。ただし、鍵サーバ１６は生成装置１２内に存在するため、図において明示されているサーバ１６はデータサーバ１６であり、ｎ－ｔ台準備されている。それ以外の前提は第１実施形態と同様である。

【００５３】

図６及び図７を参照して、本実施形態に係る秘密分散システム１０の作用を説明する。まず、図６を参照して、本実施形態に係る秘密情報を分散する分散処理を説明する。生成装置１２のＣＰＵ２０が生成プログラム２８を実行することで、図６に示す分散処理を実行する。

【００５４】

図６のステップＳ３０で、ＣＰＵ２０は、以下に示す（７）式に従って、秘密情報s_jを識別する識別情報ｄＩＤ［s_j］を、１つの秘密鍵key_0を用いて変換することによって、ｔ個のx_{ij}（ｉ＝１、…、ｔ）に対応する分散値q_{ij}を生成する。

$$q_{ij}=Enc(dID[s_j]|i,key_0) \quad (i=1,\cdots,t,j=1,\cdots,m) \quad \cdots(7)$$

【００５５】

ステップＳ３２で、ＣＰＵ２０は、上記（１）式のｋ－１次の分散式の係数ベクトルＡ（ｉ）＝［s_i、a_{i1}、…、a_{ik-1}］^{T}におけるｋ－１－ｔ次の部分ベクトルA'_{k-1-t}＝［a_{it+1}、…、a_{ik-1}］^{T}を、真性乱数を用いてｉごとに定める。その後、ＣＰＵ２０は、ステップＳ３０で生成したｔ個の擬似乱数であるq_{ij}からなるＱ＝［q_{i1}、…、q_{it}］^{T}及び鍵サーバ１６のＩＤ系列を用いて、以下の（８）式に従って、分散式のｋ－１次の部分ベクトル$A(i)_{k-1}$＝［a_{i1}、…、a_{ik-1}］^{T}における残りの部分ベクトル$A'_t(i)$＝［a_{i1}、…、a_{it}］^{T}をｉごとに算出する。

【００５６】

【数３】

$$\begin{bmatrix} q_{i1} \\ \vdots \\ q_{it} \end{bmatrix} = \begin{bmatrix} S_i \\ \vdots \\ S_i \end{bmatrix} + \begin{bmatrix} x_1 & \cdots & x_1^{k-1} \\ \vdots & \ddots & \vdots \\ x_t & \cdots & x_t^{k-1} \end{bmatrix} + \begin{bmatrix} a_{i1} \\ \vdots \\ a_{ik-1} \end{bmatrix} \cdots (8)$$

【００５７】

ステップＳ３４で、ＣＰＵ２０は、サーバＩＤがx_{it+1}、…、x_{in}であるデータサーバ１６に関する分散値W_{it+1}、…、W_{in}を、ステップＳ３２で生成した係数行列を用いて、（ｋ、ｎ）閾値秘密分散法と同様の手順により算出する。

【００５８】

ステップＳ３６で、ＣＰＵ２０は、ステップＳ３４で算出した分散値W_{ij}（ｊ＝ｔ＋１、…、ｎ）を、対応するデータサーバ１６に送信する。各データサーバ１６は、受信した分散値W_{ij}を保管する。ステップＳ３６の処理が終了すると、本分散処理が終了する。

【００５９】

次に、図７を参照して、図６に示す分散処理によって分散された分散値から、秘密情報を復元する復元処理を説明する。復元装置１４のＣＰＵ３０が復元プログラム３８を実行することで、図７に示す復元処理を実行する。

【００６０】

図７のステップＳ４０で、ＣＰＵ３０は、ｄＩＤ［s_j］に対応する分散値の送信を要求する要求情報を生成装置１２に送信する。オーナは、生成装置１２が要求情報を受信した後、秘密情報の復元を許可する場合は、入力装置２４を介して復元を許可する操作を行い、秘密情報の復元を許可しない場合は、入力装置２４を介して復元を禁止する操作を行う。生成装置１２のＣＰＵ２０は、オーナにより復元を許可する操作が行われた場合に、ステップＳ３０と同様の処理により分散値q_{ij}を生成し、生成した分散値q_{ij}を復元装置１４に送信する。また、ＣＰＵ２０は、オーナにより復元を禁止する操作が行われた場合は、分散値q_{ij}を生成しない。すなわち、この場合、ＣＰＵ２０は、分散値q_{ij}を復元装置１４に送信しない。

【００６１】

ステップＳ４２で、ＣＰＵ３０は、ステップＳ４０での要求に対応して生成装置１２から送信された分散値q_{ij}を受信したか否かを判定する。この判定が肯定判定となった場合は、処理はステップＳ４４に移行する。

【００６２】

ステップＳ４４で、ＣＰＵ３０は、データサーバ１６のW_{ij}と合わせて得られたｋ個の分散値を用いて、秘密情報s_jを復元する。ステップＳ４４の処理が終了すると、本復元処理が終了する。

【００６３】

一方、例えば、ステップＳ４０で分散値q_{ij}の送信を要求してから所定期間を経過しても分散値q_{ij}が受信されなかった場合等では、ステップＳ４２の判定が否定判定となる。ステップＳ４２の判定が否定判定となった場合は、ステップＳ４４の処理は実行されずに本復元処理が終了する。

【００６４】

なお、本実施形態では、分散処理のステップＳ３０で、ｄＩＤ［s_j］とｉとを連結して得られた情報を、key_0を用いて暗号化することによってq_{ij}を生成する場合について説明したが、これに限定されない。例えば、ｉを、key_0を用いて暗号化することによってkey_iを生成し、ｄＩＤ［s_j］を、key_iを用いて暗号化することによっ

てｑ$_{ij}$を生成してもよい。また、例えば、ｄＩＤ［ｓ$_j$］を、ｋｅｙ$_0$を用いて暗号化することによってｋｅｙ$_j$を生成し、ｉを、ｋｅｙ$_j$を用いて暗号化することによってｑ$_{ij}$を生成してもよい。

【００６５】

［第３実施形態］

開示の技術の第３実施形態を説明する。なお、秘密分散システム１０の構成（図１参照）、生成装置１２のハードウェア構成（図２参照）、及び復元装置１４のハードウェア構成（図３参照）は、第１実施形態と同様であるため説明を省略する。

【００６６】

第１実施形態では分散値を識別するｘ$_{ij}$を１つの秘密鍵ｋｅｙ$_0$を用いて生成し、第２実施形態では分散値ｑ$_{ij}$を１つの秘密鍵ｋｅｙ$_0$を用いて生成する形態例を説明したが、本実施形態では、これらの形態例を組み合わせた形態例を説明する。この組み合わせによって、上記（Ａ）の問題点に加え、上記（Ｂ）の問題点も解決することができる。

【００６７】

一般に、秘密情報ｓは、ｋ個の分散値ｙ$_i$と、分散値ｙ$_i$を識別するｋ個の値ｘ$_i$とを用いて、Ｌａｇｒａｎｇｅの補間公式から以下に示すように求められる。ｘ座標が異なるｋ個の点（ｘ$_1$、ｙ$_1$）、（ｘ$_2$、ｙ$_2$）、…、（ｘ$_k$、ｙ$_k$）を通るｋ－１次以下の多項式Ｗ（ｘ）は以下の（９）式で表される。

【００６８】

【数４】

$$W(x)=\sum_{i=1}^{n} y_i \frac{f_i(x)}{f_i(x_i)} \cdots (9)$$

【００６９】

ただし、

【００７０】

【数５】

$$f_i(x)=\prod_{k\neq i}(x-x_k)$$

【００７１】

とする。従って、秘密情報ｓはＷ（０）の値となる。ここで、サーバ１６が攻撃者に乗っ取られ、乗っ取られたサーバ１６が偽の値ｙ$_i$＋Δｙ$_i$を出力した場合、偽のＷ’（０）は以下に示す（１０）式で表される。

【００７２】

【数６】

$$W'(0)=W(0)+\sum_{i=1}^{n} \Delta y_i \frac{f_i(0)}{f_i(x_i)} \cdots (10)$$

【００７３】

例えば、１回目の復元で偽の分散値ｙ$_i$＋Δｙ$_i$がサーバ１６から出力されてＷ’（０）が復元され、２回目の復元で１回目とは異なる偽のｙ$_j$＋Δｙ$_j$がサーバ１６から出力されてＷ’’（０）が復元された場合、その差分は以下に示す（１１）式で表される。

【００７４】

【数７】

$$W'(0)-W''(0)=\Delta y_i\frac{f_i(0)}{f_i(x_i)}-\Delta y_j\frac{f_j(0)}{f_j(x_j)}\cdots(11)$$

【００７５】

従って、Δｙ$_i$とΔｙ$_j$とに以下に示す（１２）式の関係があれば、Ｗ’（０）とＷ’’（０）との差分は０となる、すなわち、１回目と２回目とにそれぞれ復元された秘密情報が一致する。このため、この場合、復元された秘密情報が偽りの値であることは判定できない。

【００７６】

【数８】

$$\Delta y_i=\Delta y_j\frac{f_j(0)}{f_j(x_j)}\frac{f_i(x_i)}{f_i(0)}\cdots(12)$$

【００７７】

しかしながら、

【００７８】

【数９】

$$f_i(x)=\prod_{k\neq i}(x-x_k)$$

【００７９】

は、全てのｘ$_k$が特定できなければ定まらない。従って、攻撃者にとって未知のｘ$_k$が存在する場合、攻撃者は乗っ取ったサーバ１６に対して、Ｗ’（０）とＷ’’（０）との差分が０となるΔｙ$_i$及びΔｙ$_j$を設定することができない。

【００８０】

そこで、本実施形態では、以下に示す処理を行うことによって、秘密情報の漏洩に関する上記（Ａ）の問題点、及び秘密情報の改ざんに関する上記（Ｂ）の問題点の双方を解決する。なお、以下では、第２実施形態と同様にサーバ１６はデータサーバ１６のみでｎ－ｔ台準備され、そのデータサーバ１６のサーバＩＤであるｘ$_{ij}$は予め定められ、かつ公開されているものとする。また、ｎ＞ｋであり、これら以外の前提は第１実施形態と同様とする。

【００８１】

図８及び図９を参照して、本実施形態に係る秘密分散システム１０の作用を説明する。まず、図８を参照して、本実施形態に係る秘密情報を分散する分散処理を説明する。生成装置１２のＣＰＵ２０が生成プログラム２８を実行することで、図８に示す分散処理を実行する。

【００８２】

図８のステップＳ５０で、ＣＰＵ２０は、以下に示す（１３）式に従って、秘密情報ｓ$_j$を識別する識別情報ｄＩＤ［ｓ$_j$］を、１つの秘密鍵ｋｅｙ$_0$を用いて変換することによって、ｔ台のサーバ１６に対応するｔ個のｘ$_{ij}$を生成する。

$x_{ij}=\mathrm{Enc}(dID[s_j]||i|0,key_0)\quad(i=1,\cdots,t,j=1,\cdots,m)\quad\cdots(13)$

【００８３】

ステップＳ５２で、ＣＰＵ２０は、以下に示す（１４）式に従って、秘密情報s_jを識別する識別情報ｄＩＤ［s_j］を、１つの秘密鍵key_0を用いて変換することによって、ｔ個のx_{ij}（ｉ＝１、…、ｔ）に対応する分散値q_{ij}を生成する。

$$q_{ij}=Enc(dID[s_j]||i|1,key_0) \quad (i=1,\cdots,t,j=1,\cdots,m) \quad \cdots(14)$$

【００８４】

このように、本実施形態では、ステップＳ５０で生成されるｔ（≦ｋ－１）個のx_{ij}は、連結させる値（x_{ij}は０、q_{ij}は１）を異ならせることにより、分散値q_{ij}とは異なる値として生成される。連結させる値は０及び１を用いたがこれに限られず、任意の値を用いてもよい。

【００８５】

ステップＳ５４で、ＣＰＵ２０は、ステップＳ５０、Ｓ５２で生成した値を用いて、ｔ個の分散値q_{ij}以外の分散値W_{it+1}、…、W_{in}を求めるための係数を算出する。具体的には、ＣＰＵ２０は、上記（１）式のｋ－１次の分散式の係数ベクトルＡ（ｉ）＝［s_i、a_{i1}、…、a_{ik-1}］^{T}におけるｋ－１－ｔ次の部分ベクトルＡ’$_{k-1-t}$＝［a_{it+1}、…、a_{ik-1}］^{T}を、真性乱数を用いてｉごとに定める。その後、ＣＰＵ２０は、ステップＳ５０で生成したｔ個の擬似乱数であるq_{ij}からなるＱ＝［q_{i1}、…、q_{it}］^{T}及びステップＳ５２で生成した鍵サーバ１６のＩＤ系列を用いて、以下の（１５）式に従って、分散式のｋ－１次の部分ベクトルＡ（ｉ）$_{k-1}$＝［a_{i1}、…、a_{ik-1}］^{T}における残りの部分ベクトルＡ’$_t$（ｉ）＝［a_{i1}、…、a_{it}］^{T}をｉごとに算出する。

【００８６】

【数１０】

$$\begin{bmatrix} q_{i1} \\ \vdots \\ q_{it} \end{bmatrix} = \begin{bmatrix} S_i \\ \vdots \\ S_i \end{bmatrix} + \begin{bmatrix} x_1 & \cdots & x_1^{k-1} \\ \vdots & \ddots & \vdots \\ x_t & \cdots & x_t^{k-1} \end{bmatrix} + \begin{bmatrix} a_{i1} \\ \vdots \\ a_{ik-1} \end{bmatrix} \cdots(15)$$

【００８７】

ステップＳ５６で、ＣＰＵ２０は、サーバＩＤがx_{it+1}、…、x_{in}であるデータサーバ１６に関する分散値W_{it+1}、…、W_{in}を、ステップＳ５４で生成した係数行列を用いて、（ｋ、ｎ）閾値秘密分散法と同様の手順により算出する。

【００８８】

ステップＳ５８で、ＣＰＵ２０は、ステップＳ５６で算出した分散値W_{ij}（ｊ＝ｔ＋１、…、ｎ）を、対応するデータサーバ１６に送信する。各データサーバ１６は、受信した分散値W_{ij}を保管する。ステップＳ５８の処理が終了すると、本分散処理が終了する。

【００８９】

次に、図９を参照して、図８に示す分散処理によって分散された分散値から、秘密情報を復元する復元処理を説明する。復元装置１４のＣＰＵ３０が復元プログラム３８を実行することで、図９に示す復元処理を実行する。

【００９０】

図９のステップＳ６０で、ＣＰＵ３０は、データサーバ１６に保管されているｋ－ｔ＋ｕ個の分散値W_{ij}を受信する。なお、ｕは、検証用の分散値の数（１以上の整数）を表す。ステップＳ６２で、ＣＰＵ３０は、ｔ個の分散値と、その分散値を識別するためのx_{ij}との送信を要求する要求情報を生成装置１２に送信する。オーナは、生成装置１２が要求情報を受信した後、秘密情報の復元を許可する場合は、入力装置２４を介して復元を許可する操作を行い、秘密情報の復元を許可しない場合は、入力装置２４を介して復元を

禁止する操作を行う。生成装置１２のＣＰＵ２０は、オーナにより復元を許可する操作が行われた場合に、ステップＳ５０、Ｓ５２と同様の処理によりx_{ij}及び分散値q_{ij}を生成し、生成したx_{ij}及び分散値q_{ij}を復元装置１４に送信する。また、ＣＰＵ２０は、オーナにより復元を禁止する操作が行われた場合は、x_{ij}及び分散値q_{ij}を生成しない。すなわち、この場合、ＣＰＵ２０は、x_{ij}及び分散値q_{ij}を復元装置１４に送信しない。

【００９１】

ステップＳ６４で、ＣＰＵ３０は、ステップＳ６２での要求に対応して生成装置１２から送信されたx_{ij}及び分散値q_{ij}を受信したか否かを判定する。この判定が肯定判定となった場合は、処理はステップＳ６６に移行する。

【００９２】

ステップＳ６６で、ＣＰＵ３０は、得られたｋ＋ｕ個のq_{ij}、W_{ij}、及びx_{ij}を用いて、データサーバ１６から得られた分散値W_{ij}を異ならせて（q_{ij}は同一）、秘密情報s_jをｕ＋１回復元する。換言すると、ＣＰＵ３０は、ステップＳ５４で生成された係数を用いてステップＳ５６で同一の秘密情報から生成された異なる分散値の組み合わせによって、秘密情報を２回以上復元する。

【００９３】

ステップＳ６８で、ＣＰＵ３０は、ステップＳ６６で復元したｕ＋１個の秘密情報s_jを用いて、復元した秘密情報s_jを検証する。具体的には、ＣＰＵ３０は、ステップＳ６６で復元したｕ＋１個の秘密情報s_jが一致した場合は、その秘密情報s_jは正しいと判断し、異なる場合は、その秘密情報s_jは不正である、すなわち、データサーバ１６の少なくとも１台が攻撃されたと判断する。ＣＰＵ３０は、秘密情報s_jが正しいと判断した場合は、その秘密情報s_jを用いた処理を行う。一方、ＣＰＵ３０は、秘密情報s_jが不正であると判断した場合は、例えば、表示装置３３にエラーメッセージを表示する。ステップＳ６８の処理が終了すると、本復元処理が終了する。

【００９４】

一方、例えば、ステップＳ６２でx_{ij}及び分散値q_{ij}の送信を要求してから所定期間を経過してもx_{ij}及び分散値q_{ij}が受信されなかった場合等では、ステップＳ６４の判定が否定判定となる。ステップＳ６４の判定が否定判定となった場合は、ステップＳ６６、Ｓ６８の処理は実行されずに本復元処理が終了する。

【００９５】

以下に具体例を用いて説明する。例えば、ｋ＝３、ｎ＝４、ｔ＝２、ｕ＝ｎ－ｋ＝１とした場合、生成装置１２はｔ＝２より２台分の鍵サーバ１６を含むため、それをx_1、x_2とし、x_1、x_2に対応する分散値q_{ij}を生成する。また、この場合、データサーバ１６はｎ－ｔ＝２より２台存在し、そのサーバＩＤであるx_3、x_4は公開されている。この２台のデータサーバ１６が、攻撃者により乗っ取られ、偽の分散値$y_3+\Delta y_3$、$y_4+\Delta y_4$を出力する場合を考える。

【００９６】

この場合、復元装置１４は、外部からの分散値を異ならせて復元処理を行う、すなわち１回目はx_1、x_2、x_3の組み合わせによって秘密情報の復元を行い、２回目はx_1、x_2、x_4の組み合わせによって秘密情報の復元を行う。ここで、x_1、x_2は生成装置１２が生成した分散値を識別する値であるため、攻撃者はx_1、x_2を知ることはできない。この場合、１回目の復元によって、以下の（１６）式に示す偽の$W'(0)$が復元される。

【００９７】

【数１１】

$$W'(0)=s+\Delta y_3\frac{(-x_1)(-x_2)}{(x_3-x_1)(x_3-x_2)}\cdots(16)$$

【００９８】

また、この場合、２回目の復元によって、以下の（１７）式に示す偽のＷ’’（０）が復元される。

【００９９】

【数１２】

$$W''(0)=s+\Delta y_4\frac{(-x_1)(-x_2)}{(x_4-x_1)(x_4-x_2)}\cdots(17)$$

【０１００】

ここで、x_3、x_4に対応する２台のデータサーバ１６が同一の攻撃者によって乗っ取られたとしても、その攻撃者は、x_1、x_2を知ることはできないため、以下の（１８）式の関係のΔy_3及びΔy_4を設定することができない。このため、Ｗ’（０）とＷ’’（０）とは一致しない。従って、復元装置１４は、データサーバ１６から偽の分散値が出力されたことを検証することができる。

【０１０１】

【数１３】

$$\Delta y_3=\Delta y_4\frac{(x_3-x_1)(x_3-x_2)}{(x_4-x_1)(x_4-x_2)}\cdots(18)$$

【０１０２】

上記の例では、ｕ＝１とし、x_3に対応するデータサーバ１６の他に、x_4に対応するデータサーバ１６から検証用の分散値を得ている。この場合、２台のデータサーバ１６のうちの何れか、又は双方が不正であるかは判断できない。

【０１０３】

例えば、ｋ＝３、ｎ＝５、ｔ＝２、ｕ＝ｎ－ｋ＝２とすれば、３台のデータサーバ１６が存在し、x_3の他に２つの分散値を検証用に使用することができる。ここで、x_5に対応するデータサーバ１６のみが乗っ取られ、偽の分散値$y_5+\Delta y_5$を出力する場合を考える。この場合、x_3、x_4に対応する分散値を用いて２回の復元により得られた２つの秘密情報は、$\Delta y_3=\Delta y_4=0$となるため一致し、x_5に対応する分散値を用いて復元された秘密情報のみが他の秘密情報と異なる。すなわち、この場合、不正なデータサーバ１６を特定することができる。

【０１０４】

また、２台のデータサーバ１６が乗っ取られた場合、３回の復元により得られた３つの秘密情報が全て異なるため、この場合は、２台以上のデータサーバ１６が不正であると判断することができる。従って、乗っ取られて偽の分散値を出力するデータサーバ１６の数をｅとした場合、ｅ＜ｕであれば、不正なデータサーバ１６を特定することができる。

【０１０５】

なお、本実施形態では、分散処理のステップＳ５０、Ｓ５２において分散値q_{ij}と、分散値q_{ij}を識別するx_{ij}とを異なる数（本実施形態では０と１）を連結させることによって異なる値として生成する場合について説明したが、これに限定されない。x_{ij}

と分散値q_{ij}とは異なる値であれば、他の任意の手法によって生成してもよい。

【０１０６】

また、本実施形態では、分散処理のステップＳ５０、Ｓ５２においてx_{ij}及び分散値q_{ij}をｄＩＤ［s_j］、ｉ、及び０（又は１）を連結させ、かつ１回の暗号化によって生成する場合について説明したが、これに限定されない。例えば、連結させる組み合わせを本実施形態とは異ならせてもよいし、複数回の暗号化によってx_{ij}及び分散値q_{ij}を生成してもよい。

【０１０７】

また、第１～第３実施形態を通して、x_{ij}は秘密情報s_iごとに生成しているが、これは、１つの秘密情報の復元時に、生成装置１２からx_{ij}を得ることができても、異なるx_{ij}が設定されている他の秘密情報を復元できないようにするためである。これにより、安全性の低下を抑制することができる。また、攻撃者が分散値からx_{ij}を得ようとしても、分散値はｎ－ｔ（＜ｋ）台のデータサーバ１６にしか保管されていないため、同様にx_{ij}を得ることはできない。

【０１０８】

［第４実施形態］

開示の技術の第４実施形態を説明する。なお、秘密分散システム１０の構成（図１参照）、生成装置１２のハードウェア構成（図２参照）、及び復元装置１４のハードウェア構成（図３参照）は、第１実施形態と同様であるため説明を省略する。

【０１０９】

第３実施形態では、ｎ＞ｋを前提とし、検証用の分散値を使用する形態例を説明した。本実施形態では、ｎ＝ｋとし、検証用の分散値を使用しない形態例を説明する。本実施形態では、１つの秘密情報を異なる分散式で２回分散し、第３実施形態と同様に、２回の復元によって得られた秘密情報が一致するか否かによって検証する。

【０１１０】

図１０及び図１１を参照して、本実施形態に係る秘密分散システム１０の作用を説明する。まず、図１０を参照して、本実施形態に係る秘密情報を分散する分散処理を説明する。生成装置１２のＣＰＵ２０が生成プログラム２８を実行することで、図１０に示す分散処理を実行する。

【０１１１】

図１０のステップＳ７０で、ＣＰＵ２０は、以下に示す（１９）式及び（２０）式に従って、秘密情報s_jを識別する識別情報ｄＩＤ［s_j］を、１つの秘密鍵key_0を用いて変換することによって、x_{ij}及びx'_{ij}を生成する。このx_{ij}及びx'_{ij}は、互いに異なる２個のサーバＩＤに対応する。

$$x_{ij}=Enc(dID[s_j]||i|0,key_0) \quad (i=1,\cdots,t,j=1,\cdots,m) \quad \cdots(19)$$

$$x'_{ij}=Enc(dID[s_j]||i|2,key_0) \quad (i=1,\cdots,t,j=1,\cdots,m) \quad \cdots(20)$$

【０１１２】

ステップＳ７２で、ＣＰＵ２０は、以下に示す（２１）式に従って、秘密情報s_jを識別する識別情報ｄＩＤ［s_j］を、１つの秘密鍵key_0を用いて変換することによって、ｔ個のx_{ij}（ｉ＝１、…、ｔ）に対応する分散値q_{ij}を生成する。また、ＣＰＵ２０は、以下に示す（２２）式に従って、秘密情報s_jを識別する識別情報ｄＩＤ［s_j］を、１つの秘密鍵key_0を用いて変換することによって、ｔ個のx'_{ij}（ｉ＝１、…、ｔ）に対応する分散値q'_{ij}を生成する。

$$q_{ij}=Enc(dID[s_j]||i|1,key_0) \quad (i=1,\cdots,t,j=1,\cdots,m) \quad \cdots(21)$$

$$q'_{ij}=Enc(dID[s_j]||i|2,key_0) \quad (i=1,\cdots,t,j=1,\cdots,m) \quad \cdots(22)$$

【０１１３】

ステップＳ７４で、ＣＰＵ２０は、非対称秘密分散法の＜分散＞の４．の手順を用いて、ｔ個の（x_{ij}、q_{ij}）と（０、s_j）とを通るｋ－１次の多項式Ｗ（ｘ）を求める。多項式Ｗ（ｘ）は、以下の（２３）式で表される。

【０１１４】

【数１４】

$$W(x)=s_i+a_{i1}x+\cdots+a_{ik-1}x^{k-1}\cdots(23)$$

【０１１５】

ステップＳ７６で、ＣＰＵ２０は、非対称秘密分散法の＜分散＞の４．の手順を用いて、ｔ個の（ｘ’ｉｊ、ｑ’ｉｊ）と（０、ｓｊ）とを通るｋ－１次の多項式Ｖ（ｘ）を求める。多項式Ｖ（ｘ）は、以下の（２４）式で表される。

【０１１６】

【数１５】

$$V(x)=s_i+b_{i1}x+\cdots+b_{ik-1}x^{k-1}\cdots(24)$$

【０１１７】

ステップＳ７８で、ＣＰＵ２０は、サーバＩＤがｘｉｔ＋１、…、ｘｉｎであるデータサーバ１６に関する分散値Ｗｉｔ＋１、…、Ｗｉｎを、ステップＳ７４で算出した多項式Ｗ（ｘ）を用いて、（ｋ、ｎ）閾値秘密分散法と同様の手順により算出する。

【０１１８】

ステップＳ８０で、ＣＰＵ２０は、サーバＩＤがｘｉｔ＋１、…、ｘｉｎであるデータサーバ１６に関する分散値Ｖｉｔ＋１、…、Ｖｉｎを、ステップＳ７６で算出した多項式Ｖ（ｘ）を用いて、（ｋ、ｎ）閾値秘密分散法と同様の手順により算出する。

【０１１９】

ステップＳ８２で、ＣＰＵ２０は、ステップＳ７８、Ｓ８０で算出した分散値Ｗｉｊ、Ｖｉｊ（ｊ＝ｔ＋１、…、ｎ）を、対応するデータサーバ１６に送信する。各データサーバ１６は、受信した分散値Ｗｉｊ、Ｖｉｊを保管する。ステップＳ８２の処理が終了すると、本分散処理が終了する。

【０１２０】

次に、図１１を参照して、図１０に示す分散処理によって分散された分散値から、秘密情報を復元する復元処理を説明する。復元装置１４のＣＰＵ３０が復元プログラム３８を実行することで、図１１に示す復元処理を実行する。

【０１２１】

図１１のステップＳ９０で、ＣＰＵ３０は、データサーバ１６に保管されているｋ－ｔ個の分散値Ｗｉｊ、Ｖｉｊを受信する。ステップＳ９２で、ＣＰＵ３０は、ｔ個の分散値ｑｉｊ、ｑ’ｉｊと、その分散値ｑｉｊ、ｑ’ｉｊを識別するためのｘｉｊ、ｘ’ｉｊの送信を要求する要求情報を生成装置１２に送信する。オーナは、生成装置１２が要求情報を受信した後、秘密情報の復元を許可する場合は、入力装置２４を介して復元を許可する操作を行い、秘密情報の復元を許可しない場合は、入力装置２４を介して復元を禁止する操作を行う。生成装置１２のＣＰＵ２０は、オーナにより復元を許可する操作が行われた場合に、ステップＳ７０、Ｓ７２と同様の処理によりｘｉｊ、ｘ’ｉｊ及び分散値ｑｉｊ、ｑ’ｉｊを生成し、生成したｘｉｊ、ｘ’ｉｊ及び分散値ｑｉｊ、ｑ’ｉｊを復元装置１４に送信する。また、ＣＰＵ２０は、オーナにより復元を禁止する操作が行われた場合は、ｘｉｊ、ｘ’ｉｊ及び分散値ｑｉｊ、ｑ’ｉｊを生成しない。すなわち、この場合、ＣＰＵ２０は、ｘｉｊ、ｘ’ｉｊ及び分散値ｑｉｊ、ｑ’ｉｊを復元装置１４に送信しない。

【０１２２】

ステップＳ９４で、ＣＰＵ３０は、ステップＳ９２での要求に対応して生成装置１２から送信されたｘｉｊ、ｘ’ｉｊ及び分散値ｑｉｊ、ｑ’ｉｊを受信したか否かを判定する。この判定が肯定判定となった場合は、処理はステップＳ９６に移行する。

【０１２３】

ステップＳ９６で、ＣＰＵ３０は、得られたｋ個のｑｉｊ、Ｗｉｊ、及びｘｉｊを用い

て、多項式Ｗ（ｘ）により秘密情報ｓ$_{j}$を復元する。また、ＣＰＵ３０は、得られたｋ個のｑ’$_{ij}$、Ｖ$_{ij}$、及びｘ’$_{ij}$を用いて、多項式Ｖ（ｘ）により秘密情報ｓ$_{j}$を復元する。

【０１２４】

ステップＳ９８で、ＣＰＵ３０は、ステップＳ９６で復元した２個の秘密情報ｓ$_{j}$を用いて、復元した秘密情報ｓ$_{j}$を検証する。具体的には、ＣＰＵ３０は、多項式Ｗ（ｘ）により復元した秘密情報ｓ$_{j}$と多項式Ｖ（ｘ）により復元した秘密情報ｓ$_{j}$とが一致した場合、その秘密情報ｓ$_{j}$が正しいと判断する。一方、ＣＰＵ３０は、多項式Ｗ（ｘ）により復元した秘密情報ｓ$_{j}$と多項式Ｖ（ｘ）により復元した秘密情報ｓ$_{j}$とが異なる場合、その秘密情報ｓ$_{j}$は不正である、すなわち、データサーバ１６の少なくとも１台が攻撃されたと判断する。ＣＰＵ３０は、秘密情報ｓ$_{j}$が正しいと判断した場合は、その秘密情報ｓ$_{j}$を用いた処理を行う。一方、ＣＰＵ３０は、秘密情報ｓ$_{j}$が不正であると判断した場合は、例えば、表示装置３３にエラーメッセージを表示する。ステップＳ９８の処理が終了すると、本復元処理が終了する。

【０１２５】

一方、例えば、ステップＳ９２でｘ$_{ij}$、ｘ’$_{ij}$及び分散値ｑ$_{ij}$、ｑ’$_{ij}$の送信を要求してから所定期間を経過してもｘ$_{ij}$、ｘ’$_{ij}$及び分散値ｑ$_{ij}$、ｑ’$_{ij}$が受信されなかった場合等では、ステップＳ９４の判定が否定判定となる。ステップＳ９４の判定が否定判定となった場合は、ステップＳ９６、Ｓ９８の処理は実行されずに本復元処理が終了する。

【０１２６】

以下に具体例を用いて説明する。例えば、ｎ＝ｋ＝２、ｔ＝１とした場合を考える。この場合、データサーバ１６はｎ－ｔ＝１台であり、このデータサーバ１６のサーバＩＤをｘ$_{2}$とし、攻撃者に乗っ取られているものとする。このｘ$_{2}$に対応するデータサーバ１６が偽の分散値Ｗ$_{2}$＋ΔＷ$_{2}$、Ｖ$_{2}$＋ΔＶ$_{2}$を出力する場合を考える。

【０１２７】

この場合、復元装置１４は、１回目には、多項式Ｗ（ｘ）により秘密情報の復元を行うことによって、以下の（２５）式で表されるＷ’（０）を復元する。

【０１２８】

【数１６】

$$W'(0)=s+\Delta W_2\frac{f_2(0)}{f_2(x_2)}\cdots(25)$$

【０１２９】

更に、この場合、復元装置１４は、２回目には、多項式Ｖ（ｘ）により秘密情報の復元を行うことによって、以下の（２６）式で表されるＶ’（０）を復元する。

【０１３０】

【数１７】

$$V'(0)=s+\Delta V_2\frac{g_2(0)}{g_2(x_2)}\cdots(26)$$

【０１３１】

Ｗ’（０）とＶ’（０）との差分は、ΔＷ$_{2}$とΔＶ$_{2}$とが以下の（２７）式の関係であれば０となるが、攻撃者はｘ$_{1}$、ｘ’$_{1}$を知ることはできないため、Ｗ’（０）とＶ’（０）との差分が０となるΔＷ$_{2}$とΔＶ$_{2}$とを設定することはできない。このため、Ｗ’（０）とＶ’（０）とは一致しない。従って、復元装置１４は、データサーバ１６から偽の

分散値が出力されたことを検証することができる。

【０１３２】

【数１８】

$$\Delta W_2 \frac{\left(-x_1\right)}{\left(x_2-x_1\right)} = \Delta V_2 \frac{\left(-x_1^{'}\right)}{\left(x_2-x_1^{'}\right)} \cdots (27)$$

【０１３３】

［第５実施形態］

開示の技術の第５実施形態を説明する。第３及び第４実施形態では、分散値の検証のためにＭＡＣ等の秘密分散とは異なる技術を用いずに、検証用の分散値を用いることによって、復元した秘密情報の正当性を検証する手法を説明した。この手法は、ＭＡＣ等を用いずに同じ形式の分散値を用いるため簡易であり、かつビット長等の形式を同じにすることができる（ＭＡＣはビット長が規定されている）。本実施形態では、第３及び第４実施形態と同様の特徴を持つ復号結果の検証法を暗号技術に適用した形態例を説明する。

【０１３４】

まず、図１２を参照して、本実施形態に係る通信システム４０の構成を説明する。図１２に示すように、通信システム４０は、送信装置４２及び受信装置４４を含む。送信装置４２及び受信装置４４は、ネットワークＮに接続され、互いに通信可能とされる。また、本実施形態では、送信装置４２及び受信装置４４は、同一の秘密鍵を記憶している。

【０１３５】

送信装置４２は、秘密情報を送信する送信者によって操作される装置であり、送信装置４２の例としては、パーソナルコンピュータ等の情報処理装置が挙げられる。受信装置４４は、秘密情報を受信する受信者によって操作される装置であり、受信装置４４の例としては、パーソナルコンピュータ等の情報処理装置が挙げられる。

【０１３６】

次に、図１３を参照して、本実施形態に係る送信装置４２のハードウェア構成を説明する。図１３に示すように、送信装置４２は、ＣＰＵ５０、一時記憶領域としてのメモリ５１、不揮発性の記憶部５２を含む。また、送信装置４２は、液晶ディスプレイ等の表示装置５３、キーボード等の入力装置５４、及びネットワークＮに接続されるネットワークＩ／Ｆ５５を含む。ＣＰＵ５０、メモリ５１、記憶部５２、表示装置５３、入力装置５４、及びネットワークＩ／Ｆ５５は、バス５６に接続される。

【０１３７】

記憶部５２は、ＨＤＤ、ＳＳＤ、及びフラッシュメモリ等によって実現される。記憶媒体としての記憶部５２には、送信プログラム５８が記憶される。ＣＰＵ５０は、記憶部５２から送信プログラム５８を読み出してからメモリ５１に展開し、展開した送信プログラム５８を実行する。ＣＰＵ５０が送信プログラム５８を実行することで、開示の技術の生成部及び送信部として動作する。

【０１３８】

次に、図１４を参照して、本実施形態に係る受信装置４４のハードウェア構成を説明する。図１４に示すように、受信装置４４は、ＣＰＵ６０、一時記憶領域としてのメモリ６１、不揮発性の記憶部６２を含む。また、受信装置４４は、液晶ディスプレイ等の表示装置６３、キーボード等の入力装置６４、及びネットワークＮに接続されるネットワークＩ／Ｆ６５を含む。ＣＰＵ６０、メモリ６１、記憶部６２、表示装置６３、入力装置６４、及びネットワークＩ／Ｆ６５は、バス６６に接続される。

【０１３９】

記憶部６２は、ＨＤＤ、ＳＳＤ、及びフラッシュメモリ等によって実現される。記憶媒体としての記憶部６２には、受信プログラム６８が記憶される。ＣＰＵ６０は、記憶部６２から受信プログラム６８を読み出してからメモリ６１に展開し、展開した受信プログラ

ム６８を実行する。ＣＰＵ７０が受信プログラム６８を実行することで、開示の技術の受信部及び検証部として動作する。

【０１４０】

次に、図１５及び図１６を参照して、本実施形態に係る通信システム４０の作用を説明する。まず、図１５を参照して、秘密情報を送信する送信処理を説明する。送信装置４２のＣＰＵ５０が送信プログラム５８を実行することで、図１５に示す送信処理を実行する。

【０１４１】

図１５のステップＳ１００で、ＣＰＵ５０は、秘密鍵を用いて、第１の乱数a_1及び第２の乱数α_1を生成する。ステップＳ１０２で、ＣＰＵ５０は、秘密情報ａに対して、ステップＳ１００で生成した第１の乱数a_1及び第２の乱数α_1を作用させて第１の送信データを生成する。具体的には、ＣＰＵ５０は、秘密情報ａに第１の乱数a_1を加算して得られた結果に、第２の乱数α_1を乗算することによって暗号化した第１の送信データ（$\alpha_1(a+a_1)$）を生成する。ステップＳ１０４で、ＣＰＵ５０は、ステップＳ１０２で生成した第１の送信データを受信装置４４に送信する。

【０１４２】

ステップＳ１０６で、ＣＰＵ５０は、秘密鍵を用いて、第３の乱数a_2及び第４の乱数α_2を生成する。ステップＳ１０８で、ＣＰＵ５０は、秘密情報ａに対して、ステップＳ１０２における第１の乱数a_1に代えた第３の乱数a_2及び第２の乱数α_1に代えた第４の乱数α_2を同様に作用させて第２の送信データを生成する。具体的には、ＣＰＵ５０は、秘密情報ａに第３の乱数a_2を加算して得られた結果に、第４の乱数α_2を乗算することによって暗号化した第２の送信データ（$\alpha_2(a+a_2)$）を生成する。ステップＳ１１０で、ＣＰＵ５０は、ステップＳ１０８で生成した第２の送信データを、第１の送信データの送信先と同一の受信装置４４に送信する。ステップＳ１１０の処理が終了すると、本送信処理が終了する。

【０１４３】

次に、図１６を参照して、図１５に示す送信処理によって送信された情報を受信する受信処理を説明する。受信装置４４のＣＰＵ６０が受信プログラム６８を実行することで、図１６に示す受信処理を実行する。

【０１４４】

図１６のステップＳ１２０で、ＣＰＵ６０は、送信処理のステップＳ１０４で送信装置４２から送信された第１の送信データを受信する。ステップＳ１２２で、ＣＰＵ６０は、送信装置４２が記憶している秘密鍵と同一の秘密鍵を用いて、第１の乱数a_1及び第２の乱数α_1を生成する。

【０１４５】

ステップＳ１２４で、ＣＰＵ６０は、ステップＳ１２０で受信した第１の送信データから、ステップＳ１２２で生成した第２の乱数α_1を排除する。具体的には、ＣＰＵ６０は、第１の送信データを第２の乱数α_1で除算する。すなわち、第１の送信データから第２の乱数α_1を排除した結果は、$\alpha_1(a+a_1)/\alpha_1$で表される。

【０１４６】

ステップＳ１２６で、ＣＰＵ６０は、送信処理のステップＳ１１０で送信装置４２から送信された第２の送信データを受信する。ステップＳ１２８で、ＣＰＵ６０は、送信装置４２が記憶している秘密鍵と同一の秘密鍵を用いて、第３の乱数a_2及び第４の乱数α_2を生成する。

【０１４７】

ステップＳ１３０で、ＣＰＵ６０は、ステップＳ１２６で受信した第２の送信データから、ステップＳ１２８で生成した第４の乱数α_2を排除する。具体的には、ＣＰＵ６０は、第２の送信データを第４の乱数α_2で除算する。すなわち、第２の送信データから第４の乱数α_2を排除した結果は、$\alpha_2(a+a_2)/\alpha_2$で表される。

【０１４８】

ステップＳ１３２で、ＣＰＵ６０は、ステップＳ１２４による排除結果と、ステップＳ１３０による排除結果との差分を算出する。この差分は、以下に示す（２８）式で表される。

$$\alpha_1(a+a_1)/\alpha_1-\alpha_2(a+a_2)/\alpha_2 \cdots (28)$$

【０１４９】

そして、ＣＰＵ６０は、算出した差分が、ステップＳ１２２で生成した第１の乱数a_1と、ステップＳ１２８で生成した第３の乱数a_2との差分（a_1-a_2）に等しいか否かを検証する。具体的には、ＣＰＵ６０は、（２８）式で算出した差分が、第１の乱数a_1と第３の乱数a_2との差分に等しい場合、受信したデータが正しいと判断し、秘密情報ａを復号する。一方、ＣＰＵ６０は、（２８）式で算出した差分が、第１の乱数a_1と第３の乱数a_2との差分とは異なる場合、受信したデータに改ざんや偽造等の不正があると判断し、例えば、エラーメッセージを表示装置６３に表示する。ステップＳ１３２の処理が終了すると、本受信処理が終了する。

【０１５０】

本実施形態において、攻撃者が、偽の第１の送信データとして$\alpha_1(a+a_1)+\Delta_1$を受信装置４４に受信させ、偽の第２の送信データとして$\alpha_2(a+a_2)+\Delta_2$を受信装置４４に受信させることができる。しかしながら、攻撃者はa_1、α_1、a_2、α_2を知ることはできないため、（２８）式で表される差分とa_1-a_2とが等しくなるような偽の送信データを生成することはできない。従って、受信装置４４は、受信したデータの正当性を検証することができる。

【０１５１】

なお、上記各実施形態でＣＰＵがソフトウェア（プログラム）を実行することにより実行した各種処理を、ＣＰＵ以外の各種のプロセッサが実行してもよい。この場合のプロセッサとしては、ＦＰＧＡ（Field-Programmable Gate Array）等の製造後に回路構成を変更可能なＰＬＤ（Programmable Logic Device）、及びＡＳＩＣ（Application Specific Integrated Circuit）等の特定の処理を実行させるために専用に設計された回路構成を有するプロセッサである専用電気回路等が例示される。また、各種処理を、これらの各種のプロセッサのうちの１つで実行してもよいし、同種又は異種の２つ以上のプロセッサの組み合わせ（例えば、複数のＦＰＧＡ、及びＣＰＵとＦＰＧＡとの組み合わせ等）で実行してもよい。また、これらの各種のプロセッサのハードウェア的な構造は、より具体的には、半導体素子等の回路素子を組み合わせた電気回路である。

【０１５２】

また、上記第１～第４実施形態では、生成プログラム２８が記憶部２２に予め記憶（インストール）されている態様を説明したが、これに限定されない。生成プログラム２８は、ＣＤ－ＲＯＭ（Compact Disc Read Only Memory）、ＤＶＤ－ＲＯＭ（Digital Versatile Disc Read Only Memory）、及びＵＳＢ（Universal Serial Bus）メモリ等の記録媒体に記録された形態で提供されてもよい。また、生成プログラム２８は、ネットワークを介して外部装置からダウンロードされる形態としてもよい。

【０１５３】

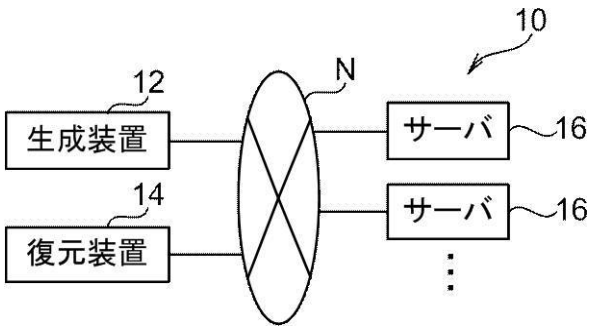
また、上記第１～第４実施形態では、復元プログラム３８が記憶部３２に予め記憶（インストール）されている態様を説明したが、これに限定されない。復元プログラム３８は、ＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、及びＵＳＢメモリ等の記録媒体に記録された形態で提供されてもよい。また、復元プログラム３８は、ネットワークを介して外部装置からダウンロードされる形態としてもよい。

【０１５４】

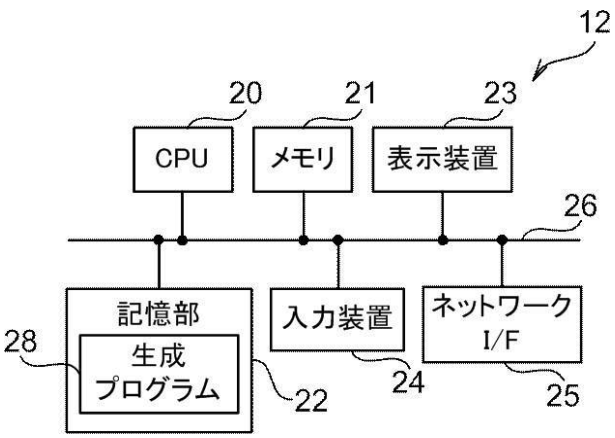
また、上記第５実施形態では、送信プログラム５８が記憶部５２に予め記憶（インストール）されている態様を説明したが、これに限定されない。送信プログラム５８は、ＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、及びＵＳＢメモリ等の記録媒体に記録された形態で提供されてもよい。また、送信プログラム５８は、ネットワークを介して外部装置からダウンロードされる形態としてもよい。

【０１５５】
　また、上記第５実施形態では、受信プログラム６８が記憶部６２に予め記憶（インストール）されている態様を説明したが、これに限定されない。受信プログラム６８は、ＣＤ－ＲＯＭ、ＤＶＤ－ＲＯＭ、及びＵＳＢメモリ等の記録媒体に記録された形態で提供されてもよい。また、受信プログラム６８は、ネットワークを介して外部装置からダウンロードされる形態としてもよい。
【符号の説明】
【０１５６】
１０　秘密分散システム
１２　生成装置
１４　復元装置
１６　サーバ
２０、３０、５０、６０　ＣＰＵ
２１、３１、５１、６１　メモリ
２２、３２、５２、６２　記憶部
２８　生成プログラム
３８　復元プログラム
４０　通信システム
４２　送信装置
４４　受信装置
５８　送信プログラム
６８　受信プログラム

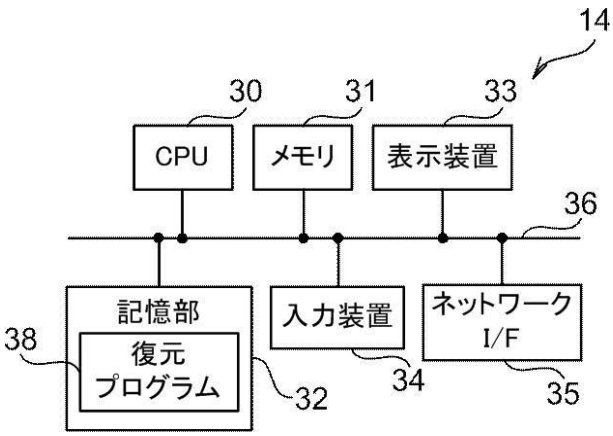
【図１】



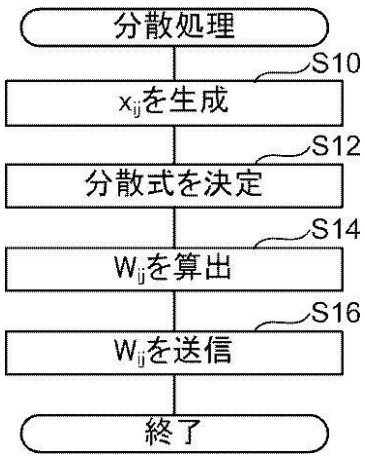
【図２】



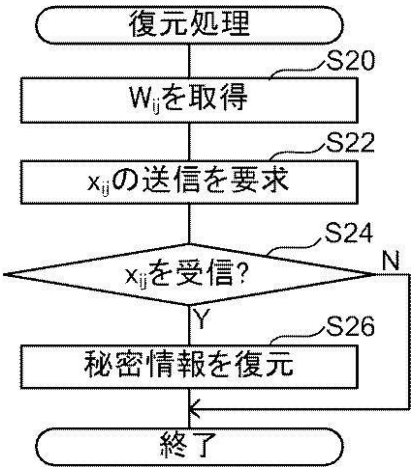
【図３】



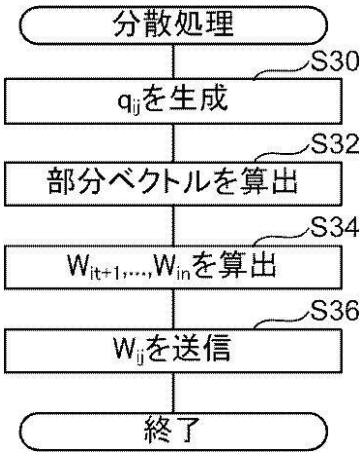
【図４】



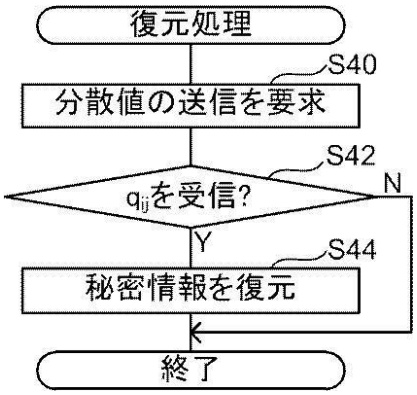
【図５】


復元処理
S20
W_{ij}を取得
S22
x_{ij}の送信を要求
S24
x_{ij}を受信?
N
Y
S26
秘密情報を復元
終了

【図６】


分散処理
S30
q_{ij}を生成
S32
部分ベクトルを算出
S34
$W_{it+1},...,W_{in}$を算出
S36
W_{ij}を送信
終了

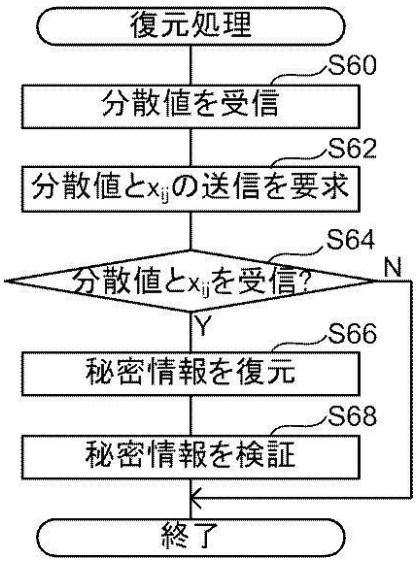
【図７】


復元処理
S40
分散値の送信を要求
S42
q_{ij}を受信?
N
Y
S44
秘密情報を復元
終了

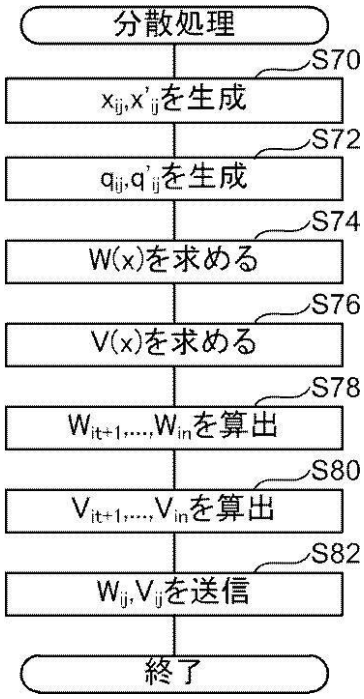
【図８】


分散処理
S50
x_{ij}を生成
S52
q_{ij}を生成
S54
部分ベクトルを算出
S56
$W_{it+1},...,W_{in}$を算出
S58
W_{ij}を送信
終了

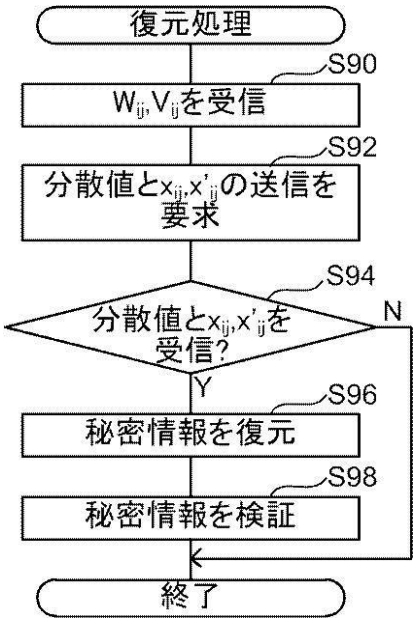
【図９】


復元処理
S60
分散値を受信
S62
分散値とx_{ij}の送信を要求
S64
分散値とx_{ij}を受信?
N
Y
S66
秘密情報を復元
S68
秘密情報を検証
終了

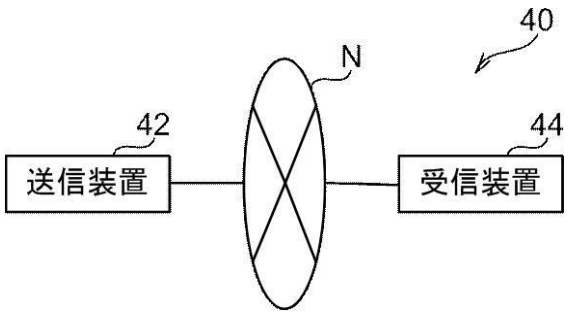
【図１０】



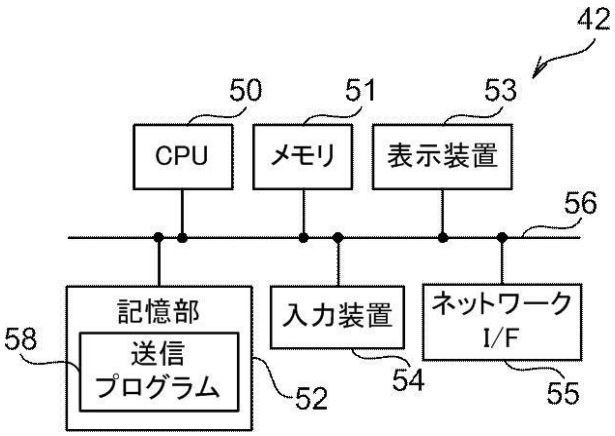
【図１１】



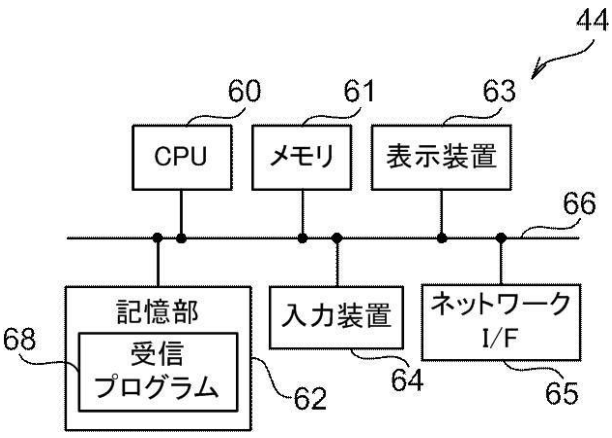
【図１２】



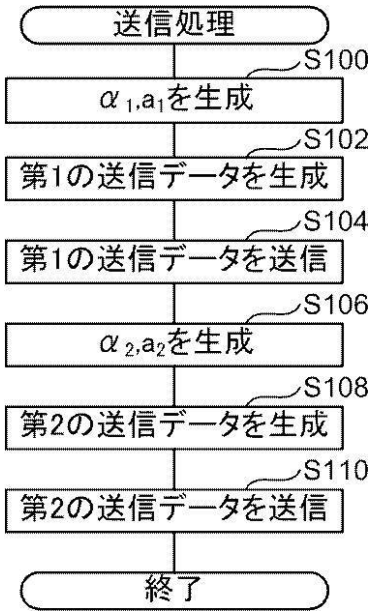
【図１３】



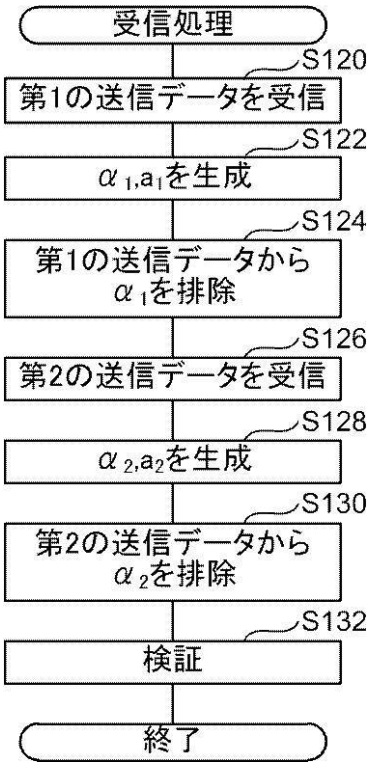
【図１４】



【図１５】


送信処理
S100
α_1,a_1を生成
S102
第1の送信データを生成
S104
第1の送信データを送信
S106
α_2,a_2を生成
S108
第2の送信データを生成
S110
第2の送信データを送信
終了

【図１６】


受信処理
S120
第1の送信データを受信
S122
α_1,a_1を生成
S124
第1の送信データから
α_1を排除
S126
第2の送信データを受信
S128
α_2,a_2を生成
S130
第2の送信データから
α_2を排除
S132
検証
終了