(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2009 年5 月7 日 (07.05.2009)

PCT

(10) 国際公開番号
WO 2009/057338 A1

(51) 国際特許分類:
G09C 1/00 (2006.01) *H04L 9/32* (2006.01)

(21) 国際出願番号: PCT/JP2008/057962

(22) 国際出願日: 2008 年4 月24 日 (24.04.2008)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願 2007-280287
2007 年10 月29 日 (29.10.2007) JP

(71) 出願人 (米国を除く全ての指定国について): 日本電信電話株式会社 (NIPPON TELEGRAPH AND TELEPHONE CORPORATION) [JP/JP]; 〒1008116 東京都千代田区大手町二丁目３番１号 Tokyo (JP).

(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 鈴木 幸太郎 (SUZUKI, Koutarou) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目９番１１号 ＮＴＴ 知的財産センタ内 Tokyo (JP). 阿部 正幸 (ABE, Masayuki) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目９番１１号 ＮＴＴ 知的財産センタ内 Tokyo (JP). 岡本 龍明 (OKAMOTO, Tatsuaki) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目９番１１号 ＮＴＴ 知的財産センタ内 Tokyo (JP). 藤岡 淳 (FUJIOKA, Atsushi) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目９番１１号 ＮＴＴ 知的財産センタ内 Tokyo (JP). 山本 剛 (YAMAMOTO, Go) [JP/JP]; 〒1808585 東京都武蔵野市緑町三丁目９番１１号 ＮＴＴ 知的財産センタ内 Tokyo (JP).

(74) 代理人: 中尾 直樹, 外 (NAKAO, Naoki et al.); 〒1600022 東京都新宿区新宿三丁目１番２２号 新宿ＮＳＯビル４階 Tokyo (JP).

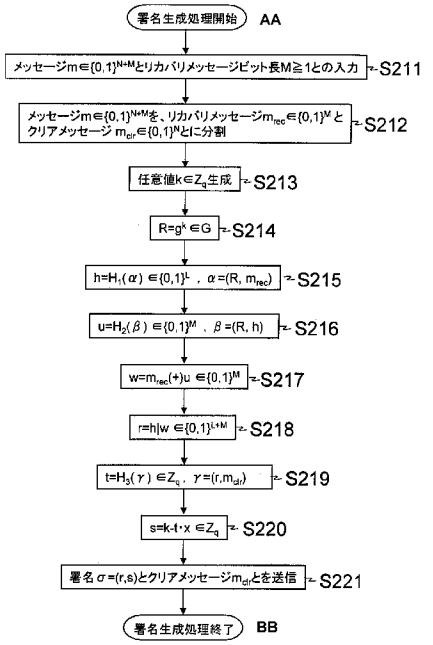
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG,

/続葉有/

(54) Title: SIGNATURE GENERATING DEVICE, SIGNATURE VERIFYING DEVICE, THEIR METHODS, AND THEIR PROGRAMS

(54) 発明の名称: 署名生成装置、署名検証装置、それらの方法及びプログラム

[図16]



AA START SIGNATURE GENERATION

S211 INPUT MESSAGE $m \in \{0,1\}^{N+M}$ AND RECOVERY MESSAGE BIT LENGTH $M \geq 1$

S212 DIVIDE MESSAGE $m \in \{0,1\}^{N+M}$ INTO RECOVERY MESSAGE $m_{rec} \in \{0,1\}^{M}$ AND CLEAR MESSAGE $m_{cir} \in \{0,1\}^{N}$

S213 GENERATE GIVEN VALUE $k \in Z_q$

S221 TRANSMIT SIGNATURE $\sigma = (r,s)$ AND CLEAR MASSAGE m_{cir}

BB END SIGNATURE GENERATION

(57) Abstract: A signature is generated by a scheme in which x denotes a secret key of a signature generating device, a recovery message is defined as $m_{rec} \in \{0,1\}^M$, k denotes a given value, g^k denotes a generator of a cyclic group G of an order q, R denotes $g^k \in G$, H_1 denotes a hash function H_1: $\{0,1\}^* \rightarrow \{0,1\}^L$, H_2 denotes a hash function H_2: $\{0,1\}^* \rightarrow \{0,1\}^M$ having an output variable length, H_3 denotes a hash function H_3: $\{0,1\}^* \rightarrow Z_q$, r is defined as $r = H_1(R,m_{rec})|m_{rec}(+)H_2(R,H_1(R,m_{rec}))$ (where (+) represents an exclusive OR operator), t is defined as $t = H_3(\gamma)$ where γ depends on r, s is defined as $s = k-t \cdot x \in Z$, and a signature is $\sigma = (r,s)$.

(57) 要約: ｘを署名生成装置の秘密鍵とし、$m_{rec} \in \{0,1\}^M$をリカバリメッセージとし、ｋを任意値とし、ｇを位数ｑの巡回群Ｇの生成元とし、Ｒを$g^k \in G$とし、Ｈ１をハッシュ関数H_1:$\{0,1\}^* \rightarrow \{0,1\}^L$とし、Ｈ２を出力可変長のハッシュ関数$H_2$:$\{0,1\}^* \rightarrow \{0,1\}^M$とし、Ｈ３をハッシュ関数$H_3$:$\{0,1\}^* \rightarrow Z_q$とし、$r=H_1(R,m_{rec})|m_{rec}(+)H_2(R,H_1(R,m_{rec}))$とし（(+)は排他的論理和演算子）、ｒに依存するγに対して$t=H_3(\gamma)$とし、$s=k-t \cdot x \in Z$とし、署名を$\sigma=(r,s)$とする。

WO 2009/057338 A1

BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), **ユーラシア** (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), **ヨーロッパ** (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— **国際調査報告書**

明 細 書

署名生成装置、署名検証装置、それらの方法及びプログラム

技術分野

[0001] 本発明は、情報セキュリティ技術の応用技術に関し、特に、署名からメッセージが復元できるメッセージ復元署名に関する。

背景技術

[0002] メッセージ復元署名の従来技術として非特許文献1に示すものがある。この方式は、ランダムオラクルモデルで安全性が証明される方式である。以下にこの方式の概要を示す。

この方式では、

メッセージ$m \in \{0, 1\}^{k2}$

関数$F_1 : \{0, 1\}^{k2} \rightarrow \{0, 1\}^{k1}$

関数$F_2 : \{0, 1\}^{k1} \rightarrow \{0, 1\}^{k2}$

関数$H : \{0, 1\}^{k1+k2} \rightarrow \{0, 1\}^{k}$

E：有限体F_q上で定義された楕円曲線

p：楕円曲線E上の点Rに対して$p \cdot R = O$（Oは無限遠点）を満たす素数

G1：楕円曲線E上の位数pの部分集合の点

$w \in Z / pZ$

秘密鍵：$x \in Z / pZ$

公開鍵：$(F_q, E, G1, Y)$ $(Y = -x \cdot G1 (\in E))$

とする。なお、$\{0, 1\}^{\delta}$は、2進δ桁のビットデータを示し、$\{0, 1\}^{\delta} \rightarrow \{0, 1\}^{\varepsilon}$は、2進δ桁のビットデータから2進ε桁のビットデータへの写像である関数を示す。

[0003] ＜署名生成＞

署名生成は以下のように行う。ただし、R_xは点$R \in E$のx座標を示し、（＋）は排他的論理和演算子を示す。

$$m' = F_1(m) \mid (F_2(F_1(m)) (+) m) \quad \cdots (1)$$

$$R_x = (w \cdot G1)_x$$

r=R(+)m' …(2)

c=H(r)

z=w+c・x mod p

署名 σ =(r,z)

[0004] ＜署名検証＞

署名検証は以下のように行う。ただし、$[m']^{k1}$はm'の先頭k1ビットを示し、$[m']^{k2}$はm'の残りのk2ビットを示す。

$m'=r(+)(z\cdot G1+H(r)\cdot Y)_x$

$m=[m']^{k2}(+)F_2([m']^{k1})$

$[m']^{k1}=F_1(m)$であれば検証合格

非特許文献1:Masayuki Abe, Tatsuaki Okamoto, "A Signature Scheme with Message Recovery as Secure as Discrete Logarithm," ASIACRYPT 1999, pp.378-389

発明の開示

発明が解決しようとする課題

[0005] しかし、非特許文献1の方式では、式(1)の$(F_2(F_1(m))$や式(2)のR_xのビット長が固定長であり、メッセージmのビット長も固定長としなければならない。

このため、メッセージmの長さが固定長より短い場合であっても、それに併せて署名σの一部分rのビット長を短くすることができず、非効率である。また、メッセージmのビット長が固定長よりも長い場合には、式(1)にメッセージmの一部分しか代入することができず、メッセージmの全てのビットを対象としたメッセージ復元署名を構成できない。

課題を解決するための手段

[0006] 本発明の署名生成装置は以下のように署名生成を行う。

まず、署名生成装置の記憶部に整数の秘密鍵xが格納され、さらに、Mビットのリカバリメッセージ$m_{rec}\in\{0,1\}^M$が格納される。ここで、リカバリメッセージm_{rec}が、署名対象の少なくとも一部となる。そして、署名生成装置が、整数の任意値kを生成し、位数qの巡回群をGとし、当該巡回群Gの生成元をgとした場合における$R=g^k\in G$を算出し、当該演算結果Rを出力する。なお、「$g^k\in G$」とは、巡回群Gをなす演算をgに

ついてk回実行することを意味する(詳細は後述)。次に、署名生成装置が、入力値に対してLビットのハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^*\rightarrow\{0,1\}^L$を、演算結果Rとリカバリメッセージ$m_{rec}$とに依存する値αに作用させ、その演算結果であるLビットのハッシュ値$h=H_1(\alpha)\in\{0,1\}^L$を出力する。なお、Lは署名検証装置と共有される正の整数である。また、「関数εをδに作用させる」とは、「δ又はδを特定するための値を関数εに代入する」ことを意味する。次に、署名生成装置が、リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がMビットに定まるハッシュ関数$H_2:\{0,1\}^*\rightarrow\{0,1\}^M$を、演算結果Rとハッシュ値hとに依存する値βに作用させ、その演算結果であるMビットのハッシュ値$u=H_2(\beta)\in\{0,1\}^M$を出力する。また、署名生成装置が、リカバリメッセージm_{rec}とハッシュ値uとの排他的論理和$w=m_{rec}(+)u\in\{0,1\}^M$((+)は排他的論理和演算子)を算出し、当該排他的論理和値wを出力する。さらに、署名生成装置が、ハッシュ値$h\in\{0,1\}^L$を第1ビット位置に配置し、排他的論理和値$w\in\{0,1\}^M$を第2ビット位置に配置したL+Mビットのビット結合値$h\mid w\in\{0,1\}^{L+M}$に依存し、ハッシュ値h及び排他的論理和値wを復元可能な値rを算出し、当該値rを出力する。なお、第1ビット位置は必ずしも連続したLビットの位置である必要はなく、離散的に配置された合計Lビットの位置でもよい。同様に、第2ビット位置も必ずしも連続したMビットの位置である必要はなく、離散的に配置された合計Mビットの位置でもよい。ただし、「第1ビット位置」及び「第2ビット位置」がどのビット位置であるかについては、署名生成装置と署名検証装置とで統一しておく。次に、署名生成装置が、入力値に対して整数を出力するハッシュ関数$H_3:\{0,1\}^*\rightarrow Z$(整数)を、値rに依存する値γに作用させ、その演算結果であるハッシュ値$t=H_3(\gamma)\in Z$を出力する。そして、署名生成装置が、$s=k-t\cdot x\in Z$を算出し、署名$\sigma=(r,s)$を出力する。

[0007] 本発明の署名検証装置は以下のように署名検証を行う。なお、署名検証装置が受け取る署名を$\sigma'=(r',s')$と表現する。また、署名検証装置の記憶部には署名生成装置の公開鍵$y=g^x\in G$が格納されている。

[0008] まず、署名検証装置に署名$\sigma'=(r',s')$が入力され、その署名$\sigma'=(r',s')$が記憶部に格納される。また、署名σ'に対応するリカバリメッセージm_{rec}'のビット長M'

が記憶部に格納される。なお、署名検証装置がビット長M'の値を取得する方法については後述する。そして、署名検証装置が、入力値に対して整数を出力するハッシュ関数$H_3:\{0,1\}^*\rightarrow Z$(整数)を、署名σ'が有するr'に依存する値γ'に作用させ、その演算結果であるハッシュ値$t'=H_3(\gamma')\in Z$を出力する。また、署名検証装置が$R'=g^{s'}\cdot y^{t'}\in G$の演算を行い、その演算結果R'を出力する。なお、「$g^{s'}\cdot y^{t'}\in G$」とは、巡回群Gをなす演算をgについてs'回施し、当該演算をyについてt'回施し、それらの各演算結果に対して当該演算を施す演算を意味する(詳細は後述)。次に、署名検証装置が、リカバリメッセージm_{rec}'のビット長M'に応じて出力ビット長がM'ビットに定まるハッシュ関数$H_2:\{0,1\}^*\rightarrow\{0,1\}^{M'}$を、演算結果R'とr'の第1ビット位置のLビットの値$h'\in\{0,1\}^L$とに依存する値β'に作用させ、その演算結果であるM'ビットのハッシュ値$u'=H_2(\beta')\in\{0,1\}^{M'}$を出力する。また、署名検証装置が、r'の第2ビット位置のM'ビットの値に依存する値$w'\in\{0,1\}^{M'}$とハッシュ値u'との排他的論理和w'(+)u'を算出し、その演算結果をリカバリメッセージ$m_{rec}'\in\{0,1\}^{M'}$として出力する。さらに、署名検証装置が、入力値に対してLビットのハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^*\rightarrow\{0,1\}^L$を、演算結果R'と算出されたリカバリメッセージ$m_{rec}'$とに依存する値α'に作用させ、その演算結果であるLビットのハッシュ値$H_1(\alpha')\in\{0,1\}^L$を出力する。そして、署名検証装置が、Lビットの値h'とハッシュ値$H_1(\alpha')$とを比較し、$h'=H_1(\alpha')$であることを条件に、検証が成功である旨の情報を出力する。なお、「εとδとに依存する値」には、この値がεとδとのみに依存する場合のみならず、εとδと他の情報とに依存する場合も含まれる。また、「εに依存する値」には、この値がεのみに依存する場合のみならず、εと他の情報とに依存する場合も含まれる。ただし、署名生成装置で用いる値α, β, γの構成方法と、署名検証装置で用いる値α', β', γ'の構成方法とが、それぞれ同一である必要がある(詳細は後述)。

[0009] ここで本発明では、リカバリメッセージのビット長に従って出力ビット長が変化するハッシュ関数を用い、処理方法を工夫することにより、リカバリメッセージのビット長が変化しても、各排他的論理和演算の対象となる2つの被演算子のビット長を常に同一にできる。これにより、リカバリメッセージのビット長が短い場合には、それに併せて各演

算過程での演算ビット数や署名σのビット数を短くできる。また、リカバリメッセージのビット長が長くなっても、リカバリメッセージm_{rec}の全てのビットを対象としたメッセージ復元署名が生成できる。

[0010] また、本発明では、署名生成装置で算出されたハッシュ値h，uと署名検証装置で算出されたハッシュ値h'，u'とがそれぞれ整合性を持たないと署名検証が成功とならないため、ハッシュ値hとハッシュ値h'との整合性のみで署名検証を行う場合より高い安全性を保証できる。

なお、本発明では、従来と異なり、メッセージの全てのビットをメッセージ復元署名の対象としてもよい（$m=m_{rec}$）。

[0011] また、メッセージmの全てのビットをメッセージ復元署名の対象としなくてもよい。メッセージmの全てのビットをメッセージ復元署名の対象としない場合、Mビットのリカバリメッセージm_{rec}がメッセージ復元署名の対象となり、Nビットのクリアメッセージm_{clr}がメッセージ復元署名ではない通常の署名対象となる。この場合好ましくは、さらに、署名生成装置が、Nビットのクリアメッセージ$m_{clr}\in\{0,1\}^N$を記憶部に格納し、ハッシュ関数$H_3:\{0,1\}^*\rightarrow Z$を、値rとクリアメッセージ$m_{clr}$とに依存する値γに作用させて$t=H_3(\gamma)\in Z$を算出し、$s=k-t\cdot x\in Z$を算出して署名$\sigma=(r,s)$とクリアメッセージ$m_{clr}$とを出力する。署名検証装置には署名σ'とクリアメッセージm_{clr}'とが入力される。署名検証装置は、ハッシュ関数$H_3:\{0,1\}^*\rightarrow Z$を、署名σ'が有するr'とクリアメッセージ$m_{clr}$'とに依存する値γ'に作用させ、その演算結果であるハッシュ値$t'=H_3(\gamma')\in Z$を出力する。

これにより、メッセージの全てのビットをメッセージ復元署名の対象とする必要のない場合にまで、メッセージの全ビットをメッセージ復元署名の対象とし、各演算過程における演算ビット数が長くなってしまうことを回避できる。即ち、様々なビット長のメッセージ及び様々な用途に対し、柔軟に対応可能なメッセージ復元署名が実現できる。

発明の効果

[0012] 本発明では、様々なビット長のメッセージに柔軟に対応可能なメッセージ復元署名を実現できる。

図面の簡単な説明

[0013] [図1]図1は、第1実施形態の署名システムの全体構成を示した概念図である。

[図2]図2は、第1実施形態における署名生成装置のハードウェア構成を例示したブロック図である。

[図3]図3は、第1実施形態における署名生成装置の機能構成を例示したブロック図である。

[図4]図4Aは、ハッシュ演算部の機能構成の詳細を示した図であり、図4Bは、ハッシュ演算部の機能構成の詳細を示した図である。

[図5]図5は、第1実施形態の署名検証装置の機能構成を例示したブロック図である。

[図6]図6は、第1実施形態の署名生成処理を説明するためのフローチャートである。

[図7]図7Aは、ステップS15の処理の例を説明するためのフローチャートであり、図7Bは、ステップS17の処理の例を説明するためのフローチャートである。

[図8]図8は、第1実施形態の署名検証処理を説明するためのフローチャートである。

[図9]図9Aは「第1ビット位置」及び「第2ビット位置」の設定例を示した図であり、図9Bは「第1ビット位置」及び「第2ビット位置」の設定例を示した図であり、図9Cは「第1ビット位置」及び「第2ビット位置」の設定例を示した図である。

[図10]図10は、第2実施形態における署名生成装置の機能構成を例示したブロック図である。

[図11]図11は、第2実施形態の署名検証装置の機能構成を例示したブロック図である。

[図12]図12は、第2実施形態の署名生成処理を説明するためのフローチャートである。

[図13]図13は、第2実施形態の署名検証処理を説明するためのフローチャートである。

[図14]図14は、第3実施形態における署名生成装置の機能構成を例示したブロック図である。

[図15]図15は、第3実施形態の署名検証装置の機能構成を例示したブロック図である。

[図16]図16は、第3実施形態の署名生成処理を説明するためのフローチャートである。

[図17]図17は、第3実施形態の署名検証処理を説明するためのフローチャートである。

[図18]図18は、第4実施形態における署名生成装置の機能構成を例示したブロック図である。

[図19]図19は、第4実施形態の署名検証装置の機能構成を例示したブロック図である。

[図20]図20は、第4実施形態の署名生成処理を説明するためのフローチャートである。

[図21]図21は、第4実施形態の署名検証処理を説明するためのフローチャートである。

符号の説明

[0014] 1 署名システム

10, 110, 210, 310 署名生成装置

20, 120, 220, 320 署名検証装置

発明を実施するための最良の形態

[0015] 以下、本発明を実施するための最良の形態を図面を参照して説明する。

〔第1実施形態〕

まず、本発明の第1実施形態について説明する。

＜全体構成＞

図1は、第1実施形態の署名システム1の全体構成を示した概念図である。

図1に示すように、本形態の署名システム1は、署名生成を行う署名生成装置10と、署名検証を行う署名検証装置20と、署名生成装置10の効果鍵を公開する公開鍵サーバ装置30とを有し、相互にネットワーク40を通じて通信可能に接続される。なお、署名生成装置10、署名検証装置20及び公開鍵サーバ装置30は、それぞれ、公知のコンピュータに所定のプログラムが読み込まれることにより構成される装置である。

[0016] ＜署名生成装置10の構成＞

次に、署名生成装置10の構成を説明する。

［ハードウェア構成］

図2は、第1実施形態における署名生成装置10のハードウェア構成を例示したブロック図である。

図2に例示するように、この例の署名生成装置10は、CPU（Central Processing Unit）11、入力部12、出力部13、補助記憶装置14、ROM（Read Only Memory）15、RAM（Random Access Memory）16、バス17及び通信部18を有している。

[0017] この例のCPU11は、制御部11a、演算部11b及びレジスタ11cを有し、レジスタ11cに読み込まれた各種プログラムに従って様々な演算処理を実行する。また、この例の入力部12は、データが入力される入力ポート、キーボード、マウス等であり、出力部13は、データを出力する出力ポート、外部記録媒体へのデータ記憶装置、印刷装置、ディスプレイなどである。補助記憶装置14は、例えば、ハードディスク、MO（Magneto-Optical disc）、半導体メモリ等であり、各種プログラムを格納したプログラム領域14a及び各種データが格納されるデータ領域14bを有している。また、RAM16は、例えば、SRAM (Static Random Access Memory)、DRAM (Dynamic Random Access Memory)等であり、上記のプログラムが書き込まれるプログラム領域16a及び各種データが書き込まれるデータ領域16bを有している。また、通信部18は、ネットワークカードなどである。また、この例のバス17は、CPU11、入力部12、出力部13、補助記憶装置14、ROM15、RAM16及び通信部18を、データのやり取りが可能なように接続する。

[0018] ［ハードウェアとプログラムとの協働］

CPU11（図2）は、読み込まれたOS（Operating System）プログラムに従い、補助記憶装置14のプログラム領域14aに格納されているプログラムをRAM16のプログラム領域16aに書き込む。同様にCPU11は、補助記憶装置14のデータ領域14bに格納されている各種データを、RAM16のデータ領域16bに書き込む。そして、このプログラムやデータが書き込まれたRAM16上のアドレスがCPU11のレジスタ11cに格納される。CPU11の制御部11aは、レジスタ11cに格納されたこれらのアドレスを

順次読み出し、読み出したアドレスが示すRAM16上の領域からプログラムやデータを読み出し、そのプログラムが示す演算を演算部11bに順次実行させ、その演算結果をレジスタ11cに格納していく。なお、各プログラムは、単一のプログラム列として記載されていてもよく、また、少なくとも一部のプログラムが別個のモジュールとしてライブラリに格納されていてもよい。

[0019] 図3は、CPU11にプログラムが読み込まれることにより構成される第1実施形態における署名生成装置10の機能構成を例示したブロック図である。なお、図3における矢印はデータの流れを示すが、一時メモリ10tや制御部10sに入出力されるデータの流れは省略してある。

図3に示すように、本形態の署名生成装置10は、記憶部10aと、秘密鍵生成部10bと、公開鍵生成部10cと、入力部10dと、メッセージ分割部10eと、任意値生成部10fと、群演算部10gと、ハッシュ演算部10h, 10i, 10j, 10pと、排他的論理和演算部10k, 10nと、ビット結合部10mと、整数演算部10qと、通信部10rと、制御部10sと、一時メモリ10tとを有する。なお、ビット結合部10mと排他的論理和演算部10nとはr値演算部10zを構成する。

[0020] また、図4Aは、ハッシュ演算部10hの機能構成の詳細を示した図であり、図4Bは、ハッシュ演算部10jの機能構成の詳細を示した図である。図4に示すように、ハッシュ演算部10hは、ハッシュ回数演算部10haと、部分ハッシュ演算部10hbと、ビット結合部10hcと、ビット削除部10hdとを有する。また、ハッシュ演算部10jは、ハッシュ回数演算部10jaと、部分ハッシュ演算部10jbと、ビット結合部10jcと、ビット削除部10jdとを有する。

[0021] なお、記憶部10aと一時メモリ10tとは、例えば、図2に記載したレジスタ11c、補助記憶装置14、RAM16、或いはこれらを結合した記憶領域に相当する。また、秘密鍵生成部10bと、公開鍵生成部10cと、メッセージ分割部10eと、任意値生成部10fと、群演算部10gと、ハッシュ演算部10h, 10i, 10j, 10pと、排他的論理和演算部10k, 10nと、ビット結合部10mと、整数演算部10qと、制御部10sとは、それぞれの処理を実現するためのプログラムがCPU11に読み込まれることにより構成されるものである。また、入力部10dは、所定のプログラムが読み込まれたCPU11の制御のもと駆

動する入力部12であり、通信部10rは、所定のプログラムが読み込まれたCPU11の制御のもと駆動する通信部18である。また、署名生成装置10は、制御部10sの制御のもと各処理を実行する。さらに、特に明示しない限り、演算過程の各データは逐一一時メモリ10tに読み書きされる。

[0022] また、上記のプログラムは単体でその機能を実現できるものでもよいし、当該プログラムがさらに他のライブラリ(記載していない)を読み出して各機能を実現するものでもよい。すなわち、各プログラムの少なくとも一部が、署名生成装置10の機能をコンピュータに実行させるためのプログラムに相当する。

＜署名検証装置20の構成＞

次に、署名検証装置20の構成を説明する。

[ハードウェア構成]

図2に示した署名生成装置10のハードウェア構成と同様である。

[0023] [ハードウェアとプログラムとの協働]

署名検証装置20も図2に示したようなコンピュータに所定のプログラムが読み込まれることにより構成される。図5は、このように構成される第1実施形態の署名検証装置20の機能構成を例示したブロック図である。なお、図5における矢印はデータの流れを示すが、一時メモリ20nや制御部20pに入出力されるデータの流れは省略してある。

図5に示すように、本形態の署名検証装置20は、記憶部20aと、通信部20bと、ビット長抽出部20cと、ハッシュ演算部20d, 20f, 20i, 20kと、群演算部20eと、排他的論理和演算部20gと、ビット抽出部20hと、排他的論理和演算部20jと、比較部20lと、出力部20mと、制御部20nと、一時メモリ20pとを有する。

[0024] なお、記憶部20aと一時メモリ20pとは、例えば、コンピュータが具備するレジスタ、補助記憶装置、RAM、或いはこれらを結合した記憶領域に相当する。また、ビット長抽出部20cと、ハッシュ演算部20d, 20f, 20i, 20kと、群演算部20eと、排他的論理和演算部20gと、ビット抽出部20hと、排他的論理和演算部20jと、比較部20lと、制御部20nとは、それぞれの処理を実現するためのプログラムがCPUに読み込まれることにより構成されるものである。また、出力部20mと通信部20bは、所定のプログラ

ムが読み込まれたCPUの制御のもと駆動する。また、署名検証装置20は、制御部20nの制御のもと各処理を実行する。さらに、特に明示しない限り、演算過程の各データは逐一一時メモリ20pに読み書きされる。

[0025] また、上記のプログラムは単体でその機能を実現できるものでもよいし、当該プログラムがさらに他のライブラリ(記載していない)を読み出して各機能を実現するものでもよい。すなわち、各プログラムの少なくとも一部が、署名検証装置20の機能をコンピュータに実行させるためのプログラムに相当する。

[0026] ＜処理＞

次に、本形態の処理について説明する。

[前処理]

まず、署名システム1で使用する位数qの離散対数問題の求解が困難な巡回群Gとその生成元$g \in G$とを決定する。このような巡回群Gとしては、例えば、楕円曲線上の有理点のなす群や有限体の乗法群などを用いることができる。なお、楕円曲線上の有理点のなす群を用いる場合、生成元gは楕円曲線上の点$g=(g_1, g_2)$であり、有限体の乗法群を用いる場合、生成元gは2以上の整数である。また、楕円曲線上の有理点のなす群をコンピュータ上で実現するための具体的方法には様々なもの(例えば、「N. Koblitz. Elliptic Curve Cryptosystems. Math. Comp., Vol. 48, No. 17, pp. 203-209, 1987.」「Victor S. Miller. Use of Elliptic Curves in Cryptography. In Advances in Cryptology - CRYPTO '85, Vol. 218 of Lecture Notes in Computer Science, pp. 417-426. Springer, 1986.」)が存在し、実際、楕円曲線上の有理点のなす群で構成され、コンピュータ上で実装可能な様々な暗号方式が存在する。また、安全性の面から位数qは素数であることが望ましいが、qの素因数分解が困難であるならqは素数でなくてもかまわない。また、署名システム1で使用するビット長パラメータ$L \in Z_{>0}$(0より大きな整数)を決定する。

[0027] さらに、後述するリカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がL+Mビットに定まる出力が可変長のハッシュ関数$H_0:\{0,1\}^* \rightarrow \{0,1\}^{L+M}$を決定し、リカバリメッセージ$m_{rec}$のビット長Mに応じて出力ビット長がMビットに定まる出力が可変長のハッシュ関数$H_2:\{0,1\}^* \rightarrow \{0,1\}^{M}$を決定する。なお、これらのハッシュ関数の

処理方法については後述する。

[0028] また、入力値に対してLビットのハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^*\rightarrow\{0,1\}^L$と、また、入力値に対して$Z_q$（qを法とする完全剰余系）の元を出力するハッシュ関数$H_3:\{0,1\}^*\rightarrow Z_q$とを決定する。ハッシュ関数$H_1$は、ハッシュ関数$H_0$やハッシュ関数$H_2$と同様に構成でき、ハッシュ関数$H_3$は、SHA－1等のハッシュ値に対し、qを法とした剰余演算を行うことで構成できる。

[0029] 以上のように決定された巡回群Gや各ハッシュ関数H_0～H_3を特定する情報は、例えば、署名生成装置10や署名検証装置20を構成する各プログラムに書き込まれ、署名生成装置10や署名検証装置20は、決定された巡回群Gでの演算や、各ハッシュ関数H_0～H_3の演算が可能になるものとする。また、ビット長パラメータ$L\in Z_{>0}$や位数qや生成元$g\in G$は、署名生成装置10の記憶部10aと署名検証装置20の記憶部20aに格納される。

[0030] ［鍵生成処理］

次に、署名生成装置10が行う鍵生成処理について説明する。

まず、署名生成装置10の秘密鍵生成部10bが任意の秘密鍵$x\in Z_q$を生成する。なお、この秘密鍵xの生成は、擬似乱数をZ_qにマッピングして行ってもよいし、署名生成者によって任意に決定された値を元に行ってもよい。生成された秘密鍵xは、署名生成装置10の記憶部10aに安全に格納される。すなわち、署名生成装置10の外部の装置は、記憶部10aから秘密鍵xを取得することができない。

[0031] 次に、署名生成装置10の公開鍵生成部10cが、記憶部10aから秘密鍵xと巡回群Gの生成元$g\in G$とを読み込み、巡回群Gで定義された演算

$$y=g^x\in G \quad \cdots(3)$$

を行って秘密鍵xに対応する公開鍵$y\in G$を生成し、記憶部10aに格納する。なお、例えば、巡回群Gが楕円曲線E上の有理点のなす群であった場合、式(3)の右辺は、楕円曲線E上の点である生成元$g=(g_1, g_2)$を楕円曲線E上でx倍する演算（$x\cdot g\in E$）を意味し、公開鍵yは楕円曲線E上の点となる。なお、楕円曲線上のスカラー倍演算をコンピュータ上で実行する具体的な方法としては、例えば、楕円曲線上の点をアフィン座標や射影座標で表し、二進展開法や移動窓法等を用いる演算方法を例

示できる(例えば、参考文献1“イアン・F・ブラケ、ガディエル・セロッシ、ナイジェル・P・スマート＝著、「楕円曲線暗号」、出版＝ピアソン・エデュケーション、ISBN4-89471-431-0”等参照)。また、例えば、巡回群Gが有限体の乗法群であった場合、式(3)の右辺は、$g^x \bmod p$(ただし、gは2以上の整数、p＝2q＋1)の演算を意味し、公開鍵yはスカラー値となる。生成された公開鍵yは、通信部10rからネットワーク40を通じて公開鍵サーバ装置30に送信され、公開鍵サーバ装置30は、送信された公開鍵yを例えば公開鍵証明書とともに公開する。なお、公開鍵y等の公開とは、公開鍵y等が公開鍵サーバ装置30の記憶部に格納され、ネットワーク40に接続可能な任意の装置が公開鍵サーバ装置30の記憶部に格納された公開鍵y等を取得可能な状態とすることを意味する。署名検証装置20は、このような公開鍵yを通信部20bによって公開鍵サーバ装置30から受信し、記憶部20aに格納する。

[0032] [署名生成処理]

次に、第1実施形態の署名生成処理について説明する。

図6は、第1実施形態の署名生成処理を説明するためのフローチャートである。以下、図6に従って本形態の署名生成処理を説明する。

まず、署名生成装置10(図3)の入力部10dに、メッセージ$m \in \{0, 1\}^{N+M}$とリカバリメッセージのビット長$M \geqq 1$とが入力される(ステップS11)。入力されたこれらの情報は、それぞれ記憶部10aに格納される。

[0033] 次に、メッセージ分割部10eが、記憶部10aからメッセージ$m \in \{0, 1\}^{N+M}$とリカバリメッセージのビット長$M \geqq 1$とを読み込む。メッセージ分割部10eは、これらの情報を用い、メッセージ$m \in \{0, 1\}^{N+M}$を、ビット長Mのリカバリメッセージ$m_{rec} \in \{0, 1\}^{M}$と、ビット長N(N≧0)のクリアメッセージ$m_{clr} \in \{0, 1\}^{N}$とに分割する(ステップS12)。例えば、メッセージ$m \in \{0, 1\}^{N+M}$の上位Mビットをリカバリメッセージ$m_{rec} \in \{0, 1\}^{M}$とし、下位Nビットをクリアメッセージ$m_{clr} \in \{0, 1\}^{N}$とする。なお、分割法はこれに限定されず、メッセージ$m \in \{0, 1\}^{N+M}$のどのビットをリカバリメッセージm_{rec}とし、どのビットをクリアメッセージm_{clr}とするかは、任意に設定できる。このように分割されたビット長Mのリカバリメッセージ$m_{rec} \in \{0, 1\}^{M}$と、ビット長Nのクリアメッセージ$m_{clr} \in \{0, 1\}^{N}$とは、それぞれ記憶部10aに格納される。

[0034] 次に、任意値生成部10fが任意値$k \in Z_q$を生成し、生成した任意値kを記憶部10aに格納する(ステップS13)。なお、任意値kの生成は、例えば、擬似乱数をZ_qにマッピングすることにより行う。

次に、群演算部10gが、記憶部10aから生成元$g \in G$と任意値$k \in Z_q$とを読み込み、

$$R=g^k \in G \quad \cdots(4)$$

を算出し、当該演算結果$R \in G$を記憶部10aに出力して格納する(ステップS14)。なお、例えば、巡回群Gが楕円曲線E上の有理点のなす群であった場合、式(4)の右辺は、楕円曲線E上の点である生成元$g=(g_1, g_2)$を楕円曲線E上でk倍する演算($k \cdot g \in E$)を意味し、演算結果Rは楕円曲線E上の点となる。なお、楕円曲線上のスカラー倍演算をコンピュータ上で実行する具体的な方法としては、例えば、楕円曲線上の点をアフィン座標や射影座標で表し、二進展開法や移動窓法等を用いる演算方法を例示できる。また、例えば、巡回群Gが有限体の乗法群であった場合、式(4)の右辺は、$g^k \bmod p$の演算を意味し、演算結果Rはスカラー値となる。

[0035] 次に、ハッシュ演算部10hが、記憶部10aから演算結果$R \in G$とリカバリメッセージのビット長Mとビット長パラメータLを読み込む。ハッシュ演算部10hは、リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がL+Mビットに定まるハッシュ関数$H_0 : \{0, 1\}^* \rightarrow \{0, 1\}^{L+M}$を演算結果Rに作用させ、その演算結果であるL+Mビットのハッシュ値

$$\Pi=H_0(R) \in \{0,1\}^{L+M} \quad \cdots(5)$$

を記憶部10aに出力して格納する(ステップS15)。なお、例えば、巡回群Gが楕円曲線E上の有理点のなす群であった場合、式(5)の右辺は、楕円曲線E上の点である演算結果Rを一義的又は限定的に特定できる値(例えば、点Rのx座標とy座標の符号との組み合わせ値、点Rのx座標若しくはy座標、又は、点Rのx座標とy座標とのビット結合値)にハッシュ関数H_0を作用させる演算を意味する。すなわち、この場合の「ハッシュ関数H_0を演算結果Rに作用させる」とは、ハッシュ関数H_0を、楕円曲線E上の点である演算結果Rを一義的又は限定的に特定できる値に作用させることを意味する。また、例えば、巡回群Gが有限体の乗法群であった場合、式(5)の右辺は、ス

カラー値である演算結果Rにハッシュ関数H_0を作用させる演算を意味する。

[0036] ［ステップS15の処理の例］

図7Aは、ステップS15の処理の例を説明するためのフローチャートである。

まず、リカバリメッセージのビット長Mとビット長パラメータLとがハッシュ演算回数算出部10haに読み込まれる。ハッシュ演算回数算出部10haは、

$$e_{max}=\text{rounddown}\{(L+M)/\text{length}(H)\} \quad \cdots(5\text{-}1)$$

の演算を行ってe_{max}を一時メモリ10tに格納する（ステップS15a）。なお、rounddown{＊}は＊の小数点以下を切り捨てる演算を意味し、length（＊）は＊のビット長を意味し、Hは公知のハッシュ関数を意味する。なお、ハッシュ関数Hの具体例としては、SHA－1（ビット長160ビット）やMD5（ビット長128ビット）などを例示できる。例えば、L＋M＝500であり、ハッシュ関数HがSHA－1〔length（H）＝160〕であるならば、e_{max}＝3である。

[0037] 次に、制御部10sは変数eに0を代入し、変数eを一時メモリ10tに格納する（ステップS15b）。

次に、部分ハッシュ演算部10hbが、一時メモリ10tから変数eを読み込み、記憶部10aから演算結果Rを読み込み、ハッシュ値

$$H(e,R) \quad \cdots(5\text{-}2)$$

を算出して一時メモリ10tに格納する（ステップS15c）。なお、例えば、巡回群Gが楕円曲線E上の有理点のなす群であった場合、式（5－2）は、楕円曲線E上の点である演算結果Rを一義的又は限定的に特定できる値（例えば、点Rのx座標とy座標の符号との組み合わせ値、点Rのx座標若しくはy座標、又は、点Rのx座標とy座標とのビット結合値）と変数eとのビット結合値にハッシュ関数Hを作用させる演算を意味する。また、例えば、巡回群Gが有限体の乗法群であった場合、式（5－2）は、スカラー値である演算結果Rと変数eとのビット結合値にハッシュ関数H_0を作用させる演算を意味する。

[0038] 次に、制御部10sが、一時メモリ10tからe_{max}と変数eとを読み込み、

$$e=e_{max} \quad \cdots(5\text{-}3)$$

を満たすか否かを判断する（ステップS15d）。ここで、式（5－3）を満たさなければ、

制御部10sはe＋1を新たなeとし、新たなeを一時メモリ10tに格納した後（ステップS15e）、処理をステップS15cに戻す。一方、式（5－3）を満たすのであれば、制御部10sはビット結合部10hcに指示を与え、ビット結合部10hcは、一時メモリ10tから各ハッシュ値H(0,R),H(1,R),H(2,R),…,H(e_{max},R)を読み込み、それらのビット結合値

$$HC(R)=H(0,R)|\cdots|H(e_{max},R) \quad \cdots(5-4)$$

を算出して一時メモリ10tに格納する（ステップS15f）。

[0039] 次に、ビット削除部10hdが、一時メモリ10tから、ビット結合値HC（R）とリカバリメッセージのビット長Mとビット長パラメータLとを読み込み、

$$\Pi=H_0(R)=delete\{length(HC(R))-(L+M),HC(R)\} \quad \cdots(5-5)$$

を算出して記憶部10aに出力する（ステップS15g）。なお、delete{δ，ε}は、εのビットを先頭からδビットだけ削除する処理を意味する。すなわち、式（5－5）は、HC（R）の先頭ビットを削除して全体のビット長をL＋Mとしたものを$\Pi=H_0(R)$とする演算を意味する。

[0040] なお、ステップS15の処理方法はこれに限定されない。例えば、eを用いるのではなく、ハッシュチェインによってハッシュ値のビット長を拡張する方法でもよい。この場合、式（5－4）のHC（R）は、例えば、

$$HC(R)=H(R)|H(H(R))|H(H(H(R)))|\cdots|H(H(H\cdots(R)\cdots))$$

となる（[ステップS15の処理の例]の説明終わり）。

[0041] ステップS15の後、ハッシュ演算部10iが、記憶部10aからハッシュ値Πとリカバリメッセージm_{rec}とビット長パラメータLとを読み込む。ハッシュ演算部10iは、入力値に対してLビットのハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^*\rightarrow\{0,1\}^L$を、ハッシュ値Πとリカバリメッセージ$m_{rec}$とに依存する値αに作用させ、その演算結果であるLビットのハッシュ値

$$h=H_1(\alpha)\in\{0,1\}^L \quad \cdots(6)$$

を記憶部10aに出力して格納する（ステップS16）。なお、第1実施形態では、αはハッシュ値Πとリカバリメッセージm_{rec}とのみに依存する値$\alpha=(\Pi, m_{rec})$である。なお、本形態のαの構成方法に限定はないが、αの構成方法は後述する署名検証装置20でのα’（後述）の構成方法と同一とする。αの構成例としては以下のようなものがあ

る。

[0042] 〔α－1〕Πを上位L＋Mビットとし、m_{rec}を下位Mビットとして結合したL＋2Mビットの値をαとする。

〔α－2〕Πを下位L＋Mビットとし、m_{rec}を上位Mビットとして結合したL＋2Mビットの値をαとする。

〔α－3〕m_{rec}を上位から奇数番目のビット（合計Mビット）とし、Πをその他のL＋Mビットとして結合したL＋2Mビットの値をαとする。

[0043] 次に、ハッシュ演算部10jが、記憶部10aからハッシュ値Πとハッシュ値hとリカバリメッセージのビット長Mとを読み込む。ハッシュ演算部10jは、リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がMビットに定まるハッシュ関数$H_2:\{0,1\}^*\rightarrow\{0,1\}^M$を、ハッシュ値Πとハッシュ値hとに依存する値βに作用させ、その演算結果であるMビットのハッシュ値

$$u=H_2(\beta)\in\{0,1\}^M \quad \cdots(7)$$

を記憶部10aに出力して格納する（ステップS17）。なお、第1実施形態では、βはハッシュ値Πとハッシュ値hとのみに依存する値β＝（Π，h）である。本形態のβの構成方法に限定はないが、βの構成方法は後述する署名検証装置20でのβ’（後述）の構成方法と同一とする。βの構成例としては以下のようなものがある。

[0044] 〔β－1〕Πを上位L＋Mビットとし、hを下位Lビットとして結合した2L＋Mビットの値をβとする。

〔β－2〕Πを下位L＋Mビットとし、hを上位Lビットとして結合した2L＋Mビットの値をβとする。

〔β－3〕hを上位から奇数番目のビット（合計Lビット）とし、Πをその他のL＋Mビットとして結合した2L＋Mビットの値をβとする。

[0045] ［ステップS17の処理の例］

図7Bは、ステップS17の処理の例を説明するためのフローチャートである。

まず、リカバリメッセージのビット長Mがハッシュ演算回数算出部10jaに読み込まれる。ハッシュ演算回数算出部10jaは、

$$e_{max}=\text{rounddown}\{M/\text{length}(H)\} \quad \cdots(7\text{-}1)$$

の演算を行ってe_{max}

を一時メモリ10tに格納する(ステップS17a)。

[0046] 次に、制御部10sは変数eに0を代入し、変数eを一時メモリ10tに格納する(ステップS17b)。

次に、部分ハッシュ演算部10jbが、一時メモリ10tから変数eを読み込み、記憶部10aからハッシュ値Π, hを読み込み、ハッシュ値

$$H(e,\beta),\beta=(\Pi,h) \quad \cdots(7\text{-}2)$$

を算出して一時メモリ10tに格納する(ステップS17c)。

[0047] 次に、制御部10sが、一時メモリ10tからe_{max}と変数eとを読み込み、

$$e=e_{max} \quad \cdots(7\text{-}3)$$

を満たすか否かを判断する(ステップS17d)。ここで、式(7－3)を満たさないのであれば、制御部10sはe＋1を新たなeとし、新たなeを一時メモリ10tに格納した後(ステップS17e)、処理をステップS17cに戻す。一方、式(7－3)を満たすのであれば、制御部10sはビット結合部10jcに指示を与え、ビット結合部10jcは、一時メモリ10tから各ハッシュ値$H(0,\beta)$, $H(1,\beta)$, $H(2,\beta)$, …, $H(e_{max},\beta)$を読み込み、それらのビット結合値

$$HC(\beta)=H(0,\beta)|\cdots|H(e_{max},\beta) \quad \cdots(7\text{-}4)$$

を算出して一時メモリ10tに格納する(ステップS17f)。

[0048] 次に、ビット削除部10jdが、一時メモリ10tから、ビット結合値$HC(\beta)$とリカバリメッセージのビット長Mを読み込み、

$$u=H_2(\beta)=delete\{length(HC(\beta))-M,HC(\beta)\} \quad \cdots(7\text{-}5)$$

を算出して記憶部10aに出力する(ステップS17g)。

なお、ステップS17の処理方法はこれに限定されない。例えば、eを用いるのではなく、ハッシュチェインによってハッシュ値のビット長を拡張する方法でもよい([ステップS17の詳細処理の例]の説明終わり)。

[0049] ステップS17の後、排他的論理和演算部10kが、記憶部10aからリカバリメッセージm_{rec}とハッシュ値uとを読み込む。排他的論理和演算部10kは、リカバリメッセージm_{re}とハッシュ値uとの排他的論理和

$$w=m_{rec}(+)u\in\{0,1\}^{M} \quad \cdots(8)$$

((＋)は排他的論理和演算子)を算出し、当該排他的論理和値wを記憶部10aに出力して格納する(ステップS18)。

[0050] 次に、ビット結合部10mが、記憶部10aからハッシュ値$h\in\{0,1\}^{L}$と排他的論理和値$w\in\{0,1\}^{M}$とを読み込む。ビット結合部10mは、ハッシュ値$h\in\{0,1\}^{L}$を第1ビット位置に配置し、排他的論理和値$w\in\{0,1\}^{M}$を第2ビット位置に配置したL＋Mビットのビット結合値

$$d=h|w\in\{0,1\}^{L+M} \quad \cdots(9)$$

を算出し、当該ビット結合値dを記憶部10aに出力して格納する(ステップS19)。なお、「第1ビット位置」及び「第2ビット位置」をどのビット位置にするかについて特に限定はない。しかし、署名生成装置10と署名検証装置20との間では、「第1ビット位置」及び「第2ビット位置」をどのビット位置にするかの基準を同一にしなければならない。図9に「第1ビット位置」及び「第2ビット位置」の設定例を示す。

[0051] 図9Aの例は、連続した上位Lビットを「第1ビット位置」とし、連続した下位Mビットを「第2ビット位置」とした例である。図9Bの例は、連続した上位Mビットを「第2ビット位置」とし、連続した下位Lビットを「第1ビット位置」とした例である。また、図9(c)の例は、L≧Mの場合の例であり、上位から奇数番目のビット(合計Mビット)の位置を「第2ビット位置」とし、その他のビット位置を「第1ビット位置」とした例である。

[0052] 次に、排他的論理和演算部10nが、記憶部10aからハッシュ値Πとビット結合値dとを読み込む。排他的論理和演算部10nは、ハッシュ値Πとビット結合値dとの排他的論理和

$$r=\Pi(+)d\in\{0,1\}^{L+M} \quad \cdots(10)$$

を算出し、当該排他的論理和値rを記憶部10aに出力して格納する(ステップS20)。

[0053] 次に、ハッシュ演算部10pが、記憶部10aから排他的論理和値rとクリアメッセージm_{clr}とを読み込む。ハッシュ演算部10pは、入力値に対して整数を出力するハッシュ関数$H_3:\{0,1\}^{*}\rightarrow Z_q$を、排他的論理和値rとクリアメッセージ$m_{clr}$とに依存する値γに作用させ、その演算結果であるハッシュ値

$$t=H_3(\gamma)\in Z_q \quad \cdots(11)$$

を記憶部10aに出力して格納する(ステップS21)。なお、第1実施形態では、γは排他的論理和値rとクリアメッセージm_{clr}とのみに依存する値γ＝(r, m_{clr})である。本形態のγの構成方法に限定はないが、γの構成方法は後述する署名検証装置20でのγ'(後述)の構成方法と同一とする。γの構成例としては以下のようなものがある。

[0054] 〔γ－1〕rを上位L＋Mビットとし、m_{clr}を下位Nビットとして結合したL＋M＋Nビットの値をγとする。

〔γ－2〕rを下位L＋Mビットとし、m_{clr}を上位Nビットとして結合したL＋M＋Nビットの値をγとする。

〔γ－3〕m_{clr}を上位から奇数番目のビット(合計Nビット)とし、rをその他のL＋Mビットとして結合したL＋M＋Nビットの値をγとする。

[0055] 次に、整数演算部10qが、記憶部10aから任意値kとハッシュ値tと秘密鍵xとqとを読み込む。整数演算部10qは、

$$s=k-t\cdot x\in Z_q \quad \cdots(12)$$

を算出し、当該演算結果sを記憶部10aに出力して格納する(ステップS22)。

次に、通信部10rに、排他的論理和値rと演算結果sとクリアメッセージm_{clr}とが読み込まれ、通信部10rは、署名σ＝(r, s)とクリアメッセージm_{clr}とをネットワーク40を通じて署名検証装置20に送信する(ステップS23)。

[0056] [署名検証処理]

次に、第1実施形態の署名検証処理について説明する。

図8は、第1実施形態の署名検証処理を説明するためのフローチャートである。以下、図8に従って本形態の署名検証処理を説明する。

まず、署名検証装置20(図5)の通信部20bが、署名σ'＝(r', s')とクリアメッセージm_{clr}'とを受信し(「入力を受け付け」に相当)、これらを記憶部20aに格納する(ステップS41)。なお、署名とクリアメッセージとが正規なものであれば、σ'＝(r', s')＝σ＝(r, s)であり、m_{clr}'＝m_{clr}であるが、ここでは、検証対象の署名をσ'＝(r', s')と表現し、検証対象のクリアメッセージをm_{clr}'と表現する。

[0057] 次に、ビット長抽出部20cが、記憶部20aから、ビット長パラメータLと署名σ'＝(r', s')のr'とを読み込む。ビット長抽出部20cは、

$M'=length(r')-L$ …(13)

により、署名σ'に対応するリカバリメッセージm_{rec}'のビット長M'を算出し、記憶部20aに格納する(ステップS42)。

[0058] 次に、ハッシュ演算部20dが、記憶部20aからr'とクリアメッセージをm_{clr}'とqとを読み込む。ハッシュ演算部20dは、署名生成装置10と同一のハッシュ関数$H_3:\{0,1\}^*\rightarrow Z_q$を、r'と$m_{clr}$'とに依存する値γ'に作用させ、その演算結果であるハッシュ値

$t'=H_3(\gamma')$ …(14)

を記憶部20aに出力して格納する(ステップS43)。なお、γ'の構成方法は前述した署名生成装置10でのγの構成方法と同一(r=r'とし、$m_{clr}=m_{clr}$'とした場合に同一)とする。

[0059] 次に、群演算部20eが、記憶部20aから、生成元g∈Gと署名生成装置10の公開鍵y∈Gと署名σ'のs'とハッシュ値t'とを読み込み、

$R'=g^{s'}\cdot y^{t'}\in G$ …(15)

の演算を行い、その演算結果R'を記憶部20aに出力して格納する(ステップS44)。なお、例えば、巡回群Gが楕円曲線E上の有理点のなす群であった場合、式(15)の右辺は、楕円曲線E上の点である生成元$g=(g_1, g_2)$を楕円曲線E上でs'倍し、公開鍵$y=(y_1, y_2)$を楕円曲線E上でt'倍し、それらの演算結果を楕円曲線E上で加算する演算(s'・g+t'・y∈E)を意味し、演算結果R'は楕円曲線E上の点となる。なお、楕円曲線上のスカラー倍演算をCPU上で実行する具体的な方法としては、例えば、楕円曲線上の点をアフィン座標や射影座標で表し、二進展開法や移動窓法等を用いる演算方法を例示できる。また、例えば、巡回群Gが有限体の乗法群であった場合、式(15)の右辺は、$g^{s'}\cdot y^{t'} \bmod p$の演算を意味し、演算結果R'はスカラー値となる。

[0060] 次に、ハッシュ演算部20fが、記憶部20aから演算結果R'∈Gとリカバリメッセージm_{rec}'のビット長M'とビット長パラメータLを読み込む。ハッシュ演算部20fは、署名生成装置10と同じハッシュ関数$H_0:\{0,1\}^*\rightarrow\{0,1\}^{L+M}$を演算結果R'に作用させ、その演算結果であるL+M'ビットのハッシュ値

$\Pi'=H_0(R')\in\{0,1\}^{L+M'}$ …(16)

を記憶部20aに出力して格納する(ステップS45)。なお、$H_0(R')$の演算は、署名生成装置10の場合と同一(R=R'の場合に同一)とする。

[0061] 次に、排他的論理和演算部20gが、記憶部20aから、ハッシュ値Π'と署名σ'が有するr'とを読み込み、それらの排他的論理和

$$d'=\Pi'(+)r'\in\{0,1\}^{L+M'} \quad \cdots(17)$$

を算出し、当該排他的論理和値d'を記憶部20aに出力して格納する(ステップS46)。

次に、ビット抽出部20hが、記憶部20aから排他的論理和値d'とリカバリメッセージm_{rec}'のビット長M'とを読み込む。ビット抽出部20hは、排他的論理和値d'の第1ビット位置のLビットの値$h'\in\{0,1\}^L$と、排他的論理和値d'の第2ビット位置のM'ビットの値$w'\in\{0,1\}^{M'}$とを抽出し、それらを記憶部20aに格納する(ステップS47)。なお、「第1ビット位置」及び「第2ビット位置」は、署名生成装置10の処理での「第1ビット位置」及び「第2ビット位置」と同一(d=d'とした場合に同一)とする。

[0062] 次に、ハッシュ演算部20iが、記憶部20aからハッシュ値Π'とh'とリカバリメッセージm_{rec}'のビット長M'とを読み込む。ハッシュ演算部20iは、署名生成装置10と同一のハッシュ関数$H_2:\{0,1\}^*\rightarrow\{0,1\}^M$を、ハッシュ値Π'と値h'とに依存する値β'に作用させ、その演算結果であるM'ビットのハッシュ値

$$u'=H_2(\beta')\in\{0,1\}^{M'} \quad \cdots(18)$$

を記憶部20aに出力して格納する(ステップS48)。なお、β'の構成方法は前述した署名生成装置10でのβの構成方法と同一(Π=Π'とし、h=h'とした場合に同一)とする。

[0063] 排他的論理和演算部20jが、記憶部20aから値$w'\in\{0,1\}^{M'}$とハッシュ値u'とを読み込む。排他的論理和演算部20jは、値w'とハッシュ値u'との排他的論理和

$$m_{rec}'=w'(+)u'\in\{0,1\}^{M'} \quad \cdots(19)$$

を算出し、その演算結果をリカバリメッセージ$m_{rec}'\in\{0,1\}^{M'}$として記憶部20aに出力して格納する(ステップS49)。

次に、ハッシュ演算部20kが、記憶部20aからハッシュ値Π'とリカバリメッセージm_{rec}'とを読み込む。ハッシュ演算部20kは、署名生成装置10と同一のハッシュ関数H_1:

$\{0, 1\}^* \rightarrow \{0, 1\}^L$を、ハッシュ値Π'とリカバリメッセージ$m_{rec}$'とに依存する値α'に作用させ、その演算結果であるLビットのハッシュ値

$$H_1(\alpha') \in \{0,1\}^L \quad \cdots(20)$$

を記憶部20aに出力して格納する(ステップS50)。なお、α'の構成方法は前述した署名生成装置10でのαの構成方法と同一(Π＝Π'とし、m_{rec}＝m_{rec}'とした場合に同一)とする。

[0064] 次に、比較部20lが、記憶部20aからハッシュ値$H_1(\alpha')$と値h'とを読み込み、

$$h' = H_1(\alpha') \quad \cdots(21)$$

を満たすか否かを判断する(ステップS51)。

ここで、式(21)を満たさない場合、比較部20lは0(検証失敗)を記憶部20aに出力して格納し、出力部20mは、記憶部20aから送られた0(検証失敗)を出力する(ステップS52)。一方、式(21)を満たす場合、比較部20lは1(検証成功)を記憶部20aに出力して格納し、出力部20mは、記憶部20aから送られた1(検証成功)を出力し(ステップS53)、さらにリカバリメッセージm_{rec}'を出力する(ステップS54)。

[0065] 〔第2実施形態〕

次に、本発明の第2実施形態について説明する。第2実施形態ではクリアメッセージを用いない。この点が第1実施形態との相違点である。以下では、第1実施形態との相違点を中心に説明し、第1実施形態と共通する事項については説明を省略する。

＜全体構成＞

第1実施形態の署名システム1の署名生成装置10が署名生成装置110に置換され、署名検証装置20が署名検証装置120に置換された構成である。

[0066] ＜署名生成装置110の構成＞

次に、署名生成装置110の構成を説明する。

[ハードウェア構成]

第1実施形態の署名生成装置10と同じである。

[ハードウェアとプログラムとの協働]

署名生成装置110もコンピュータに所定のプログラムが読み込まれることにより構成

される。

[0067] 図10は、このように構成される第2実施形態における署名生成装置110の機能構成を例示したブロック図である。なお、署名生成装置110において署名生成装置10と共通する部分には図3と同じ符号を付して説明を簡略化する。

図10に示すように、本形態の署名生成装置110は、記憶部10aと、秘密鍵生成部10bと、公開鍵生成部10cと、入力部110dと、ビット長抽出部110eと、任意値生成部10fと、群演算部10gと、ハッシュ演算部10h, 10i, 10j, 110pと、排他的論理和演算部10k, 10nと、ビット結合部10mと、整数演算部10qと、通信部110rと、制御部10sと、一時メモリ10tとを有する。

[0068] なお、ビット長抽出部110eとハッシュ演算部110pは、それぞれの処理を実現するためのプログラムがCPUに読み込まれることにより構成されるものである。また、入力部110dは、所定のプログラムが読み込まれたCPUの制御のもと駆動し、通信部110rは、所定のプログラムが読み込まれたCPUの制御のもと駆動する。

また、上記のプログラムは単体でその機能を実現できるものでもよいし、当該プログラムがさらに他のライブラリ(記載していない)を読み出して各機能を実現するものでもよい。すなわち、各プログラムの少なくとも一部が、署名生成装置110の機能をコンピュータに実行させるためのプログラムに相当する。

[0069] ＜署名検証装置120の構成＞

次に、署名検証装置120の構成を説明する。

[ハードウェア構成]審査請求時

第1実施形態の署名検証装置20と同じである。

[ハードウェアとプログラムとの協働]

署名検証装置120もコンピュータに所定のプログラムが読み込まれることにより構成される。図11は、このように構成される第2実施形態の署名検証装置120の機能構成を例示したブロック図である。

[0070] 図11に示すように、本形態の署名検証装置120は、記憶部20aと、通信部120bと、ビット長抽出部20cと、ハッシュ演算部120d, 20f, 20i, 20kと、群演算部20eと、排他的論理和演算部20gと、ビット抽出部20hと、排他的論理和演算部20jと、比較

部20lと、出力部20mと、制御部20nと、一時メモリ20pとを有する。

[0071] なお、ハッシュ演算部120dは、処理を実現するためのプログラムがCPUに読み込まれることにより構成されるものである。また、通信部120bは、所定のプログラムが読み込まれたCPUの制御のもと駆動する。また、上記のプログラムは単体でその機能を実現できるものでもよいし、当該プログラムがさらに他のライブラリ(記載していない)を読み出して各機能を実現するものでもよい。すなわち、各プログラムの少なくとも一部が、署名検証装置120の機能をコンピュータに実行させるためのプログラムに相当する。

[0072] <処理>

次に、本形態の処理について説明する。

[前処理・鍵生成処理]

第1実施形態と同じである。

[署名生成処理]

次に、第2実施形態の署名生成処理について説明する。

図12は、第2実施形態の署名生成処理を説明するためのフローチャートである。以下、図12に従って本形態の署名生成処理を説明する。

まず、署名生成装置110(図10)の入力部110dに、リカバリメッセージ$m_{rec} \in \{0, 1\}^M$が入力される(ステップS111)。入力されたリカバリメッセージm_{rec}は、記憶部10aに格納される。なお、第2実施形態では$m = m_{rec}$である。

次に、ビット長抽出部110eが、記憶部10aからリカバリメッセージ$m_{rec} \in \{0, 1\}^M$を読み込み、そのビット長Mを抽出して記憶部10aに格納する(ステップS112)。

[0073] その後、署名生成装置110において第1実施形態のステップS13～S20と同じ処理(ステップS113～S120)が実行された後、ハッシュ演算部110pが、記憶部10aから排他的論理和値rを読み込む。ハッシュ演算部110pは、第1実施形態と同じハッシュ関数$H_3 : \{0, 1\}^* \rightarrow Z_q$を、排他的論理和値rに依存する値γに作用させ、その演算結果であるハッシュ値

$$t = H_3(\gamma) \in Z_q \quad \cdots(22)$$

を記憶部10aに出力して格納する(ステップS121)。なお、第2実施形態では、γは

排他的論理和値rのみに依存する値γ＝rである。本形態のγの構成方法に限定はないが、γの構成方法は後述する署名検証装置120でのγ’（後述）の構成方法と同一とする。

[0074] 次に、整数演算部10qが、記憶部10aから任意値kとハッシュ値tと秘密鍵xとqとを読み込み、前述の式(12)によってsを算出し、当該演算結果sを記憶部10aに出力して格納する(ステップS122)。

次に、通信部110rに、排他的論理和値rと演算結果sが読み込まれ、通信部10rは、署名σ＝(r, s)をネットワーク40を通じて署名検証装置120に送信する(ステップS123)。

[署名検証処理]

次に、第2実施形態の署名検証処理について説明する。

[0075] 図13は、第2実施形態の署名検証処理を説明するためのフローチャートである。以下、図13に従って本形態の署名検証処理を説明する。

まず、署名検証装置120(図11)の通信部120bが、署名σ’＝(r’, s’)を受信し(「入力を受け付け」に相当)、これらを記憶部20aに格納する(ステップS141)。

次に、ビット長抽出部20cが、記憶部20aから、ビット長パラメータLと署名σ’＝(r’, s’)のr’とを読み込み、前述の式(13)によって署名σ’に対応するリカバリメッセージm_{rec}’のビット長M’を算出し、記憶部20aに格納する(ステップS142)。

[0076] 次に、ハッシュ演算部120dが、記憶部20aからr’とqとを読み込む。ハッシュ演算部120dは、署名生成装置110と同一のハッシュ関数$H_3:\{0,1\}^*\rightarrow Z_q$を、r’に依存する値γ’に作用させ、その演算結果であるハッシュ値

$$t'=H_3(\gamma') \quad \cdots(23)$$

を記憶部20aに出力して格納する(ステップS143)。なお、γ’の構成方法は前述した署名生成装置110でのγの構成方法と同一(r＝r’とした場合に同一)とする。

その後、第1実施形態のステップS44～S54と同じ処理によって署名検証を行う(ステップS144～S154)。

[0077] 〔第3実施形態〕

次に、本発明の第3実施形態について説明する。本形態は第1実施形態の変形例

であり、署名σ＝(r, s)を構成するrを簡略化した例である。すなわち、第1実施形態においてr=H_0(R)(+)(H_1(H_0(R), m_{rec})|m_{rec}(+)H_2(H_0(R),H_1(H_0(R),m_{rec})))としていたのに対し、第3実施形態ではr=H_1(R, m_{rec})|m_{rec}(+) H_2(R, H_1(R, m_{rec}))とする。これにより演算量を削減できる。以下では、第1実施形態との相違点を中心に説明し、第1実施形態と共通する事項については説明を省略する。

[0078] ＜全体構成＞

第1実施形態の署名システム1の署名生成装置10が署名生成装置210に置換され、署名検証装置20が署名検証装置220に置換された構成である。

＜署名生成装置210の構成＞

次に、署名生成装置210の構成を説明する。

[ハードウェア構成]

第1実施形態の署名生成装置10と同じである。

[0079] [ハードウェアとプログラムとの協働]

署名生成装置210もコンピュータに所定のプログラムが読み込まれることにより構成される。

図14は、このように構成される第3実施形態における署名生成装置210の機能構成を例示したブロック図である。なお、署名生成装置210において署名生成装置10と共通する部分には図3と同じ符号を付して説明を簡略化する。

図14に示すように、本形態の署名生成装置210は、記憶部10aと、秘密鍵生成部10bと、公開鍵生成部10cと、入力部10dと、メッセージ分割部10eと、任意値生成部10fと、群演算部10gと、ハッシュ演算部210i, 210j, 10pと、排他的論理和演算部10kと、ビット結合部210mと、整数演算部10qと、通信部10rと、制御部10sと、一時メモリ10tとを有する。

[0080] なお、ハッシュ演算部210i, 210j, 10pとビット結合部210mとは、それぞれの処理を実現するためのプログラムがCPUに読み込まれることにより構成されるものである。

また、上記のプログラムは単体でその機能を実現できるものでもよいし、当該プログラムがさらに他のライブラリ(記載していない)を読み出して各機能を実現するものでもよい。すなわち、各プログラムの少なくとも一部が、署名生成装置210の機能をコンピ

ュータに実行させるためのプログラムに相当する。

[0081] ＜署名検証装置220の構成＞

次に、署名検証装置220の構成を説明する。

[ハードウェア構成]

第1実施形態の署名検証装置20と同じである。

[ハードウェアとプログラムとの協働]

署名検証装置220もコンピュータに所定のプログラムが読み込まれることにより構成される。図11は、このように構成される第2実施形態の署名検証装置120の機能構成を例示したブロック図である。

[0082] 図15に示すように、本形態の署名検証装置220は、記憶部20aと、通信部20bと、ビット長抽出部20cと、ハッシュ演算部20d，220i，220kと、群演算部20eと、ビット抽出部220hと、排他的論理和演算部20jと、比較部20lと、出力部20mと、制御部20nと、一時メモリ20pとを有する。

なお、ハッシュ演算部220i，220kと比較部20lは、処理を実現するためのプログラムがCPUに読み込まれることにより構成されるものである。また、上記のプログラムは単体でその機能を実現できるものでもよいし、当該プログラムがさらに他のライブラリ(記載していない)を読み出して各機能を実現するものでもよい。すなわち、各プログラムの少なくとも一部が、署名検証装置120の機能をコンピュータに実行させるためのプログラムに相当する。

[0083] ＜処理＞

次に、本形態の処理について説明する。

[前処理]

ハッシュ関数H_0が設定されない点が第1実施形態との相違点である。

[鍵生成処理]

第1実施形態と同じである。

[署名生成処理]

次に、第3実施形態の署名生成処理について説明する。

[0084] 図16は、第3実施形態の署名生成処理を説明するためのフローチャートである。以

下、第1実施形態との相違点を中心に説明する。

まず、署名生成装置210が、第1実施形態のステップS11～S14と同じ処理を実行する(ステップS211～S214)。次に、ハッシュ演算部10iが、記憶部10aからステップS114の演算結果Rとリカバリメッセージm_{rec}とビット長パラメータLとを読み込む。ハッシュ演算部10iは、入力値に対してLビットのハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^*\rightarrow\{0,1\}^L$を、演算結果Rとリカバリメッセージ$m_{rec}$とに依存する値$\alpha$に作用させ(式(6))、その演算結果であるLビットのハッシュ値hを記憶部10aに出力して格納する(ステップS215)。なお、第3実施形態では、αは演算結果$R\in G$とリカバリメッセージm_{rec}とのみに依存する値$\alpha=(R, m_{rec})$である。なお、巡回群Gが有限体の乗法群であった場合、本形態のαの構成は、ΠがRに置換される点を除き、第1実施形態と同様である。また、巡回群Gが楕円曲線E上の有理点のなす群である場合、本形態のαの構成は、Πが、楕円曲線E上の点である演算結果Rを一義的又は限定的に特定できる値(例えば、点Rのx座標とy座標の符号との組み合わせ値、点Rのx座標若しくはy座標、又は、点Rのx座標とy座標とのビット結合値)に置換される点を除き、第1実施形態と同様である。

[0085] 次に、ハッシュ演算部210jが、記憶部10aから演算結果Rとハッシュ値hとリカバリメッセージのビット長Mとを読み込む。ハッシュ演算部210jは、リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がMビットに定まるハッシュ関数$H_2:\{0,1\}^*\rightarrow\{0,1\}^M$を、演算結果Rとハッシュ値hとに依存する値$\beta$に作用させ(式(7))、その演算結果であるMビットのハッシュ値uを記憶部10aに出力して格納する(ステップS216)。なお、第3実施形態では、βは演算結果Rとハッシュ値hとのみに依存する値$\beta=(R, h)$である。なお、巡回群Gが有限体の乗法群であった場合、本形態のβの構成は、ΠがRに置換される点を除き、第1実施形態と同様である。また、巡回群Gが楕円曲線E上の有理点のなす群である場合、本形態のβの構成は、Πが、楕円曲線E上の点である演算結果Rを一義的又は限定的に特定できる値(例えば、点Rのx座標若しくはy座標、又は、点Rのx座標とy座標とのビット結合値)に置換される点を除き、第1実施形態と同様である。

[0086] 次に、排他的論理和演算部10kが、記憶部10aからリカバリメッセージm_{rec}とハッシ

ュ値uとを読み込む。排他的論理和演算部10kは、リカバリメッセージm_{rec}とハッシュ値uとの排他的論理和w(式(8))を算出し、当該排他的論理和値wを記憶部10aに出力して格納する(ステップS217)。

[0087] 次に、ビット結合部210mが、記憶部10aからハッシュ値$h \in \{0, 1\}^L$と排他的論理和値$w \in \{0, 1\}^M$とを読み込む。ビット結合部210mは、ハッシュ値$h \in \{0, 1\}^L$を第1ビット位置に配置し、排他的論理和値$w \in \{0, 1\}^M$を第2ビット位置に配置したL+Mビットのビット結合値

$$r=h|w \in \{0,1\}^{L+M} \quad \cdots(24)$$

を算出し、当該ビット結合値rを記憶部10aに出力して格納する(ステップS218)。「第1ビット位置」及び「第2ビット位置」をどのビット位置にするかについては第1実施形態と同様である。

その後、第1実施形態のステップS21～S23と同じ処理が実行される(ステップS219～S221)。

[0088] [署名検証処理]

次に、第3実施形態の署名検証処理について説明する。

図17は、第1実施形態の署名検証処理を説明するためのフローチャートである。以下、第1実施形態との相違点を中心に説明する。

まず、署名検証装置220が、第1実施形態のステップS41～S44と同じ処理を実行する(ステップS241～S244)。

[0089] 次に、ビット抽出部220hが、記憶部20aから署名$\sigma' = (r', s')$のr'とリカバリメッセージm_{rec}'のビット長M'とを読み込む。ビット抽出部220hは、r'の第1ビット位置のLビットの値$h' \in \{0, 1\}^L$と、r'の第2ビット位置のM'ビットの値$w' \in \{0, 1\}^{M'}$とを抽出し、それらを記憶部20aに格納する(ステップS245)。なお、「第1ビット位置」及び「第2ビット位置」は、署名生成装置210の処理での「第1ビット位置」及び「第2ビット位置」と同一(d=d'とした場合に同一)とする。

[0090] 次に、ハッシュ演算部220iが、記憶部20aからステップS244での演算結果R'とh'とリカバリメッセージm_{rec}'のビット長M'とを読み込む。ハッシュ演算部20iは、署名生成装置210と同一のハッシュ関数$H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^M$を、演算結果R'と値h'とに

依存する値β'に作用させ(式(18))、その演算結果であるM'ビットのハッシュ値u'を記憶部20aに出力して格納する(ステップS246)。なお、β'の構成方法は署名生成装置210でのβの構成方法と同一(Π＝Π'とし、h＝h'とした場合に同一)とする。

[0091] 次に、排他的論理和演算部20jが、記憶部20aから値w'$\in\{0,1\}^{M'}$とハッシュ値u'とを読み込む。排他的論理和演算部20jは、値w'とハッシュ値u'との排他的論理和を算出し(式(10))、その演算結果をリカバリメッセージm_{rec}'$\in\{0,1\}^{M'}$として記憶部20aに出力して格納する(ステップS247)。

[0092] 次に、ハッシュ演算部220kが、記憶部20aから演算結果R'とリカバリメッセージm_{re}'とを読み込む。ハッシュ演算部220kは、署名生成装置210と同一のハッシュ関数$H_{1}^{c}:\{0,1\}^{*}\rightarrow\{0,1\}^{L}$を、演算結果R'とリカバリメッセージ$m_{rec}$'とに依存する値α'に作用させ、その演算結果であるLビットのハッシュ値(式(20))を記憶部20aに出力して格納する(ステップS248)。なお、α'の構成方法は署名生成装置210でのαの構成方法と同一(Π＝Π'とし、m_{rec}＝m_{rec}'とした場合に同一)とする。

その後、第1実施形態のステップS51～S54と同じ処理が実行される(ステップS249～S252)。

[0093] 〔第4実施形態〕

次に、本発明の第4実施形態について説明する。本形態は第3実施形態の変形例である。第4実施形態ではクリアメッセージを用いない。この点が第3実施形態との相違点である。以下では、第1～3実施形態との相違点を中心に説明し、第1～3実施形態と共通する事項については説明を省略する。

[0094] ＜全体構成＞

第1実施形態の署名システム1の署名生成装置10が署名生成装置310に置換され、署名検証装置20が署名検証装置320に置換された構成である。

＜署名生成装置310の構成＞

次に、署名生成装置310の構成を説明する。

[ハードウェア構成]

第1実施形態の署名生成装置10と同じである。

[0095] ［ハードウェアとプログラムとの協働］

署名生成装置310もコンピュータに所定のプログラムが読み込まれることにより構成される。

図18は、このように構成される第4実施形態における署名生成装置310の機能構成を例示したブロック図である。なお、署名生成装置310において署名生成装置10，110，210と共通する部分には図3，図10，図14と同じ符号を付して説明を簡略化する。

図18に示すように、本形態の署名生成装置310は、記憶部10aと、秘密鍵生成部10bと、公開鍵生成部10cと、入力部110dと、ビット長抽出部110eと、任意値生成部10fと、群演算部10gと、ハッシュ演算部210i，210j，110pと、排他的論理和演算部10kと、ビット結合部210mと、整数演算部10qと、通信部110rと、制御部10sと、一時メモリ10tとを有する。

[0096] ＜署名検証装置320の構成＞

次に、署名検証装置320の構成を説明する。

［ハードウェア構成］

第1実施形態の署名検証装置20と同じである。

［ハードウェアとプログラムとの協働］

署名検証装置320もコンピュータに所定のプログラムが読み込まれることにより構成される。図19は、このように構成される第2実施形態の署名検証装置120の機能構成を例示したブロック図である。なお、署名検証装置320において署名検証装置20，120，220と共通する部分には図5，図11，図15と同じ符号を付して説明を簡略化する。

[0097] 図19に示すように、本形態の署名検証装置320は、記憶部20aと、通信部120bと、ビット長抽出部20cと、ハッシュ演算部120d，220i，220kと、群演算部20eと、ビット抽出部220hと、排他的論理和演算部20jと、比較部20lと、出力部20mと、制御部20nと、一時メモリ20pとを有する。

[0098] ＜処理＞

次に、本形態の処理について説明する。

［前処理・鍵生成処理］

第1実施形態と同じである。

［署名生成処理］

次に、第4実施形態の署名生成処理について説明する。

図20は、第4実施形態の署名生成処理を説明するためのフローチャートである。以下、図20に従って本形態の署名生成処理を説明する。

まず、署名生成装置310が、第2実施形態のステップS111～S114と同じ処理を実行し（ステップS311～S314）、次に、第3実施形態のステップS215～S218と同じ処理を実行する（ステップS315～S318）。次に、署名生成装置310が、第2実施形態のステップS121～123と同じ処理を実行する（ステップS319～S321）。

[0099] ［署名検証処理］

次に、第4実施形態の署名検証処理について説明する。

図21は、第4実施形態の署名検証処理を説明するためのフローチャートである。以下、図21に従って本形態の署名検証処理を説明する。

まず、署名検証装置320が、第2実施形態のステップS141～S144と同じ処理を実行し（ステップS341～S344）、次に、第3実施形態のステップS245～252と同じ処理を実行する（ステップS345～S352）。

[0100] 〔署名検証となる理由〕

次に、署名検証装置20，120，220，320の処理が署名検証となる理由について説明する。

＜第1，2実施形態について＞

署名検証装置20，120では、署名$\sigma'=(r', s')$を用い、r'に依存する値γ'に対してハッシュ値$t'=H_3(\gamma')$を演算し（式(14)(23)）、$R'=g^{s'}\cdot y^{t'}\in G$の演算を行い（式(15)）、ハッシュ値$\Pi'=H_0(R')$の演算を行っている（式(16)）。署名$\sigma'$が正規なものであれば$r'=r$及び$s'=s$（$s=k-t\cdot x\in Z$）を満たすため、$\gamma'=\gamma$を満たし、$t'=H_3(\gamma')=H_3(\gamma)=t$を満たし、$y=g^x\in G$を満たすため、$R'=g^{s'}\cdot y^{t'}=g^s\cdot y^t=g^{k-t\cdot x}\cdot g^{t\cdot x}=g^k\in G$となる。そのため、$\Pi'=H_0(R')=H_0(g^k)=\Pi$となる。

[0101] また、署名検証装置20，120では、排他的論理和値$d'=\Pi'(+)r'$を求めている

が(式(17))、署名σ'が正規なものであればr'=rを満たし、r=Π(+)dを満たし、Π'=Πを満たすため、d'=dを満たす。さらに、署名検証装置20, 120では、ハッシュ値Π'と排他的論理和値d'の第1ビット位置のLビットの値h'∈{0, 1}^{L}とに依存する値β'のハッシュ値u'=H_2(β')を求めているが(式(18))、署名σ'が正規なものであればd'=dを満たすため、h'=hを満たし、さらにΠ'=Πを満たすため、β'=βを満たし、u'=uを満たす。

[0102] さらに、署名検証装置20, 120では、排他的論理和値d'の第2ビット位置のM'ビットの値w'∈{0, 1}$^{M'}$とハッシュ値u'との排他的論理和w'(+)u'を算出し、その演算結果をリカバリメッセージm_{rec}'∈{0, 1}$^{M'}$としているが(式(19))、署名σ'が正規なものであればu'=u、M'=M及びd'=dを満たす。この場合、w'=wも満たすため、m_{rec}'=w'(+)u'=w(+)u=m_{rec}(+)u(+)u=m_{rec}を満たす。

[0103] その後、署名検証装置20, 120では、ハッシュ値Π'とリカバリメッセージm_{rec}'とに依存する値α'にハッシュ値H_1を作用させたハッシュ値H_1(α')∈{0, 1}^{L}を求めている(式(20))。署名σ'が正規なものであればΠ'=Π, m_{rec}'=m_{rec}, α'=α, h'=hを満たし、署名生成装置ではh=H_1(α)としていたため、h'=H_1(α')も満たす。つまり、署名σ'が正規なものであればh'=H_1(α')を満たすことがいえる。

[0104] 一方、巡回群Gでの離散対数問題の求解が困難であると仮定すると、秘密鍵xを知らない第三者は公開鍵y=g^x∈Gから秘密鍵xを知ることができず、上記の検証に合格する署名σ'=(r', s')を生成することができない。よって、上記の検証に合格する署名σ'=(r', s')が秘密鍵xを知る者が正規に生成した署名であるといえる。

[0105] <第3, 4実施形態について>

署名検証装置220, 320では、署名σ'=(r', s')を用い、r'に依存する値γ'に対してハッシュ値t'=H_3(γ')を演算し、R'=$g^{s'}$・$y^{t'}$∈Gの演算を行っている。署名σ'が正規なものであればr'=r及びs'=s(s=k−t・x∈Z)を満たすため、γ'=γを満たし、t'=H_3(γ')=H_3(γ)=tを満たし、y=g^x∈Gを満たすため、R'=$g^{s'}$・$y^{t'}$=g^s・y^t=$g^{k-t\cdot x}$・$g^{t\cdot x}$=g^k=Rとなる。

[0106] また、署名検証装置220, 320では、演算結果R'と署名σ'のr'の第1ビット位置のLビットの値h'∈{0, 1}^{L}とに依存する値β'のハッシュ値u'=H_2(β')を求めてい

るが、署名σ'が正規なものであればr'＝rを満たすため、h'＝hを満たし、さらにR'＝Rを満たすため、β'＝βを満たし、u'＝uを満たす。

さらに、署名検証装置220，320では、署名σ'のr'の第2ビット位置のM'ビットの値$w' \in \{0, 1\}^{M'}$とハッシュ値u'との排他的論理和w'(＋)u'を算出し、その演算結果をリカバリメッセージ$m_{rec}' \in \{0, 1\}^{M'}$としているが)、署名σ'が正規なものであればu'＝u、M'＝M及びr'＝rを満たす。この場合、w'＝wも満たすため、m_{rec}'＝w'(＋)u'＝w(＋)u＝m_{rec}(＋)u(＋)u＝m_{rec}を満たす。

[0107] その後、署名検証装置220，320では、演算結果R'とリカバリメッセージm_{rec}'とに依存する値α'にハッシュ値H_1を作用させたハッシュ値$H_1(\alpha') \in \{0, 1\}^L$を求めている。署名σ'が正規なものであればR'＝R，m_{rec}'＝m_{rec}，α'＝α，h'＝hを満たし、署名生成装置ではh＝H_1(α)としていたため、h'＝H_1(α')も満たす。つまり、署名σ'が正規なものであればh'＝H_1(α')を満たすことがいえる。

[0108] 一方、巡回群Gでの離散対数問題の求解が困難であると仮定すると、秘密鍵xを知らない第三者は公開鍵y＝g^x∈Gから秘密鍵xを知ることができず、上記の検証に合格する署名σ'＝(r'，s')を生成することができない。よって、上記の検証に合格する署名σ'＝(r'，s')が秘密鍵xを知る者が正規に生成した署名であるといえる。

[0109] 〔変形例〕

本発明は上述の各実施の形態に限定されるものではない。例えば、第1，2実施形態では、αをΠとm_{rec}とのみに依存する値とし、α'をΠ'とm_{rec}'とのみに依存する値とした。しかし、αをΠとm_{rec}と第三情報とに依存する値とし、α'をΠ'とm_{rec}'と第三情報とに依存する値としてもよい。なお、第三情報としては、クリアメッセージm_{clr}や公開鍵yや群Gを特定するためのパラメータなどを例示できる。βとβ'及びγとγ'についても同様である。これにより、署名検証精度をより向上させることができる。特に、第三情報として群Gを特定するためのパラメータを用いた場合には、不正な群(例えば、離散対数問題が容易であって群演算部20eでの演算結果が正規の巡回群Gでの演算結果と同一となる群)を用いて生成された不正な署名が検証に合格してしまうことを防止できる。

[0110] 同様に、第3，4実施形態では、αをRとm_{rec}とのみに依存する値とし、α'をR'とm_r

$_{ec}$'とのみに依存する値とした。しかし、αをRとm_{rec}と第三情報とに依存する値とし、α'をR'とm_{rec}'と第三情報とに依存する値としてもよい。βとβ'及びγとγ'についても同様である。

[0111] また、各実施形態では、署名生成装置10, 110, 210, 310が鍵生成を行ったが、別の装置が鍵生成を行ってもよい。また、各実施形態では、公開鍵サーバ装置30が公開鍵yを公開したが署名生成装置10, 110, 210, 310が署名検証装置20, 120, 220, 320に公開鍵yを送信する構成であってもよい。また、各処理におけるZ_q（qを法とする完全剰余系）をZ（整数）に置換した構成であってもよい。

また、各実施形態では、署名検証装置20, 120, 220, 320が、署名σ'が具備するr'のビット長とビット長パラメータLとからリカバリメッセージのビット長を算出したが、署名生成装置10, 110, 210, 310が署名検証装置20, 120, 220, 320にリカバリメッセージのビット長を送信する構成であってもよい。

[0112] また、各実施形態では、少なくともリカバリメッセージm_{rec}を署名対象とした。つまり、リカバリメッセージm_{rec}, m_{rec}'のビット長M, M'をそれぞれ1以上とした。しかし、実施形態1, 3において、リカバリメッセージm_{rec}, m_{rec}'をそれぞれヌル値とし、クリアメッセージm_{clr}, m_{clr}'のみを署名対象としてもよい。これは、リカバリメッセージm_{rec}, m_{rec}'のビット長M, M'をそれぞれ0とすることに相当する。また、ビット長M, M'をM≧0の範囲で設定可能な構成としてもよい。これにより、ビット長M, M'の設定に応じて、メッセージリカバリ署名と通常の署名との切り替えが可能となる。なお、リカバリメッセージm_{rec}, m_{rec}'をそれぞれヌル値とし、それぞれのビット長M, M'を0としたことによって行う必要がなくなった処理は省略可能である。また、当該行う必要がなくなった処理を実行する各機能部の処理を停止させることも可能である。

[0113] また、本発明における「ハッシュ関数」とは、あるデータに対し、そのデータを代表する値を算出する関数を意味する。本発明では、SHA－1やMD5等のみならず、例えば、DESやCamelliaなどの共通鍵暗号関数に共通鍵を代入したものをハッシュ関数として用いることもできる。

また、上述の各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行され

てもよく、その他、本発明の趣旨を逸脱しない範囲で適宜変更が可能であることはいうまでもない。

また、上述の構成をコンピュータによって実現する場合、各装置が有すべき機能の処理内容はプログラムによって記述される。そして、このプログラムをコンピュータで実行することにより、上記処理機能がコンピュータ上で実現される。

[0114] この処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、例えば、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリ等どのようなものでもよいが、具体的には、例えば、磁気記録装置として、ハードディスク装置、フレキシブルディスク、磁気テープ等を、光ディスクとして、DVD(Digital Versatile Disc)、DVD－RAM(Random Access Memory)、CD－ROM(Compact Disc Read Only Memory)、CD－R(Recordable)／RW(ReWritable)等を、光磁気記録媒体として、MO(Magneto-Optical disc)等を、半導体メモリとしてEEP－ROM(Electronically Erasable and Programmable-Read Only Memory)等を用いることができる。

[0115] また、このプログラムの流通は、例えば、そのプログラムを記録したDVD、CD－ROM等の可搬型記録媒体を販売、譲渡、貸与等することによって行う。さらに、このプログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することにより、このプログラムを流通させる構成としてもよい。

[0116] このようなプログラムを実行するコンピュータは、例えば、まず、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、一旦、自己の記憶装置に格納する。そして、処理の実行時、このコンピュータは、自己の記録媒体に格納されたプログラムを読み取り、読み取ったプログラムに従った処理を実行する。また、このプログラムの別の実行形態として、コンピュータが可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することとしてもよく、さらに、このコンピュータにサーバコンピュータからプログラムが転送されるたびに、逐次、受け取ったプログラムに従った処理を実行することとしてもよい。また、サーバコンピュータから、このコンピュータへのプログラムの転送は行わず、その実行指示と

結果取得のみによって処理機能を実現する、いわゆるASP(Application Service Provider)型のサービスによって、上述の処理を実行する構成としてもよい。なお、本形態におけるプログラムには、電子計算機による処理の用に供する情報であってプログラムに準ずるもの(コンピュータに対する直接の指令ではないがコンピュータの処理を規定する性質を有するデータ等)を含むものとする。

[0117] また、この形態では、コンピュータ上で所定のプログラムを実行させることにより、本装置を構成することとしたが、これらの処理内容の少なくとも一部をハードウェア的に実現することとしてもよい。

産業上の利用可能性

[0118] 本発明は、電子署名を用いる様々な用途に適用可能である。

請求の範囲

[1] 署名生成装置であって、

整数の秘密鍵xを格納する第1記憶部と、

Mビットのリカバリメッセージ$m_{rec} \in \{0, 1\}^M$を格納する第2記憶部と、

整数の任意値kを生成する任意値生成部と、

位数qの巡回群をGとし、当該巡回群Gの生成元をgとした場合における$R = g^k \in G$を算出し、当該演算結果Rを出力するように構成される群演算部と、

入力値に対してLビット(Lは署名検証装置と共有される正の整数)のハッシュ値を出力するハッシュ関数$H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^L$を、上記演算結果Rとリカバリメッセージ$m_{rec}$とに依存する値αに作用させ、その演算結果であるLビットのハッシュ値$h = H_1(\alpha) \in \{0, 1\}^L$を出力するように構成される第1ハッシュ演算部と、

上記リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がMビットに定まるハッシュ関数$H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^M$を、上記演算結果Rと上記ハッシュ値hとに依存する値βに作用させ、その演算結果であるMビットのハッシュ値$u = H_2(\beta) \in \{0, 1\}^M$を出力するように構成される第2ハッシュ演算部と、

上記リカバリメッセージm_{rec}と上記ハッシュ値uとの排他的論理和$w = m_{rec} (+) u \in \{0, 1\}^M$((+)は排他的論理和演算子)を算出し、当該排他的論理和値wを出力するように構成される第1排他的論理和演算部と、

上記ハッシュ値$h \in \{0, 1\}^L$を第1ビット位置に配置し、上記排他的論理和値$w \in \{0, 1\}^M$を第2ビット位置に配置したL+Mビットのビット結合値$h \mid w \in \{0, 1\}^{L+M}$に依存し、上記ハッシュ値h及び上記排他的論理和値wを復元可能な値rを算出し、当該値rを出力するように構成されるr値演算部と、

入力値に対して整数を出力するハッシュ関数$H_3 : \{0, 1\}^* \rightarrow Z$(整数)を、上記値rに依存する値γに作用させ、その演算結果であるハッシュ値$t = H_3(\gamma) \in Z$を出力するように構成される第3ハッシュ演算部と、

$s = k - t \cdot x \in Z$を算出し、当該演算結果sを出力する整数演算部と、

署名$\sigma = (r, s)$を出力するように構成される署名出力部と、を含む。

[2] 請求項1の署名生成装置であって、

さらに、上記リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がL＋Mビットに定まるハッシュ関数$H_0:\{0,1\}^*\rightarrow\{0,1\}^{L+M}$を上記演算結果Rに作用させ、その演算結果であるL＋Mビットのハッシュ値$\Pi=H_0(R)\in\{0,1\}^{L+M}$を出力するように構成される第4ハッシュ演算部を含み、

上記演算結果Rとリカバリメッセージm_{rec}とに依存する値αは、上記ハッシュ値Πとリカバリメッセージm_{rec}とに依存する値であり、

上記演算結果Rと上記ハッシュ値hとに依存する値βは、上記ハッシュ値Πと上記ハッシュ値hとに依存する値であり、

上記r値演算部は、

上記ハッシュ値$h\in\{0,1\}^L$を第1ビット位置に配置し、上記排他的論理和値$w\in\{0,1\}^M$を第2ビット位置に配置したL＋Mビットのビット結合値$d=h\mid w\in\{0,1\}^{L+M}$を算出し、当該ビット結合値dを出力するように構成されるビット結合部と、

上記ハッシュ値Πと上記ビット結合値dとの排他的論理和$r=\Pi(+)d\in\{0,1\}^{L+M}$を算出し、当該値rを出力するように構成される第2排他的論理和演算部と、を含む。

[3] 請求項1の署名生成装置であって、

上記r値演算部は、

上記ハッシュ値$h\in\{0,1\}^L$を第1ビット位置に配置し、上記排他的論理和値$w\in\{0,1\}^M$を第2ビット位置に配置したL＋Mビットのビット結合値$r=h\mid w\in\{0,1\}^{L+M}$を算出し、当該値rを出力するように構成されるビット結合部を含む。

[4] 請求項1から3の何れかの署名生成装置であって、$M\geqq 1$である。

[5] 請求項1から3の何れかの署名生成装置であって、

Nビットのクリアメッセージ$m_{clr}\in\{0,1\}^N$を格納する第3記憶部をさらに有し、

上記第3ハッシュ演算部は、

上記ハッシュ関数$H_3:\{0,1\}^*\rightarrow Z$を、上記値rと上記クリアメッセージ$m_{clr}$とに依存する値γに作用させ、その演算結果であるハッシュ値$t=H_3(\gamma)\in Z$を出力するように構成され、

上記署名出力部は、

上記署名$\sigma=(r,s)$と上記クリアメッセージm_{clr}とを出力するように構成される。

[6] 請求項5の署名生成装置であって、

Ｍ≧0であり、

Ｍ＝0の場合の上記リカバリメッセージm_{rec}及び上記排他的論理和値wはヌル値であり、

Ｍ＝0の場合には、上記第2記憶部に上記リカバリメッセージm_{rec}を格納する処理と、上記第2ハッシュ演算部の処理と、上記第1排他的論理和演算部の処理と、が停止するように構成される。

[7] 請求項1の署名生成装置であって、

上記$R=g^k \in G$は、楕円曲線上の点gを当該楕円曲線上でk倍する演算である。

[8] 請求項2の署名生成装置であって、

上記$R=g^k \in G$は、楕円曲線上の点gを当該楕円曲線上でk倍する演算であり、

上記ハッシュ関数$H_0:\{0,1\}^* \rightarrow \{0,1\}^{L+M}$を上記演算結果Rに作用させる演算は、楕円曲線上の点である上記演算結果Rを一義的又は限定的に特定する値に、上記ハッシュ関数H_0を作用させる演算である。

[9] 請求項1の署名生成装置であって、

上記$R=g^k \in G$は、$g^x \bmod p$（ただし、gは2以上の整数，$p=2q+1$）の演算である。

[10] 署名検証装置であって、

位数qの巡回群をGとし、当該巡回群Gの生成元をgとした場合における、署名生成装置の秘密鍵$x \in Z$に対応する公開鍵$y=g^x \in G$を格納する第1記憶部と、

署名$\sigma'=(r', s')$の入力を受け付けるように構成される署名入力部と、

上記署名$\sigma'=(r', s')$を格納する第2記憶部と、

上記署名σ'に対応するリカバリメッセージm_{rec}'のビット長M’を格納する第3記憶部と、

入力値に対して整数を出力するハッシュ関数$H_3:\{0,1\}^* \rightarrow Z$（整数）を、上記署名$\sigma'$が有するr’に依存する値$\gamma'$に作用させ、その演算結果であるハッシュ値$t'=H_3(\gamma') \in Z$を出力するように構成される第1ハッシュ演算部と、

$R'=g^{s'} \cdot y^{t'} \in G$の演算を行い、その演算結果R’を出力するように構成される群演

算部と、

上記リカバリメッセージm_{rec}'のビット長M'に応じて出力ビット長がM'ビットに定まるハッシュ関数$H_2:\{0,1\}^*\rightarrow\{0,1\}^{M'}$を、上記演算結果R'とr'の第1ビット位置のLビット(Lは署名生成装置と共有される正の整数)の値$h'\in\{0,1\}^L$とに依存する値β'に作用させ、その演算結果であるM'ビットのハッシュ値$u'=H_2(\beta')\in\{0,1\}^{M'}$を出力するように構成される第2ハッシュ演算部と、

上記値r'の第2ビット位置のM'ビットの値に依存する値$w'\in\{0,1\}^{M'}$と上記ハッシュ値u'との排他的論理和w'(+)u'を算出し、その演算結果をリカバリメッセージ$m_{rec}'\in\{0,1\}^{M'}$として出力するように構成される第1排他的論理和演算部と、

入力値に対してLビットのハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^*\rightarrow\{0,1\}^L$を、上記演算結果R'と上記第1排他的論理和演算部で算出された上記リカバリメッセージm_{rec}'とに依存する値α'に作用させ、その演算結果であるLビットのハッシュ値$H_1(\alpha')\in\{0,1\}^L$を出力するように構成される第3ハッシュ演算部と、

上記Lビットの値h'と上記ハッシュ値$H_1(\alpha')$とを比較し、$h'=H_1(\alpha')$であることを条件に、検証が成功である旨の情報を出力するように構成される比較部と、を含む。

[11] 請求項10の署名検証装置であって、

さらに、上記リカバリメッセージm_{rec}'のビット長M'に応じて出力ビット長がL+M'ビットに定まるハッシュ関数$H_0:\{0,1\}^*\rightarrow\{0,1\}^{L+M'}$を上記演算結果R'に作用させ、その演算結果であるL+M'ビットのハッシュ値$\Pi'=H_0(R')\in\{0,1\}^{L+M'}$を出力するように構成される第4ハッシュ演算部と、

上記ハッシュ値Π'と、上記署名σ'が有するr'との排他的論理和$d'=\Pi'(+)r'\in\{0,1\}^{L+M'}$を算出し、当該排他的論理和値d'を出力するように構成される第2排他的論理和演算部と、を含み、

上記演算結果R'とr'の第1ビット位置のLビットの値$h'\in\{0,1\}^L$とに依存する値β'は、上記ハッシュ値Π'と、上記排他的論理和値d'の第1ビット位置のLビットの値$h'\in\{0,1\}^L$とに依存する値であり、

上記値r'の第2ビット位置のM'ビットの値に依存する値w'は、上記排他的論理和

値d'の第2ビット位置のM'ビットの値であり、

上記演算結果R'と上記第1排他的論理和演算部で算出された上記リカバリメッセージm_{rec}'とに依存する値α'は、上記ハッシュ値Π'と上記第1排他的論理和演算部で算出された上記リカバリメッセージm_{rec}'とに依存する値である。

[12] 請求項10の署名検証装置であって、

上記値r'の第2ビット位置のM'ビットの値に依存する値w'は、上記値r'の第2ビット位置のM'ビットの値である。

[13] 請求項10から12の何れかの署名検証装置であって、M'≧1である。

[14] 請求項10から12の何れかの署名検証装置であって、

上記署名入力部は、

上記署名σ'と上記署名σ'に対応するクリアメッセージm_{clr}'との入力を受け付けるように構成され、

当該署名検証装置は、

上記クリアメッセージm_{clr}'を格納する第4記憶部を有し、

上記第1ハッシュ演算部は、

上記ハッシュ関数$H_3:\{0,1\}^*\rightarrow Z$を、上記署名σ'が有するr'と上記クリアメッセージm_{clr}'とに依存する値γ'に作用させ、その演算結果であるハッシュ値$t'=H_3(\gamma')\in Z$を出力するように構成される。

[15] 請求項10の署名検証装置であって、

上記署名入力部は、

上記署名σ'と上記署名σ'に対応するクリアメッセージm_{clr}'との入力を受け付けるように構成され、

当該署名検証装置は、

上記クリアメッセージm_{clr}'を格納する第4記憶部を有し、

上記第1ハッシュ演算部は、

上記ハッシュ関数$H_3:\{0,1\}^*\rightarrow Z$を、上記署名σ'が有するr'と上記クリアメッセージm_{clr}'とに依存する値γ'に作用させ、その演算結果であるハッシュ値$t'=H_3(\gamma')\in Z$を出力するように構成され、

M'≧0であり、

M'＝0の場合のリカバリメッセージm_{rec}'はヌル値であり、

M'＝0の場合に、上記第2ハッシュ演算部の処理と、上記第1排他的論理和演算部とが停止し、上記第3ハッシュ演算部が、上記ハッシュ関数H_1を、上記演算結果R'に依存する値α'に作用させ、その演算結果であるハッシュ値H_1(α')を出力するように構成される。

[16] 請求項10の署名検証装置であって、

上記公開鍵$y=g^x \in G$は、楕円曲線上の点gを当該楕円曲線上でx倍した点であり、

上記$R'=g^{s'} \cdot y^{t'} \in G$は、楕円曲線上の点gを当該楕円曲線上でs'倍し、上記公開鍵yを当該楕円曲線上でt'倍し、それらの演算結果を当該楕円曲線上で加算する演算である。

[17] 請求項11の署名検証装置であって、

上記公開鍵$y=g^x \in G$は、楕円曲線上の点gを当該楕円曲線上でx倍した点であり、

上記$R'=g^{s'} \cdot y^{t'} \in G$は、楕円曲線上の点gを当該楕円曲線上でs'倍し、上記公開鍵yを当該楕円曲線上でt'倍し、それらの演算結果を当該楕円曲線上で加算する演算であり、

上記ハッシュ関数$H_0:\{0, 1\}^* \rightarrow \{0, 1\}^{L+M'}$を上記演算結果R'に作用させる演算は、楕円曲線E上の点である上記演算結果R'を一義的又は限定的に特定する値に、上記ハッシュ関数H_0を作用させる演算である。

[18] 請求項10の署名検証装置であって、

上記公開鍵$y=g^x \in G$は、$g^x \bmod p$(ただし、gは2以上の整数, p＝2q＋1)であり、

上記$R'=g^{s'} \cdot y^{t'} \in G$は、$g^{s'} \cdot y^{t'} \bmod p$の演算である。

[19] 署名生成装置が実行する署名生成方法であって、

第1記憶部に整数の秘密鍵xが格納され、第2記憶部にMビットのリカバリメッセージ$m_{rec} \in \{0, 1\}^M$が格納されており、

上記方法は、

(a)整数の任意値kを生成するステップと、

(b)位数qの巡回群をGとし、当該巡回群Gの生成元をgとした場合における$R=g^k \in G$を算出し、当該演算結果Rを出力するステップと、

(c)入力値に対してLビット(Lは署名検証装置と共有される正の整数)のハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^* \rightarrow \{0,1\}^L$を、上記演算結果Rとリカバリメッセージ$m_{rec}$とに依存する値αに作用させ、その演算結果であるLビットのハッシュ値$h=H_1(\alpha) \in \{0,1\}^L$を出力するステップと、

(d)上記リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がMビットに定まるハッシュ関数$H_2:\{0,1\}^* \rightarrow \{0,1\}^M$を、上記演算結果Rと上記ハッシュ値hとに依存する値βに作用させ、その演算結果であるMビットのハッシュ値$u=H_2(\beta) \in \{0,1\}^M$を出力するステップと、

(e)上記リカバリメッセージm_{rec}と上記ハッシュ値uとの排他的論理和$w=m_{rec}(+)u \in \{0,1\}^M$((+)は排他的論理和演算子)を算出し、当該排他的論理和値wを出力するステップと、

(f)上記ハッシュ値$h \in \{0,1\}^L$を第1ビット位置に配置し、上記排他的論理和値$w \in \{0,1\}^M$を第2ビット位置に配置したL+Mビットのビット結合値$h \mid w \in \{0,1\}^{L+M}$に依存し、上記ハッシュ値h及び上記排他的論理和値wを復元可能な値rを算出し、当該値rを出力するステップと、

(g)入力値に対して整数を出力するハッシュ関数$H_3:\{0,1\}^* \rightarrow Z$(整数)を、上記値rに依存する値γに作用させ、その演算結果であるハッシュ値$t=H_3(\gamma) \in Z$を出力するステップと、

(h)$s=k-t \cdot x \in Z$を算出し、当該演算結果sを出力するステップと、

(i)署名$\sigma=(r,s)$を出力するステップと、を含む。

[20] 請求項19の署名生成方法であって、

さらに、上記リカバリメッセージm_{rec}のビット長Mに応じて出力ビット長がL+Mビットに定まるハッシュ関数$H_0:\{0,1\}^* \rightarrow \{0,1\}^{L+M}$を上記演算結果Rに作用させ、その演算結果であるL+Mビットのハッシュ値$\Pi=H_0(R) \in \{0,1\}^{L+M}$を出力するステップを含み、

上記演算結果Rとリカバリメッセージm_{rec}とに依存する値αは、上記ハッシュ値Πとリカバリメッセージm_{rec}とに依存する値であり、

上記演算結果Rと上記ハッシュ値hとに依存する値βは、上記ハッシュ値Πと上記ハッシュ値hとに依存する値であり、

上記ステップ(f)は、

(f－1)上記ハッシュ値$h\in\{0,1\}^{L}$を第1ビット位置に配置し、上記排他的論理和値$w\in\{0,1\}^{M}$を第2ビット位置に配置したL＋Mビットのビット結合値$d=h\mid w\in\{0,1\}^{L+M}$を算出し、当該ビット結合値dを出力するステップと、

(f－2)上記ハッシュ値Πと上記ビット結合値dとの排他的論理和$r=\Pi(+)d\in\{0,1\}^{L+M}$を算出し、当該値rを出力するステップと、を含む。

[21] 請求項19の署名生成装置であって、

上記ステップ(f)は、

上記ハッシュ値$h\in\{0,1\}^{L}$を第1ビット位置に配置し、上記排他的論理和値$w\in\{0,1\}^{M}$を第2ビット位置に配置したL＋Mビットのビット結合値$r=h\mid w\in\{0,1\}^{L+M}$を算出し、当該値rを出力するステップを含む。

[22] 署名検証装置が実行する署名検証方法であって、

第1記憶部に、位数qの巡回群をGとし、当該巡回群Gの生成元をgとした場合における、署名生成装置の秘密鍵$x\in Z$に対応する公開鍵$y=g^{x}\in G$が格納されており、

上記方法は、

(a)署名σ'＝(r', s')の入力を受け付けるステップと、

(b)第2記憶部に、上記署名σ'＝(r', s')を格納するステップと、

(c)第3記憶部に、上記署名σ'に対応するリカバリメッセージm_{rec}'のビット長M'を格納するステップと、

(d)入力値に対して整数を出力するハッシュ関数$H_3:\{0,1\}^{*}\rightarrow Z$(整数)を、上記署名σ'が有するr'に依存する値γ'に作用させ、その演算結果であるハッシュ値t'＝H_3(γ')∈Zを出力するステップと、

(e)$R'=g^{s'}\cdot y^{t'}\in G$の演算を行い、その演算結果R'を出力するステップと、

(f)上記リカバリメッセージm_{rec}'のビット長M'に応じて出力ビット長がM'ビットに定

まるハッシュ関数$H_2:\{0,1\}^*\rightarrow\{0,1\}^{M'}$を、上記演算結果R'とr'の第1ビット位置のLビット(Lは署名生成装置と共有される正の整数)の値h'$\in\{0,1\}^L$とに依存する値β'に作用させ、その演算結果であるM'ビットのハッシュ値u'$=H_2(\beta')\in\{0,1\}^{M'}$を出力するステップと、

(g)上記値r'の第2ビット位置のM'ビットの値に依存する値w'$\in\{0,1\}^{M'}$と上記ハッシュ値u'との排他的論理和w'(+)u'を算出し、その演算結果をリカバリメッセージ$m_{rec}'\in\{0,1\}^{M'}$として出力するステップと、

(h)入力値に対してLビットのハッシュ値を出力するハッシュ関数$H_1:\{0,1\}^*\rightarrow\{0,1\}^L$を、上記演算結果R'と上記ステップ(g)で算出された上記リカバリメッセージm_{rec}'とに依存する値α'に作用させ、その演算結果であるLビットのハッシュ値$H_1(\alpha')\in\{0,1\}^L$を出力するステップと、

(i)上記Lビットの値h'と上記ハッシュ値$H_1(\alpha')$とを比較し、h'$=H_1(\alpha')$であることを条件に、検証が成功である旨の情報を出力するステップと、を含む。

[23] 請求項22の署名検証方法であって、

さらに、上記リカバリメッセージm_{rec}'のビット長M'に応じて出力ビット長がL+M'ビット(Lは署名生成装置と共有される正の整数)に定まるハッシュ関数$H_0:\{0,1\}^*\rightarrow\{0,1\}^{L+M'}$を上記演算結果R'に作用させ、その演算結果であるL+M'ビットのハッシュ値$\Pi'=H_0(R')\in\{0,1\}^{L+M'}$を出力するステップと、

上記ハッシュ値Π'と上記署名σ'が有するr'との排他的論理和d'$=\Pi'(+)r'\in\{0,1\}^{L+M'}$を算出し、当該排他的論理和値d'を出力するステップと、を含み、

上記演算結果R'とr'の第1ビット位置のLビットの値h'$\in\{0,1\}^L$とに依存する値β'は、上記ハッシュ値Π'と、上記排他的論理和値d'の第1ビット位置のLビットの値h'$\in\{0,1\}^L$とに依存する値であり、

上記値r'の第2ビット位置のM'ビットの値に依存する値w'は、上記排他的論理和値d'の第2ビット位置のM'ビットの値であり、

上記演算結果R'と上記ステップ(g)で算出された上記リカバリメッセージm_{rec}'とに依存する値α'は、上記ハッシュ値Π'と上記ステップ(g)で算出された上記リカバリメッセージm_{rec}'とに依存する値である。

[24] 請求項22の署名検証方法であって、

上記値r'の第2ビット位置のM'ビットの値に依存する値w'は、上記値r'の第2ビット位置のM'ビットの値である。

[25] 請求項1の署名生成装置としてコンピュータを機能させるためのプログラム。

[26] 請求項10の署名検証装置としてコンピュータを機能させるためのプログラム。

[図1]

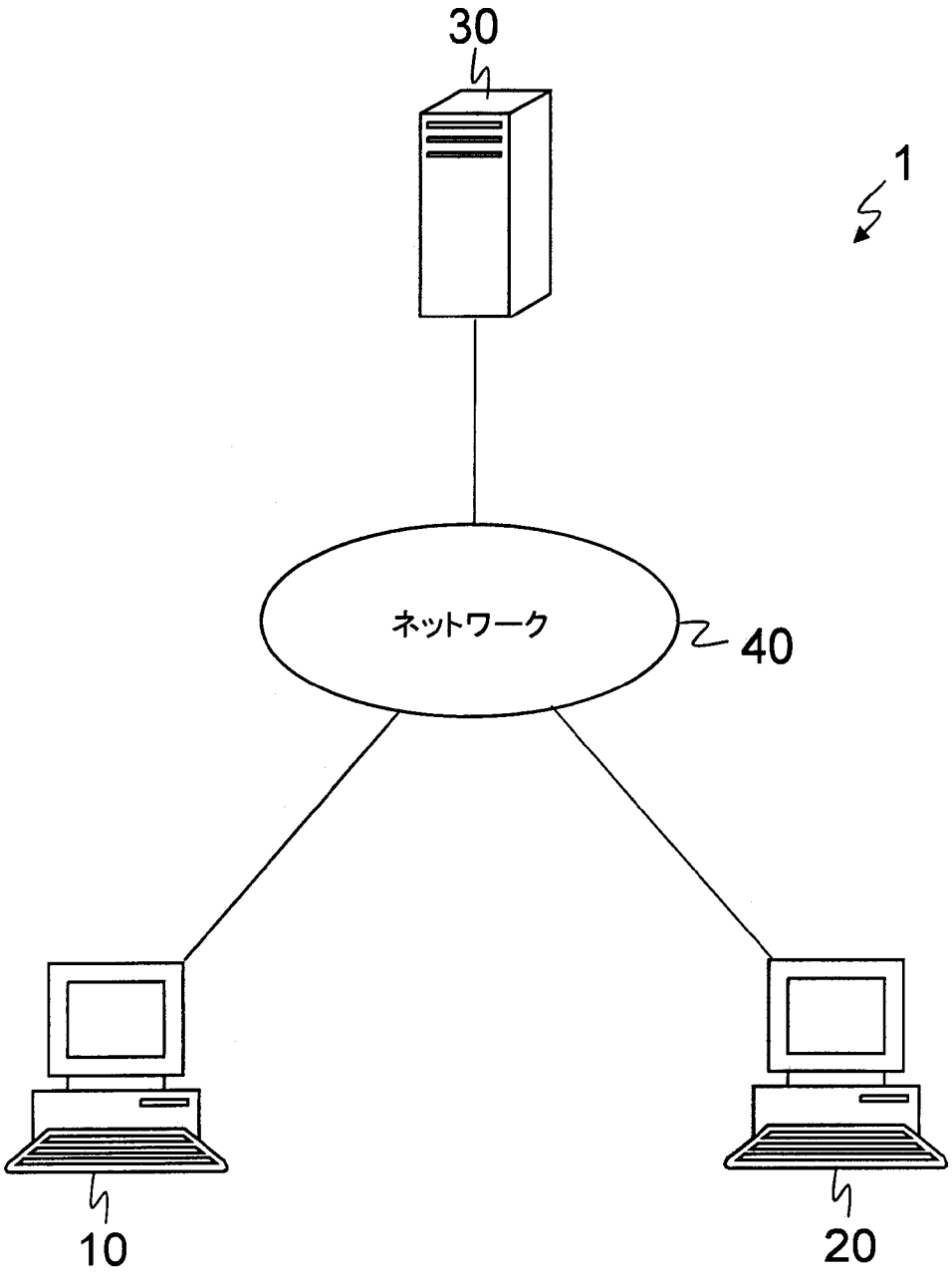


図1

[図2]

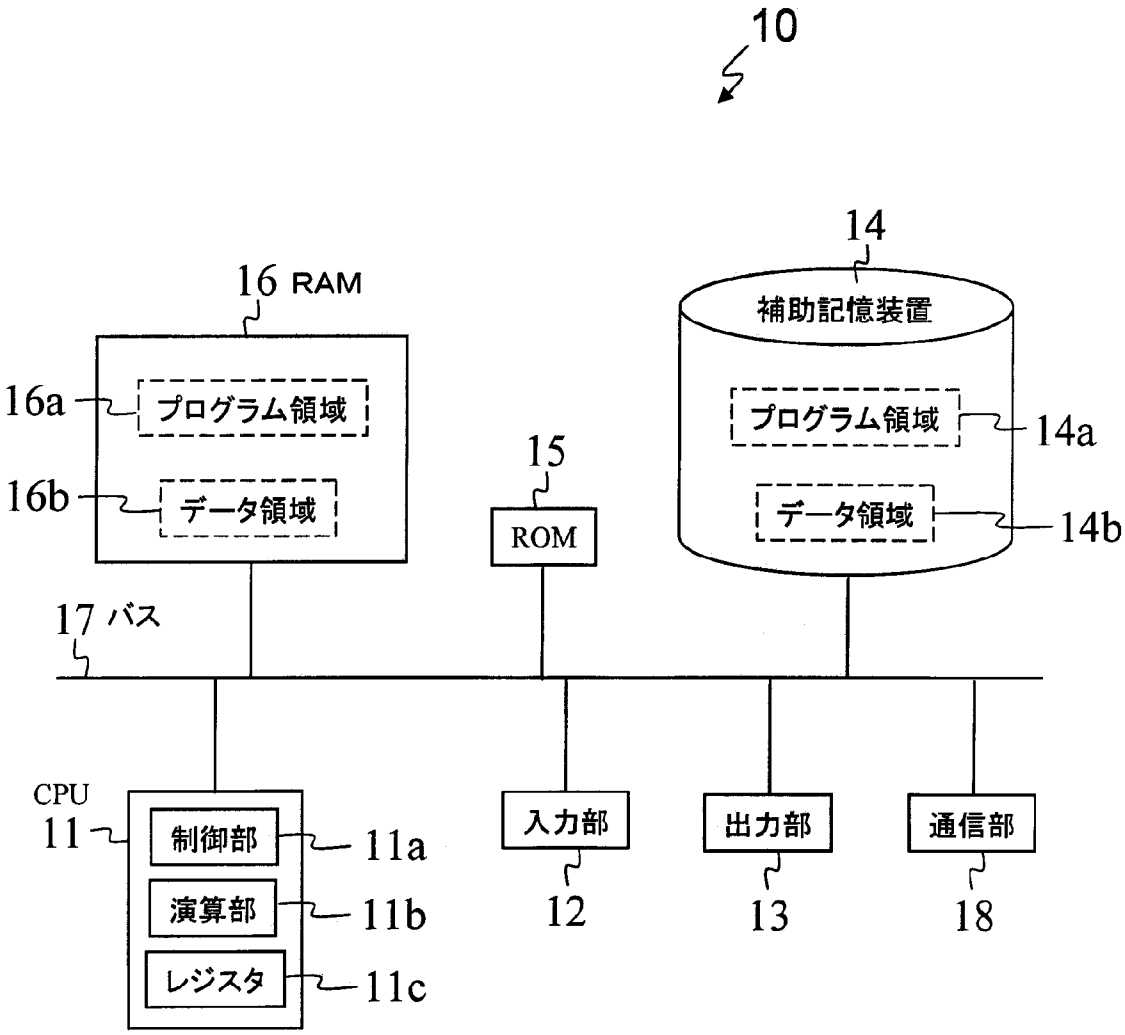

10
14
補助記憶装置
16 RAM
プログラム領域
14a
16a
プログラム領域
15
16b
データ領域
ROM
データ領域
14b
17 バス
CPU
11
制御部
11a
演算部
11b
レジスタ
11c
入力部
12
出力部
13
通信部
18

図2

[図3]

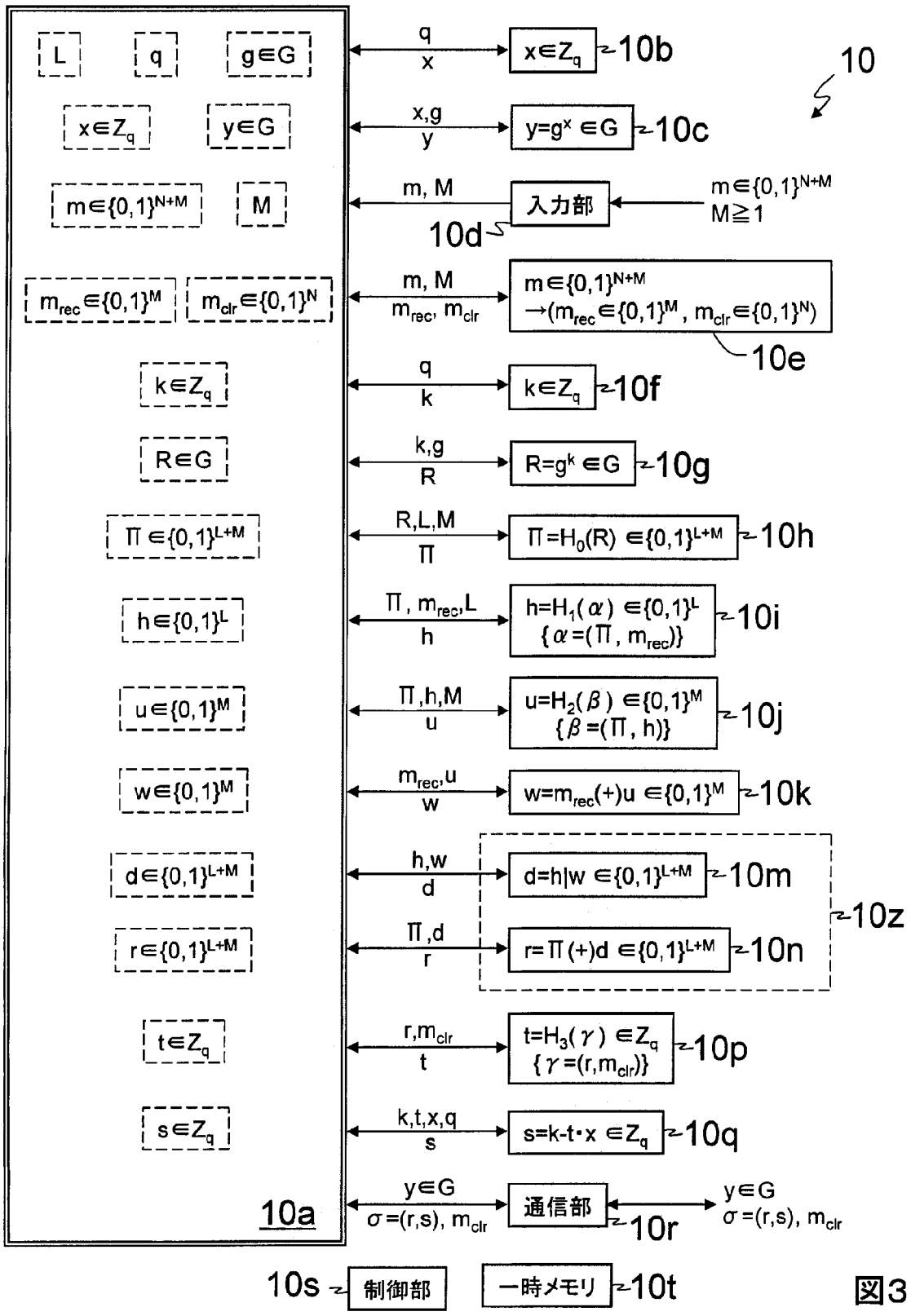

10
L
q
g∈G
x∈Z_q
y∈G
m∈{0,1}^{N+M}
M
m_rec∈{0,1}^M
m_clr∈{0,1}^N
k∈Z_q
R∈G
Π∈{0,1}^{L+M}
h∈{0,1}^L
u∈{0,1}^M
w∈{0,1}^M
d∈{0,1}^{L+M}
r∈{0,1}^{L+M}
t∈Z_q
s∈Z_q
10a
q / x
x∈Z_q
10b
x,g / y
y=g^x ∈G
10c
m, M
入力部
m∈{0,1}^{N+M}
M≧1
10d
m, M / m_rec, m_clr
m∈{0,1}^{N+M} →(m_rec∈{0,1}^M, m_clr∈{0,1}^N)
10e
q / k
k∈Z_q
10f
k,g / R
R=g^k ∈G
10g
R,L,M / Π
Π=H_0(R) ∈{0,1}^{L+M}
10h
Π, m_rec,L / h
h=H_1(α) ∈{0,1}^L {α=(Π, m_rec)}
10i
Π,h,M / u
u=H_2(β) ∈{0,1}^M {β=(Π, h)}
10j
m_rec,u / w
w=m_rec(+)u ∈{0,1}^M
10k
h,w / d
d=h|w ∈{0,1}^{L+M}
10m
Π,d / r
r=Π(+)d ∈{0,1}^{L+M}
10n
10z
r,m_clr / t
t=H_3(γ) ∈Z_q {γ=(r,m_clr)}
10p
k,t,x,q / s
s=k-t・x ∈Z_q
10q
y∈G / σ=(r,s), m_clr
通信部
10r
y∈G σ=(r,s), m_clr
10s
制御部
一時メモリ
10t

図3

[図4]

図4A

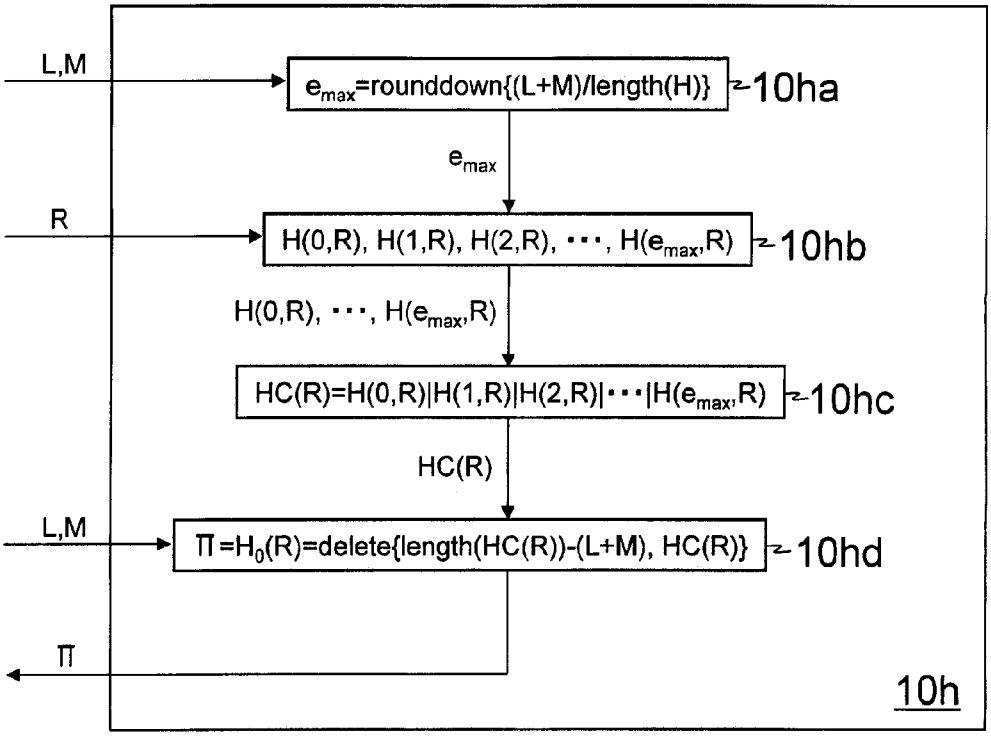
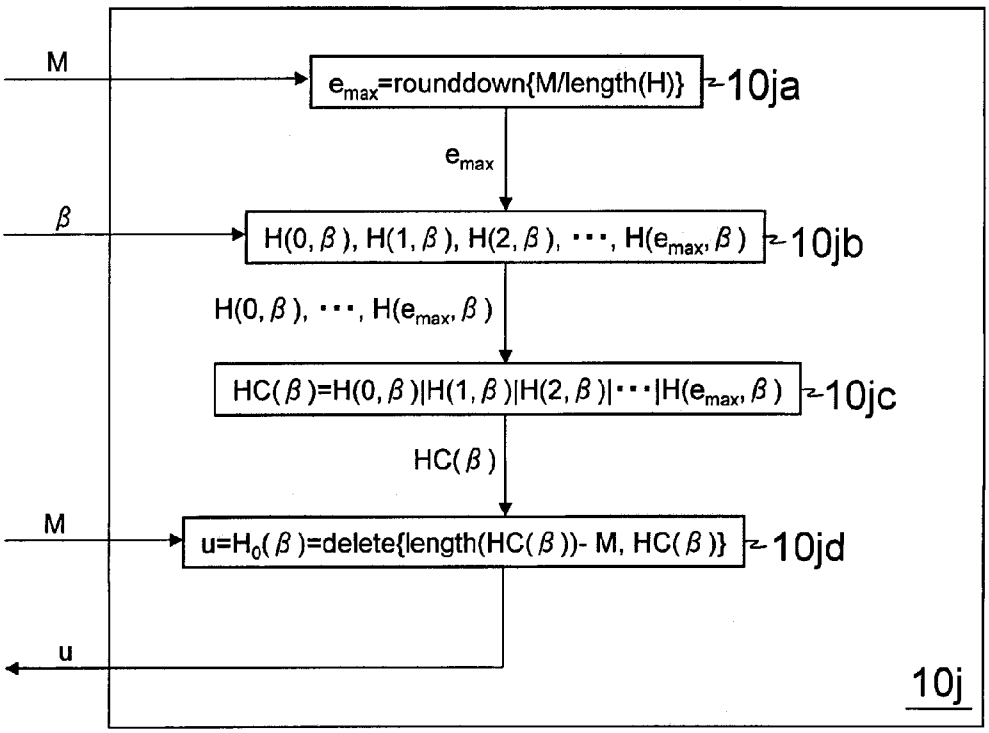

L,M
e_{max}=rounddown{(L+M)/length(H)}
10ha
e_{max}
R
H(0,R), H(1,R), H(2,R), ・・・, H(e_{max},R)
10hb
H(0,R), ・・・, H(e_{max},R)
HC(R)=H(0,R)|H(1,R)|H(2,R)|・・・|H(e_{max},R)
10hc
HC(R)
L,M
Π=H_0(R)=delete{length(HC(R))-(L+M), HC(R)}
10hd
Π
10h

図4B


M
e_{max}=rounddown{M/length(H)}
10ja
e_{max}
β
H(0,β), H(1,β), H(2,β), ・・・, H(e_{max},β)
10jb
H(0,β), ・・・, H(e_{max},β)
HC(β)=H(0,β)|H(1,β)|H(2,β)|・・・|H(e_{max},β)
10jc
HC(β)
M
u=H_0(β)=delete{length(HC(β))- M, HC(β)}
10jd
u
10j

[図5]

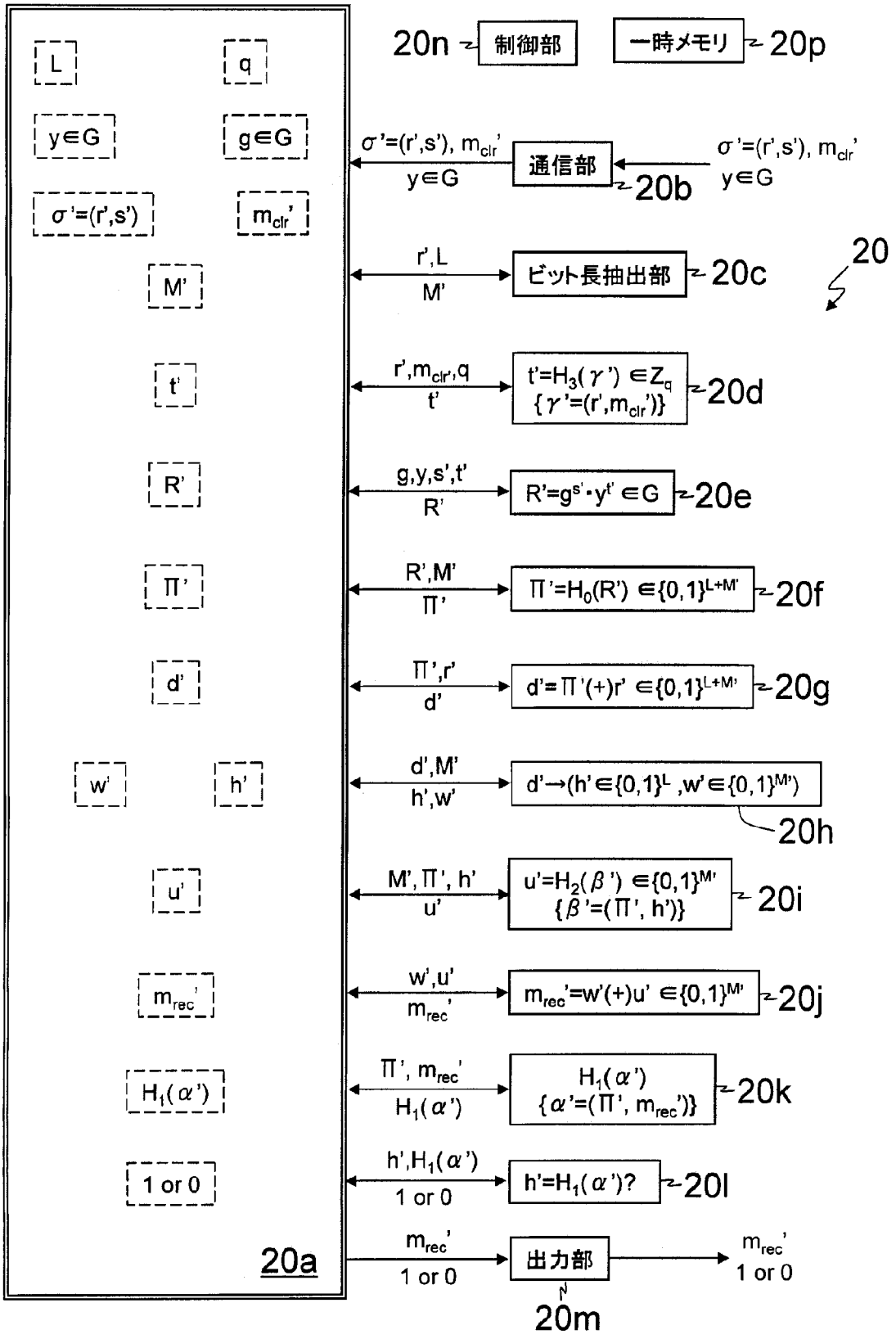

20n
制御部
一時メモリ
20p
L
q
y∈G
g∈G
σ'=(r',s')
m_clr'
σ'=(r',s'), m_clr'
y∈G
通信部
20b
σ'=(r',s'), m_clr'
y∈G
r',L
M'
ビット長抽出部
20c
20
M'
t'
r',m_clr',q
t'
t'=H3(γ') ∈Zq
{γ'=(r',m_clr')}
20d
R'
g,y,s',t'
R'
R'=g^s'・y^t' ∈G
20e
Π'
R',M'
Π'
Π'=H0(R') ∈{0,1}^L+M'
20f
d'
Π',r'
d'
d'=Π'(+)r' ∈{0,1}^L+M'
20g
w'
h'
d',M'
h',w'
d'→(h'∈{0,1}^L ,w'∈{0,1}^M')
20h
u'
M',Π', h'
u'
u'=H2(β') ∈{0,1}^M'
{β'=(Π', h')}
20i
m_rec'
w',u'
m_rec'
m_rec'=w'(+)u' ∈{0,1}^M'
20j
H1(α')
Π', m_rec'
H1(α')
H1(α')
{α'=(Π', m_rec')}
20k
1 or 0
h',H1(α')
1 or 0
h'=H1(α')?
20l
20a
m_rec'
1 or 0
出力部
m_rec'
1 or 0
20m

図5

[図6]

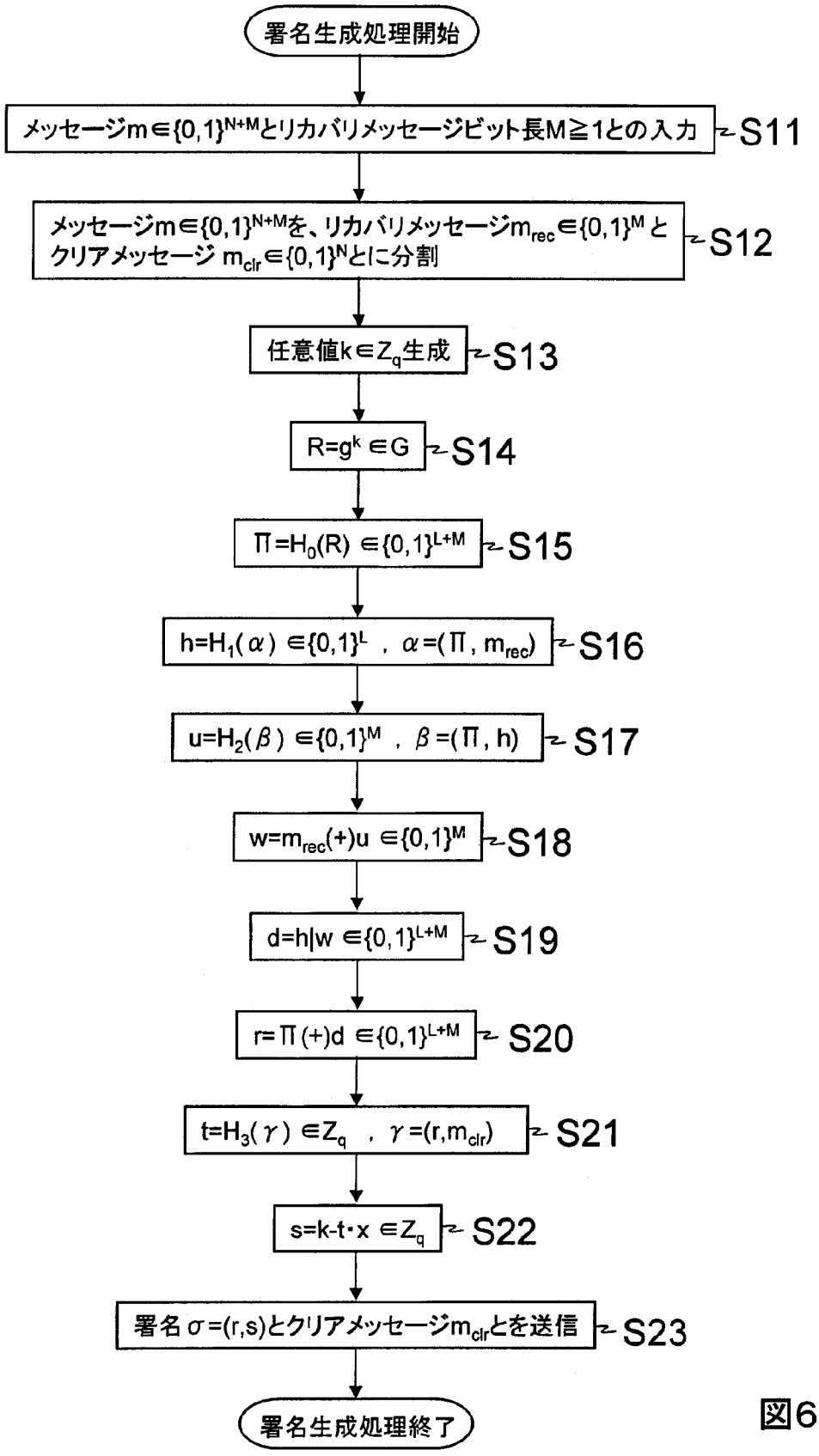

署名生成処理開始
メッセージm∈{0,1}^{N+M}とリカバリメッセージビット長M≧1との入力
S11
メッセージm∈{0,1}^{N+M}を、リカバリメッセージm_rec∈{0,1}^M とクリアメッセージ m_clr∈{0,1}^Nとに分割
S12
任意値k∈Z_q生成
S13
R=g^k ∈G
S14
Π=H_0(R) ∈{0,1}^{L+M}
S15
h=H_1(α) ∈{0,1}^L , α=(Π, m_rec)
S16
u=H_2(β) ∈{0,1}^M , β=(Π, h)
S17
w=m_rec(+)u ∈{0,1}^M
S18
d=h|w ∈{0,1}^{L+M}
S19
r=Π(+)d ∈{0,1}^{L+M}
S20
t=H_3(γ) ∈Z_q , γ=(r,m_clr)
S21
s=k-t･x ∈Z_q
S22
署名 σ=(r,s)とクリアメッセージm_clrとを送信
S23
署名生成処理終了

図6

[図7]

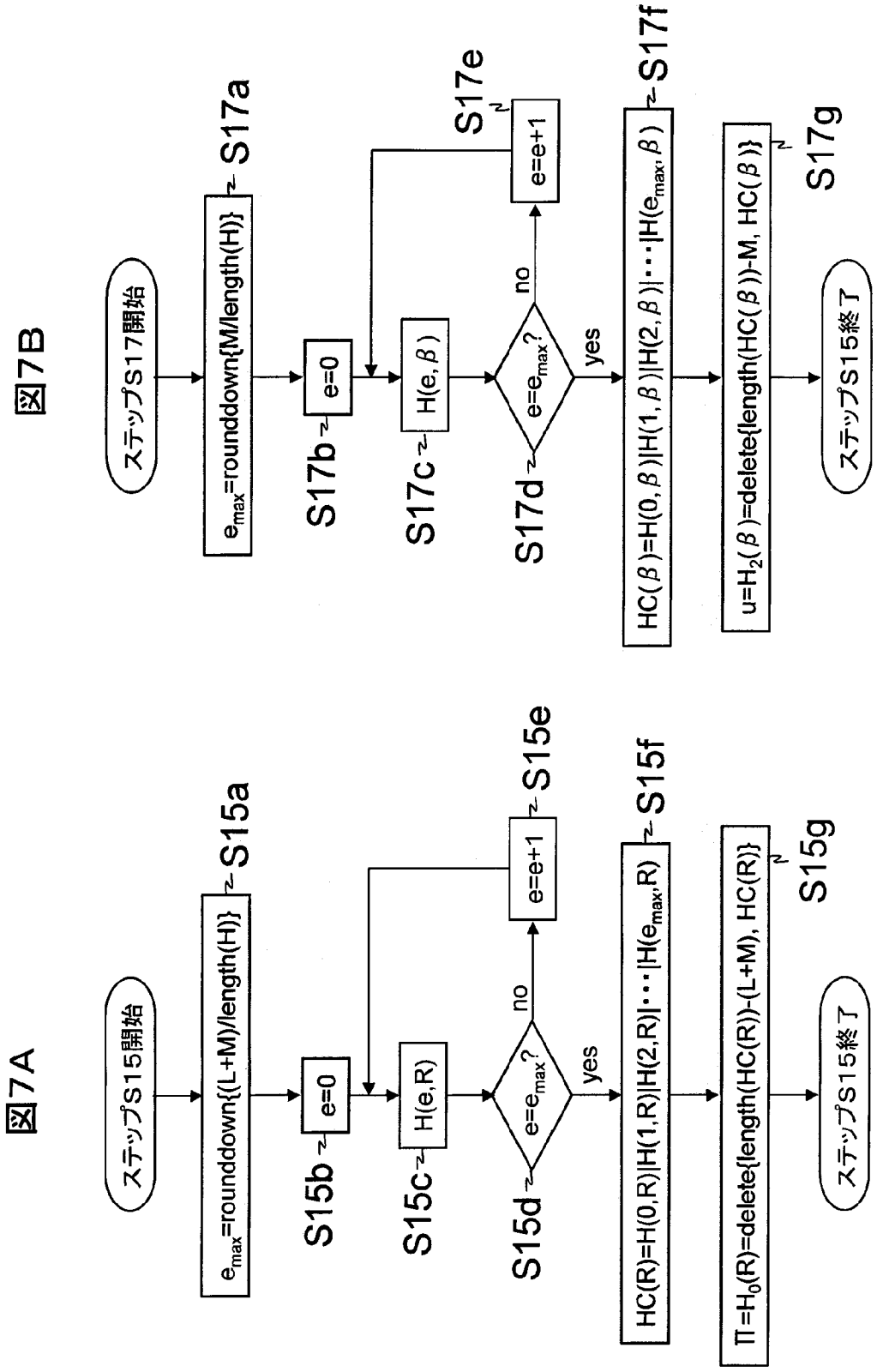


図7A
ステップS15開始
e_{max}=rounddown{(L+M)/length(H)}
S15a
S15b
e=0
S15c
H(e,R)
S15d
$e=e_{max}$?
no
yes
e=e+1
S15e
HC(R)=H(0,R)|H(1,R)|H(2,R)|・・・|H(e_{max},R)
S15f
Π=H_0(R)=delete{length(HC(R))-(L+M), HC(R)}
S15g
ステップS15終了
図7B
ステップS17開始
e_{max}=rounddown{M/length(H)}
S17a
S17b
e=0
S17c
H(e, β)
S17d
$e=e_{max}$?
no
yes
e=e+1
S17e
HC(β)=H(0, β)|H(1, β)|H(2, β)|・・・|H(e_{max}, β)
S17f
u=H_2(β)=delete{length(HC(β))-M, HC(β)}
S17g
ステップS15終了

[図8]

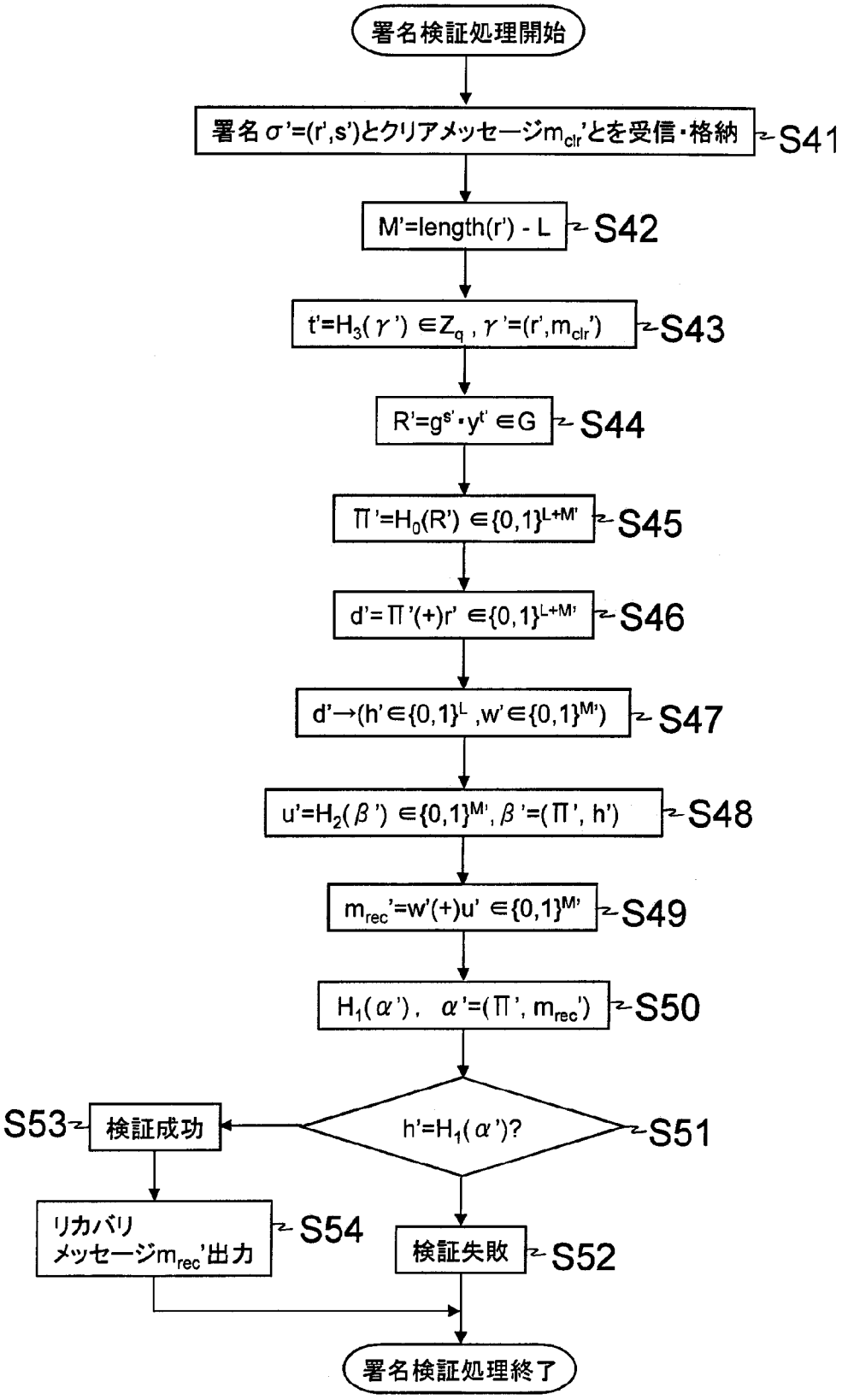

署名検証処理開始
署名σ'=(r',s')とクリアメッセージm_{clr}'とを受信・格納
S41
M'=length(r') - L
S42
t'=$H_3(\gamma')\in Z_q$, $\gamma'=(r',m_{clr}')$
S43
R'=$g^{s'}\cdot y^{t'}\in G$
S44
Π'=$H_0(R')\in\{0,1\}^{L+M'}$
S45
d'=Π'(+)r' $\in\{0,1\}^{L+M'}$
S46
d'→(h'$\in\{0,1\}^{L}$,w'$\in\{0,1\}^{M'}$)
S47
u'=$H_2(\beta')\in\{0,1\}^{M'}$, $\beta'=(\Pi', h')$
S48
m_{rec}'=w'(+)u' $\in\{0,1\}^{M'}$
S49
$H_1(\alpha')$, $\alpha'=(\Pi', m_{rec}')$
S50
h'=$H_1(\alpha')$?
S51
S53
検証成功
リカバリ
メッセージm_{rec}'出力
S54
検証失敗
S52
署名検証処理終了

図8

[図9]

図9A

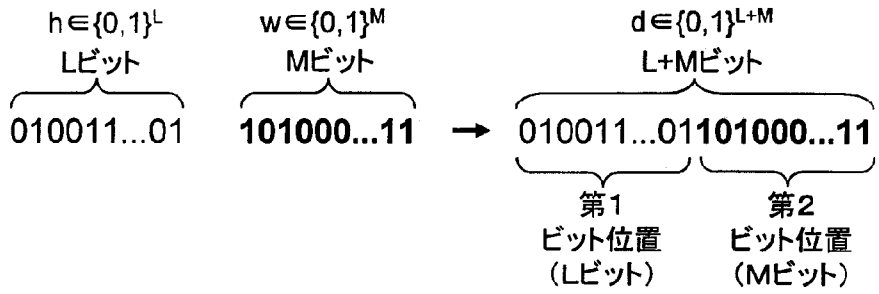


図9B

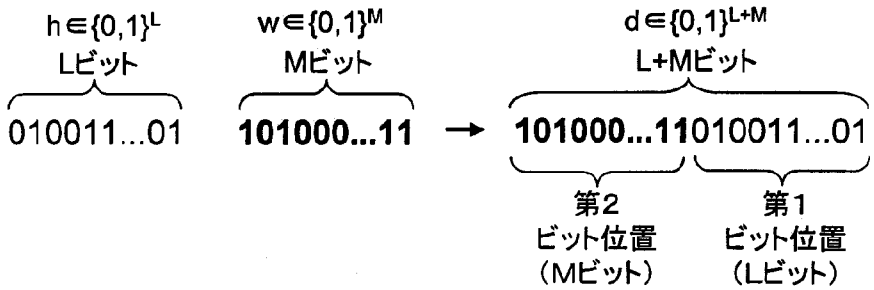
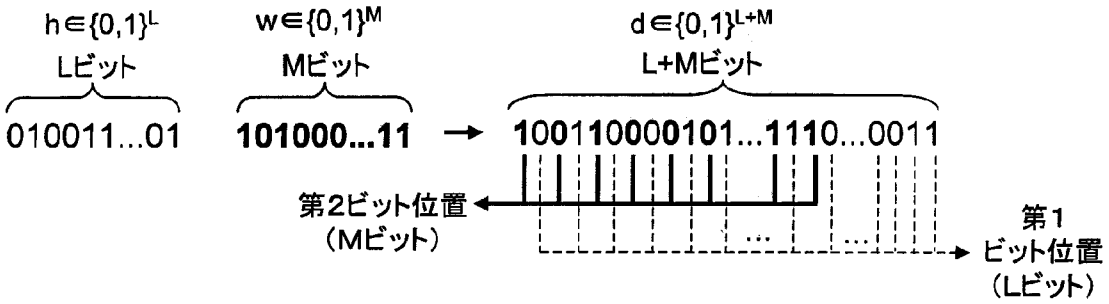


図9C



[図10]

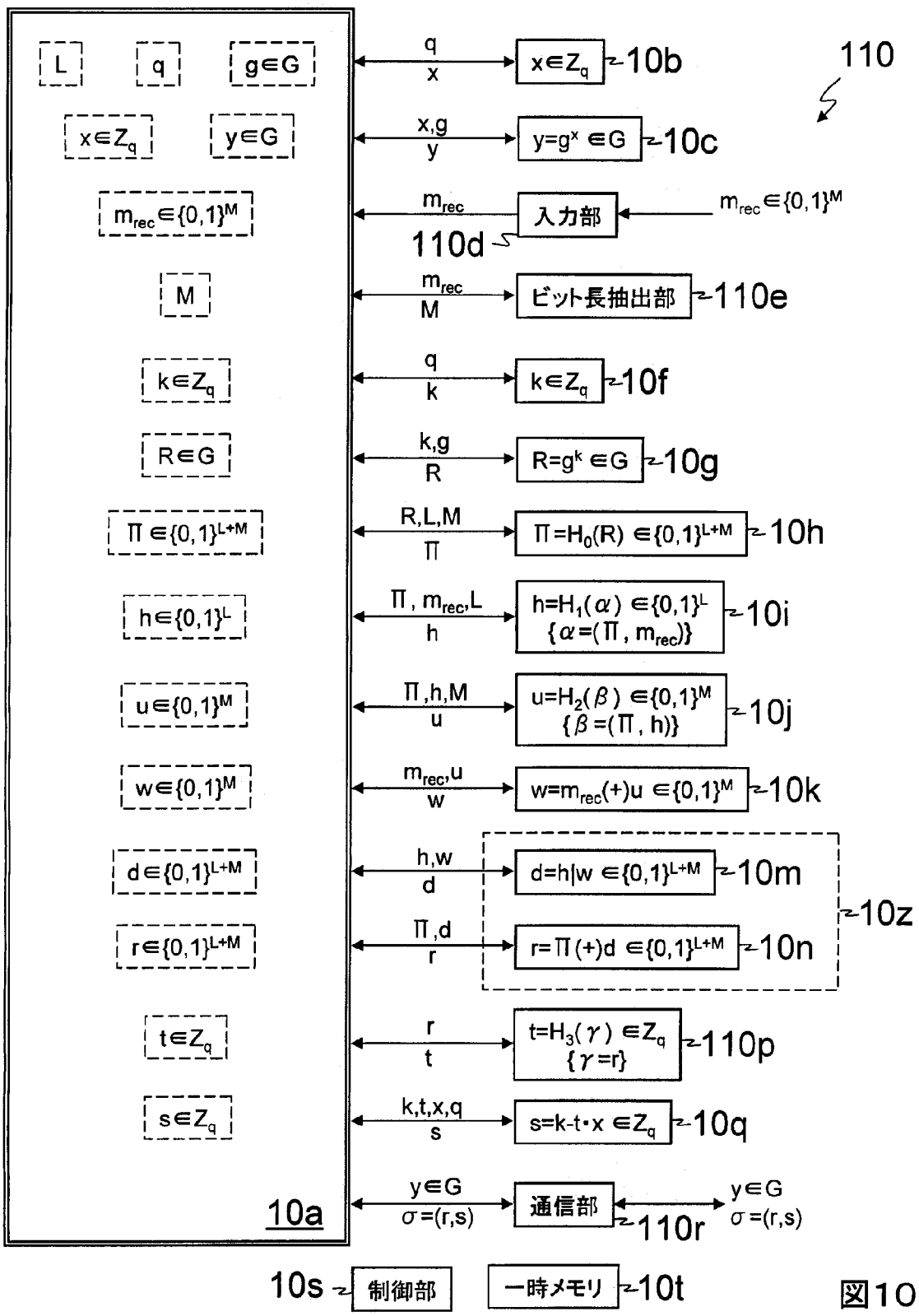

L
q
g∈G
x∈Zq
y∈G
mrec∈{0,1}M
M
k∈Zq
R∈G
Π∈{0,1}L+M
h∈{0,1}L
u∈{0,1}M
w∈{0,1}M
d∈{0,1}L+M
r∈{0,1}L+M
t∈Zq
s∈Zq
10a
q
x
x∈Zq
10b
110
x,g
y
y=gx ∈G
10c
mrec
入力部
mrec∈{0,1}M
110d
mrec
M
ビット長抽出部
110e
q
k
k∈Zq
10f
k,g
R
R=gk ∈G
10g
R,L,M
Π
Π=H0(R) ∈{0,1}L+M
10h
Π, mrec,L
h
h=H1(α) ∈{0,1}L
{α=(Π, mrec)}
10i
Π,h,M
u
u=H2(β) ∈{0,1}M
{β=(Π, h)}
10j
mrec,u
w
w=mrec(+)u ∈{0,1}M
10k
h,w
d
d=h|w ∈{0,1}L+M
10m
Π,d
r
r=Π(+)d ∈{0,1}L+M
10n
10z
r
t
t=H3(γ) ∈Zq
{γ=r}
110p
k,t,x,q
s
s=k-t・x ∈Zq
10q
y∈G
σ=(r,s)
通信部
110r
y∈G
σ=(r,s)
10s
制御部
一時メモリ
10t

図10

[図11]

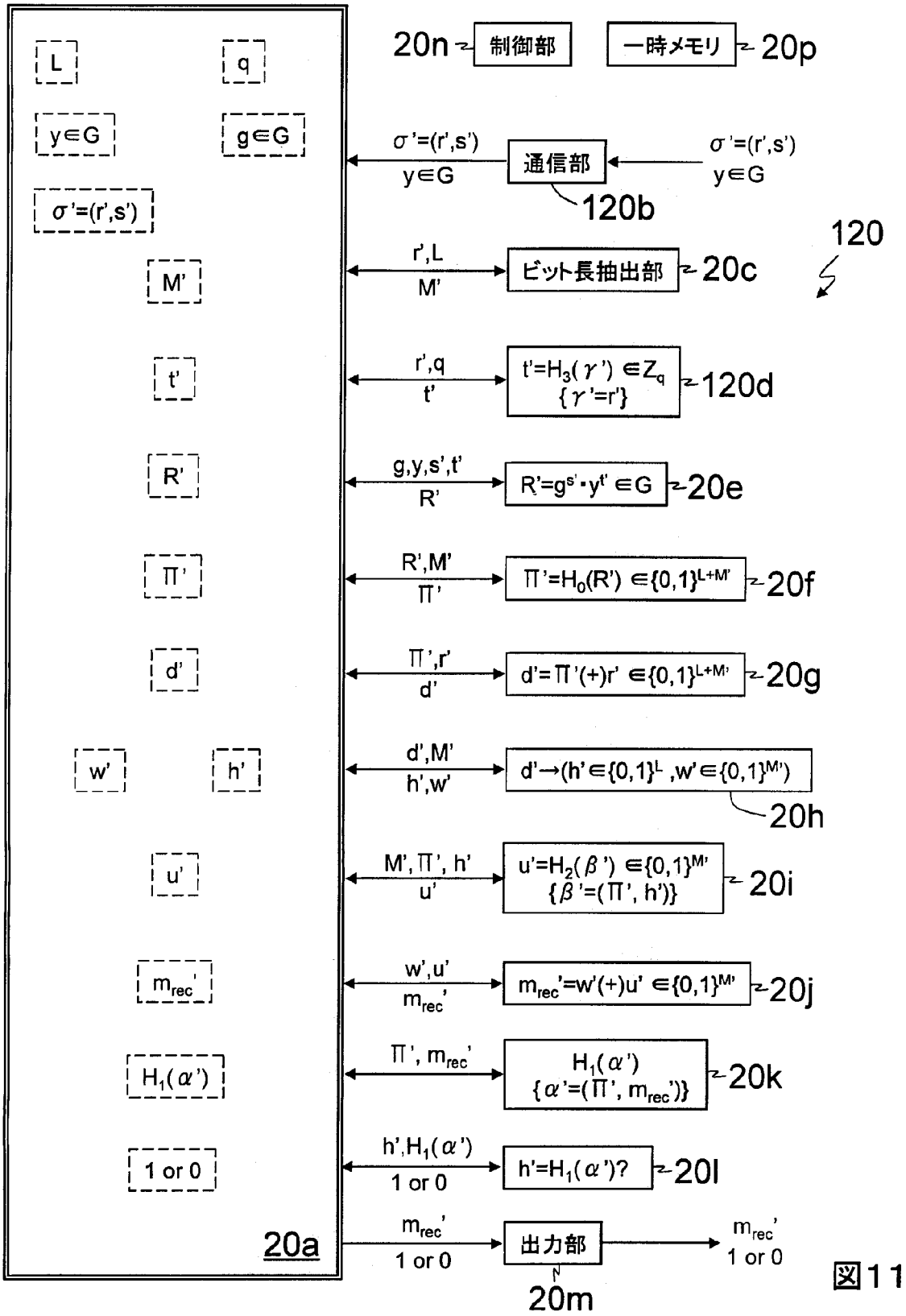

20n
制御部
一時メモリ
20p
L
q
y∈G
g∈G
σ'=(r',s')
M'
t'
R'
Π'
d'
w'
h'
u'
m_rec'
H1(α')
1 or 0
20a
σ'=(r',s')
y∈G
通信部
σ'=(r',s')
y∈G
120b
120
r',L
M'
ビット長抽出部
20c
r',q
t'
t'=H3(γ') ∈Zq
{γ'=r'}
120d
g,y,s',t'
R'
R'=gs'・yt' ∈G
20e
R',M'
Π'
Π'=H0(R') ∈{0,1}L+M'
20f
Π',r'
d'
d'=Π'(+)r' ∈{0,1}L+M'
20g
d',M'
h',w'
d'→(h'∈{0,1}L ,w'∈{0,1}M')
20h
M',Π', h'
u'
u'=H2(β') ∈{0,1}M'
{β'=(Π', h')}
20i
w',u'
m_rec'
m_rec'=w'(+)u' ∈{0,1}M'
20j
Π', m_rec'
H1(α')
{α'=(Π', m_rec')}
20k
h',H1(α')
1 or 0
h'=H1(α')?
20l
m_rec'
1 or 0
出力部
20m
m_rec'
1 or 0

図11

[図12]

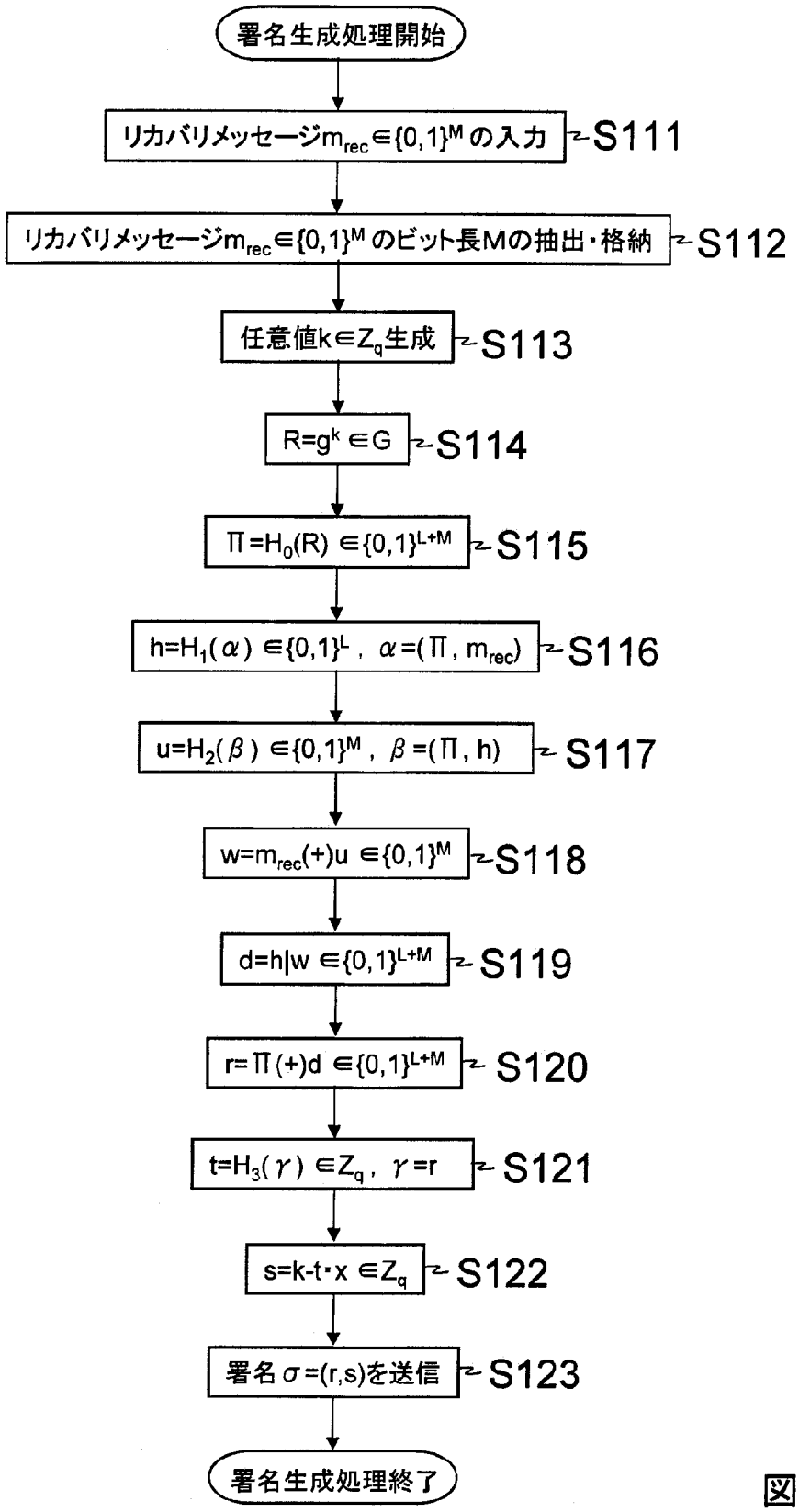

署名生成処理開始
リカバリメッセージ$m_{rec}\in\{0,1\}^M$の入力
S111
リカバリメッセージ$m_{rec}\in\{0,1\}^M$のビット長Mの抽出・格納
S112
任意値$k\in Z_q$生成
S113
$R=g^k\in G$
S114
$\Pi=H_0(R)\in\{0,1\}^{L+M}$
S115
$h=H_1(\alpha)\in\{0,1\}^L$, $\alpha=(\Pi, m_{rec})$
S116
$u=H_2(\beta)\in\{0,1\}^M$, $\beta=(\Pi, h)$
S117
$w=m_{rec}(+)u\in\{0,1\}^M$
S118
$d=h|w\in\{0,1\}^{L+M}$
S119
$r=\Pi(+)d\in\{0,1\}^{L+M}$
S120
$t=H_3(\gamma)\in Z_q$, $\gamma=r$
S121
$s=k-t\cdot x\in Z_q$
S122
署名$\sigma=(r,s)$を送信
S123
署名生成処理終了

図12

[図13]

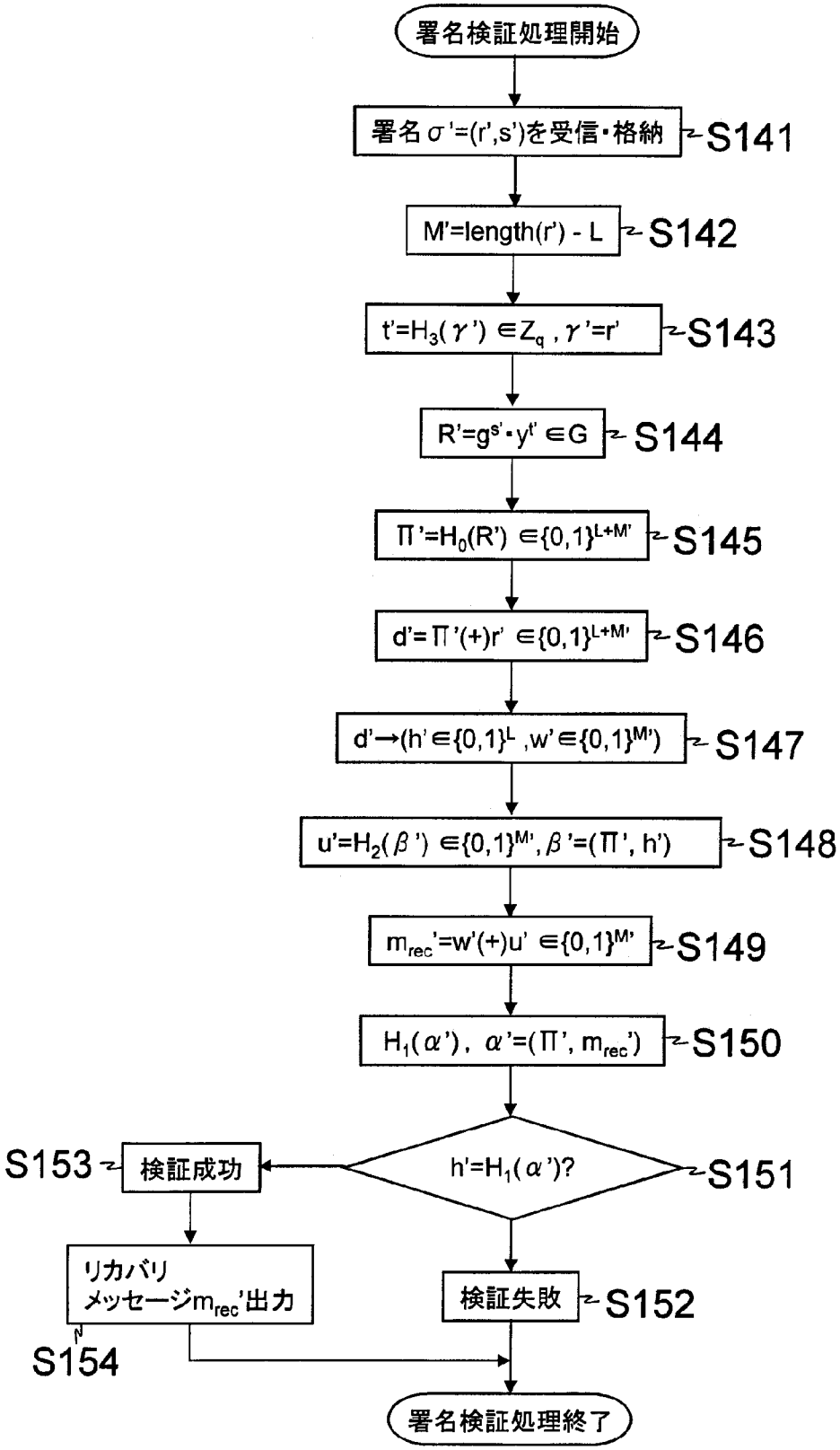

署名検証処理開始
署名σ'=(r',s')を受信・格納
S141
M'=length(r') - L
S142
t'=H3(γ') ∈Zq , γ'=r'
S143
R'=g^s'・y^t' ∈G
S144
Π'=H0(R') ∈{0,1}^L+M'
S145
d'=Π'(+)r' ∈{0,1}^L+M'
S146
d'→(h'∈{0,1}^L ,w'∈{0,1}^M')
S147
u'=H2(β') ∈{0,1}^M', β'=(Π', h')
S148
mrec'=w'(+)u' ∈{0,1}^M'
S149
H1(α'), α'=(Π', mrec')
S150
h'=H1(α')?
S151
S153
検証成功
リカバリ
メッセージmrec'出力
S154
検証失敗
S152
署名検証処理終了

図13

[図14]

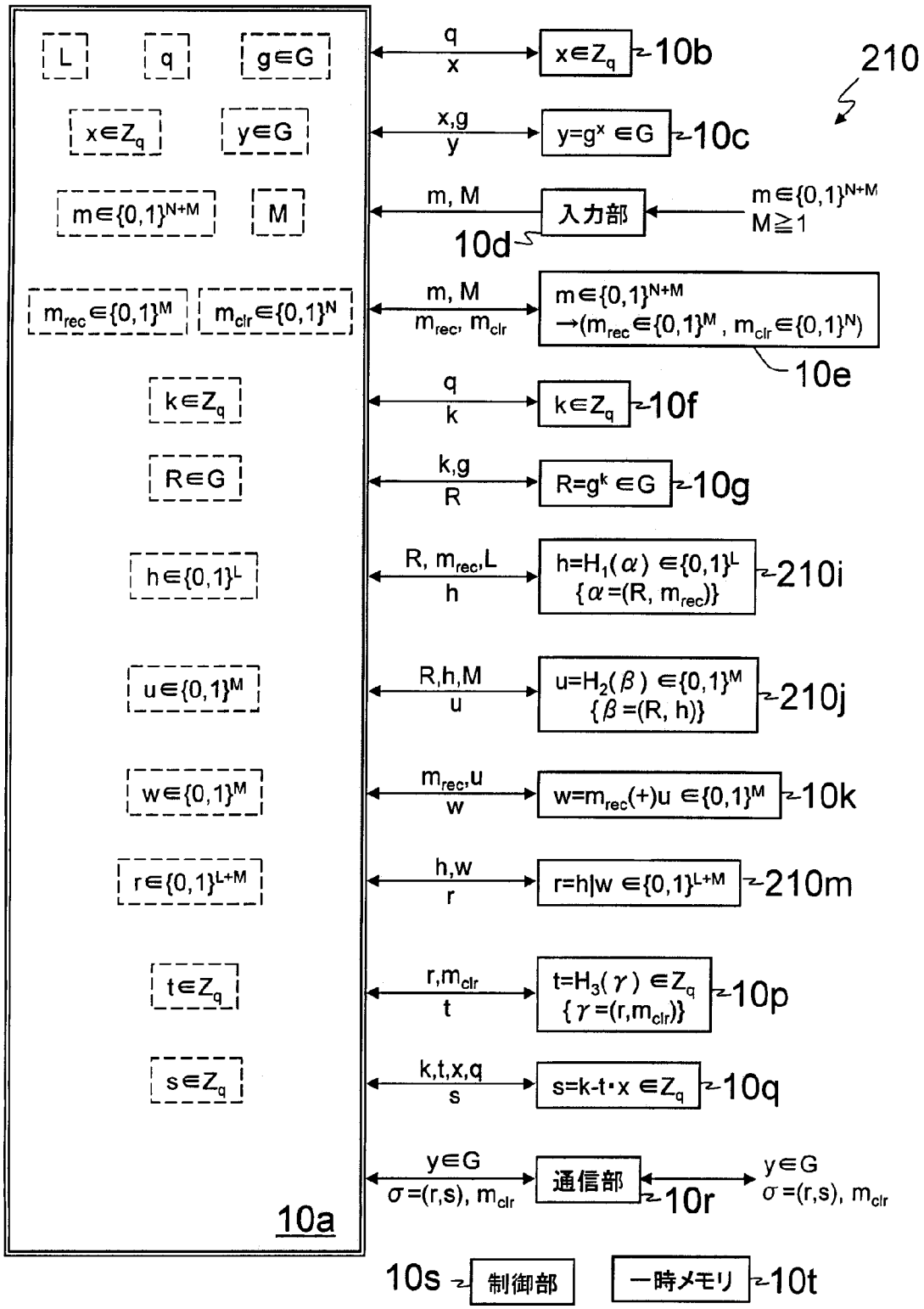

L
q
g∈G
x∈Zq
y∈G
m∈{0,1}N+M
M
mrec∈{0,1}M
mclr∈{0,1}N
k∈Zq
R∈G
h∈{0,1}L
u∈{0,1}M
w∈{0,1}M
r∈{0,1}L+M
t∈Zq
s∈Zq
10a
q
x
x∈Zq
10b
210
x,g
y
y=gx ∈G
10c
m, M
入力部
m∈{0,1}N+M
M≧1
10d
m, M
mrec, mclr
m∈{0,1}N+M
→(mrec∈{0,1}M , mclr∈{0,1}N)
10e
q
k
k∈Zq
10f
k,g
R
R=gk ∈G
10g
R, mrec,L
h
h=H1(α) ∈{0,1}L
{α=(R, mrec)}
210i
R,h,M
u
u=H2(β) ∈{0,1}M
{β=(R, h)}
210j
mrec,u
w
w=mrec(+)u ∈{0,1}M
10k
h,w
r
r=h|w ∈{0,1}L+M
210m
r,mclr
t
t=H3(γ) ∈Zq
{γ=(r,mclr)}
10p
k,t,x,q
s
s=k-t･x ∈Zq
10q
y∈G
σ=(r,s), mclr
通信部
10r
y∈G
σ=(r,s), mclr
10s
制御部
一時メモリ
10t

図14

[図15]

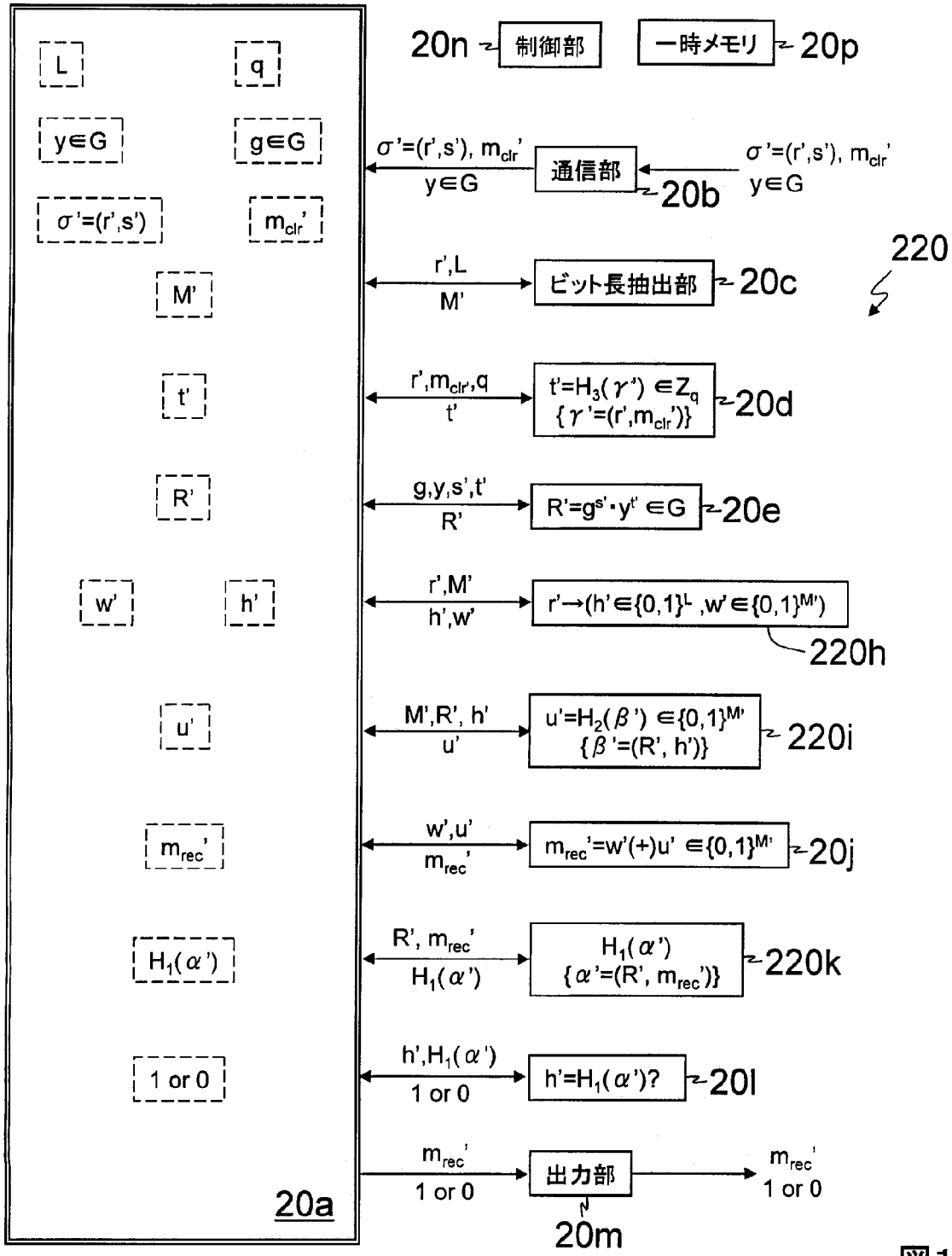

L
q
y∈G
g∈G
σ'=(r',s')
m_clr'
M'
t'
R'
w'
h'
u'
m_rec'
H1(α')
1 or 0
20a
20n
制御部
一時メモリ
20p
σ'=(r',s'), m_clr'
y∈G
通信部
20b
σ'=(r',s'), m_clr'
y∈G
220
r',L
M'
ビット長抽出部
20c
r',m_clr',q
t'
t'=H3(γ') ∈Zq
{γ'=(r',m_clr')}
20d
g,y,s',t'
R'
R'=g^s'・y^t' ∈G
20e
r',M'
h',w'
r'→(h'∈{0,1}^L ,w'∈{0,1}^M')
220h
M',R', h'
u'
u'=H2(β') ∈{0,1}^M'
{β'=(R', h')}
220i
w',u'
m_rec'
m_rec'=w'(+)u' ∈{0,1}^M'
20j
R', m_rec'
H1(α')
H1(α')
{α'=(R', m_rec')}
220k
h',H1(α')
1 or 0
h'=H1(α')?
20l
m_rec'
1 or 0
出力部
20m
m_rec'
1 or 0

図15

[図16]

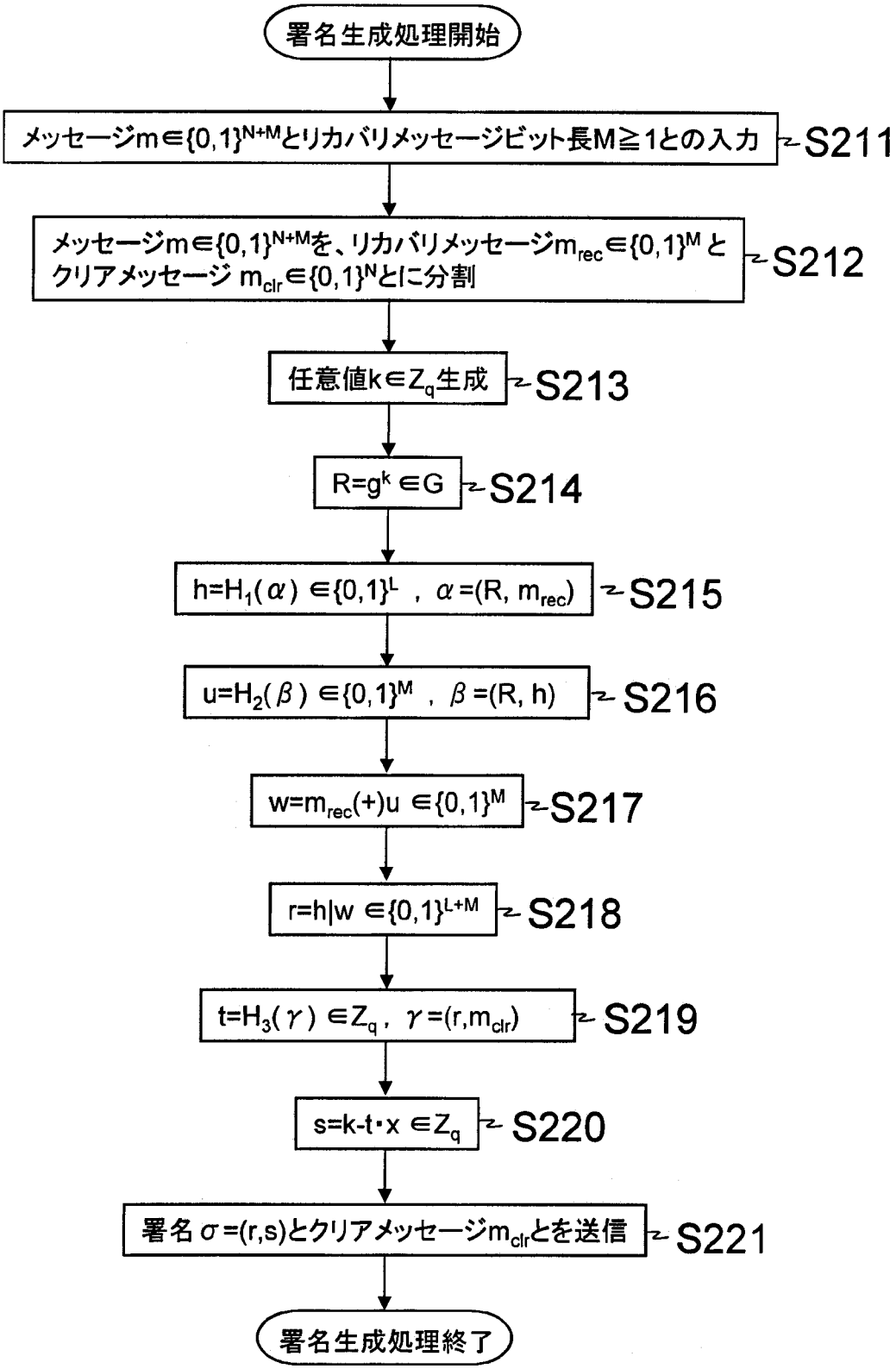

署名生成処理開始
メッセージm∈{0,1}^{N+M}とリカバリメッセージビット長M≧1との入力
S211
メッセージm∈{0,1}^{N+M}を、リカバリメッセージm_rec∈{0,1}^Mとクリアメッセージ m_clr∈{0,1}^Nとに分割
S212
任意値k∈Z_q生成
S213
R=g^k ∈G
S214
h=H_1(α) ∈{0,1}^L , α=(R, m_rec)
S215
u=H_2(β) ∈{0,1}^M , β=(R, h)
S216
w=m_rec(+)u ∈{0,1}^M
S217
r=h|w ∈{0,1}^{L+M}
S218
t=H_3(γ) ∈Z_q , γ=(r,m_clr)
S219
s=k-t･x ∈Z_q
S220
署名 σ=(r,s)とクリアメッセージm_clrとを送信
S221
署名生成処理終了

図16

[図17]

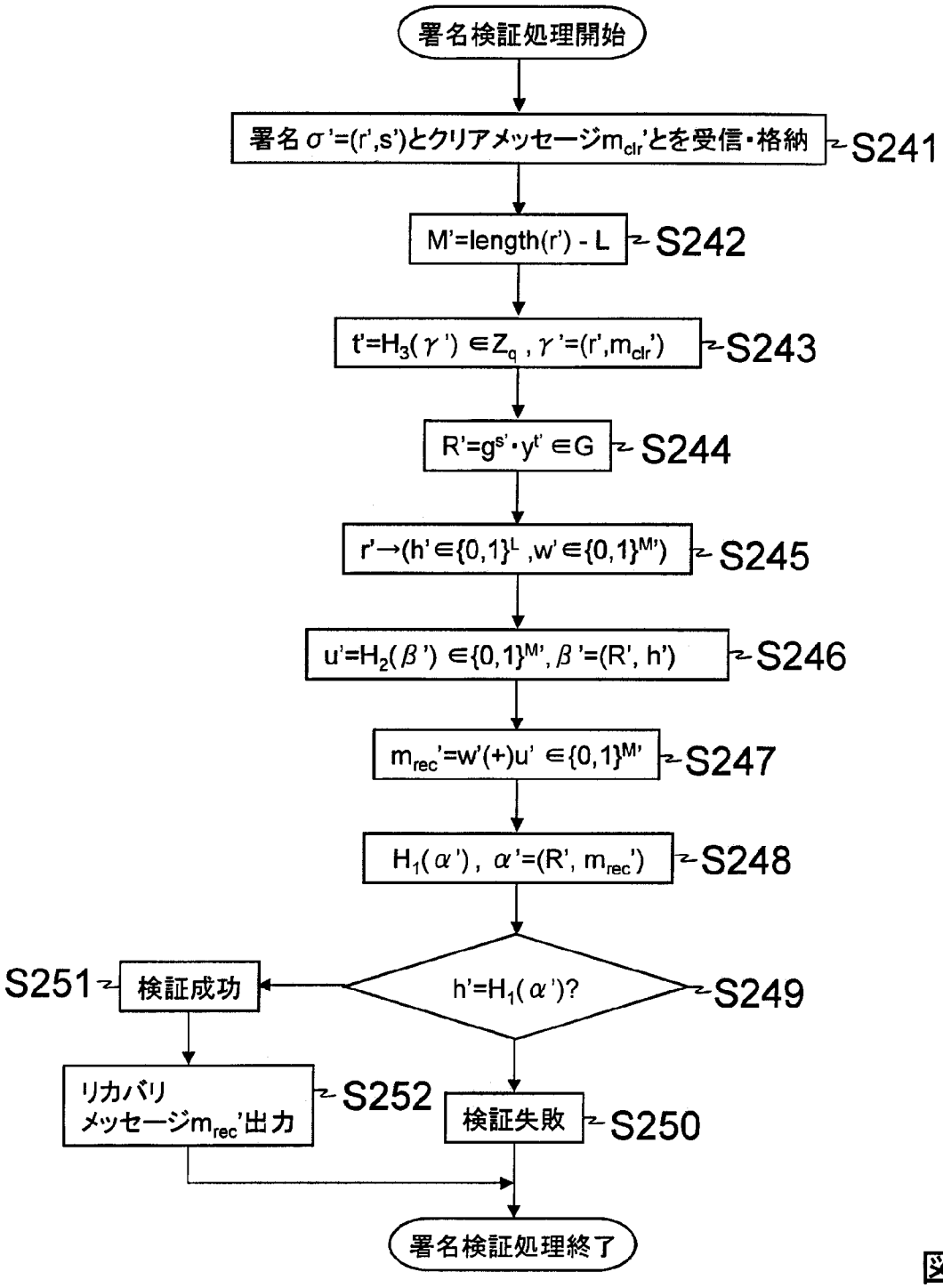

署名検証処理開始
署名σ'=(r',s')とクリアメッセージm_clr'とを受信・格納
S241
M'=length(r') - L
S242
t'=H3(γ') ∈Zq , γ'=(r',m_clr')
S243
R'=g^s'・y^t' ∈G
S244
r'→(h'∈{0,1}^L ,w'∈{0,1}^M')
S245
u'=H2(β') ∈{0,1}^M', β'=(R', h')
S246
m_rec'=w'(+)u' ∈{0,1}^M'
S247
H1(α'), α'=(R', m_rec')
S248
h'=H1(α')?
S249
S251
検証成功
検証失敗
S250
リカバリ
メッセージm_rec'出力
S252
署名検証処理終了

図17

[図18]

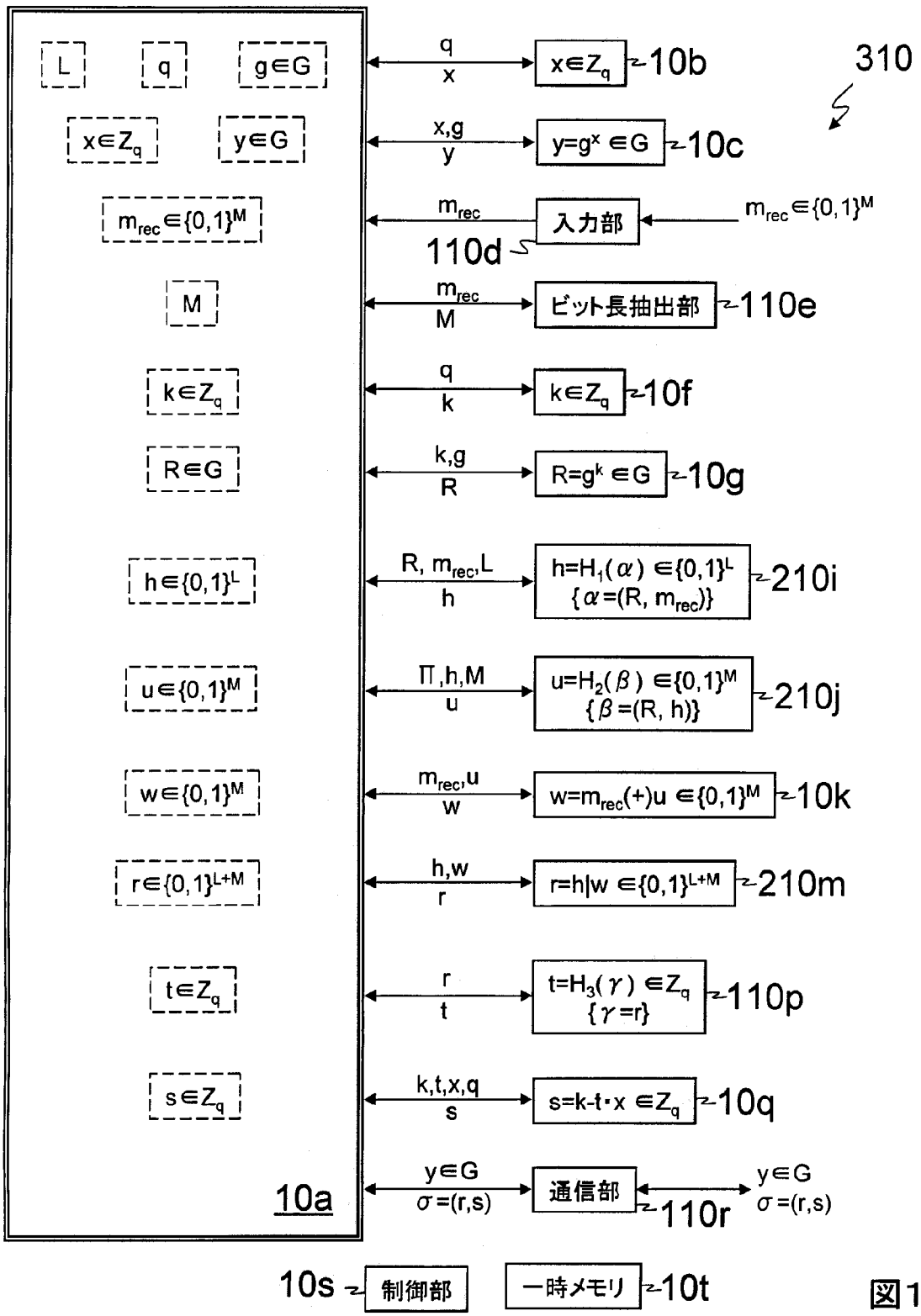

L
q
g∈G
x∈Zq
y∈G
mrec∈{0,1}M
M
k∈Zq
R∈G
h∈{0,1}L
u∈{0,1}M
w∈{0,1}M
r∈{0,1}L+M
t∈Zq
s∈Zq
10a
q
x
x∈Zq
10b
310
x,g
y
y=gx ∈G
10c
mrec
入力部
mrec∈{0,1}M
110d
mrec
M
ビット長抽出部
110e
q
k
k∈Zq
10f
k,g
R
R=gk ∈G
10g
R, mrec,L
h
h=H1(α) ∈{0,1}L
{α=(R, mrec)}
210i
Π,h,M
u
u=H2(β) ∈{0,1}M
{β=(R, h)}
210j
mrec,u
w
w=mrec(+)u ∈{0,1}M
10k
h,w
r
r=h|w ∈{0,1}L+M
210m
r
t
t=H3(γ) ∈Zq
{γ=r}
110p
k,t,x,q
s
s=k-t･x ∈Zq
10q
y∈G
σ=(r,s)
通信部
110r
y∈G
σ=(r,s)
10s
制御部
一時メモリ
10t

図18

[図19]

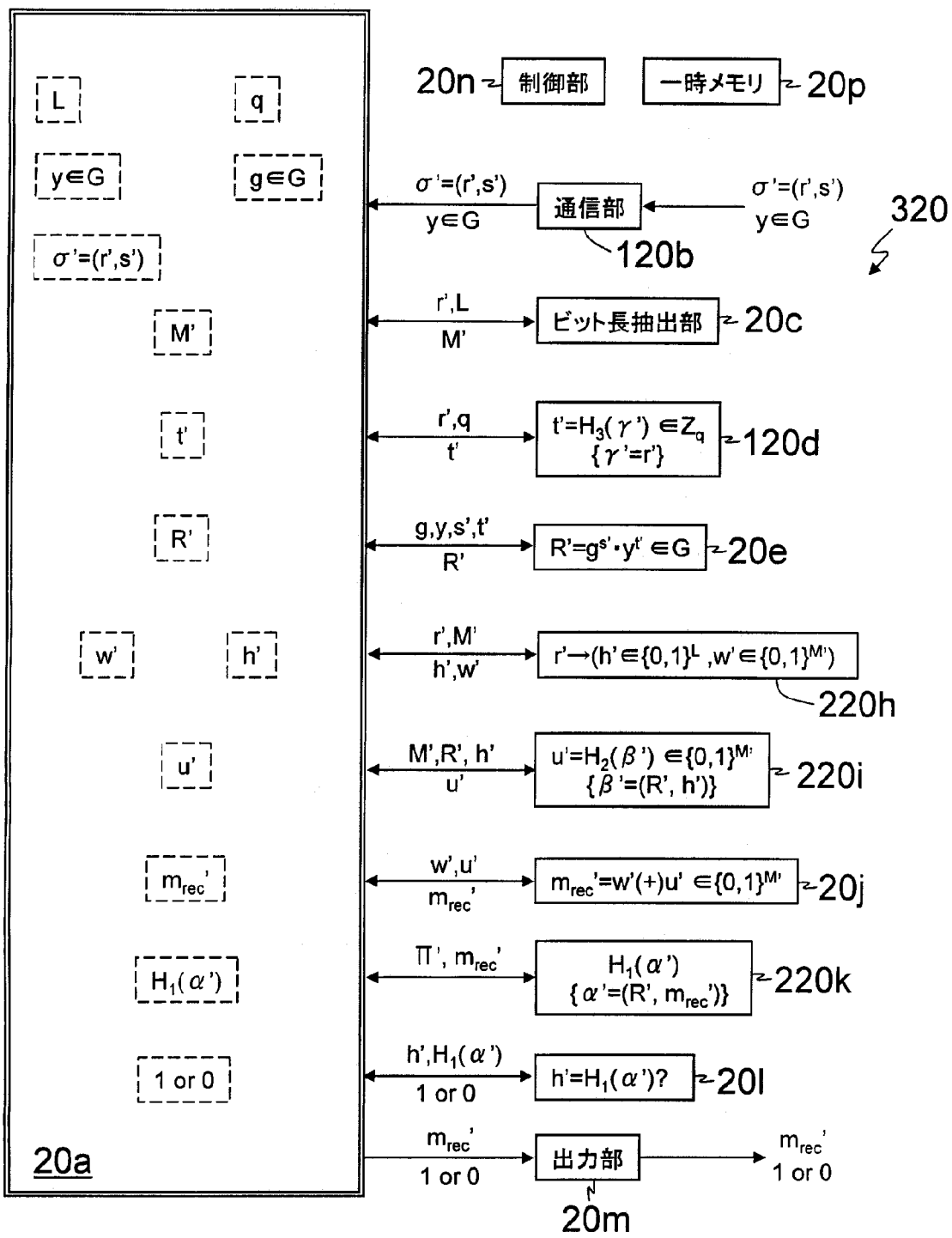

20n
制御部
一時メモリ
20p
L
q
y∈G
g∈G
σ'=(r',s')
M'
t'
R'
w'
h'
u'
m_rec'
H1(α')
1 or 0
20a
σ'=(r',s')
y∈G
通信部
σ'=(r',s')
y∈G
320
120b
r',L
M'
ビット長抽出部
20c
r',q
t'
t'=H3(γ') ∈Zq
{γ'=r'}
120d
g,y,s',t'
R'
R'=gs'・yt' ∈G
20e
r',M'
h',w'
r'→(h'∈{0,1}L ,w'∈{0,1}M')
220h
M',R', h'
u'
u'=H2(β') ∈{0,1}M'
{β'=(R', h')}
220i
w',u'
m_rec'
m_rec'=w'(+)u' ∈{0,1}M'
20j
Π', m_rec'
H1(α')
{α'=(R', m_rec')}
220k
h',H1(α')
1 or 0
h'=H1(α')?
20l
m_rec'
1 or 0
出力部
m_rec'
1 or 0
20m

図19

[図20]

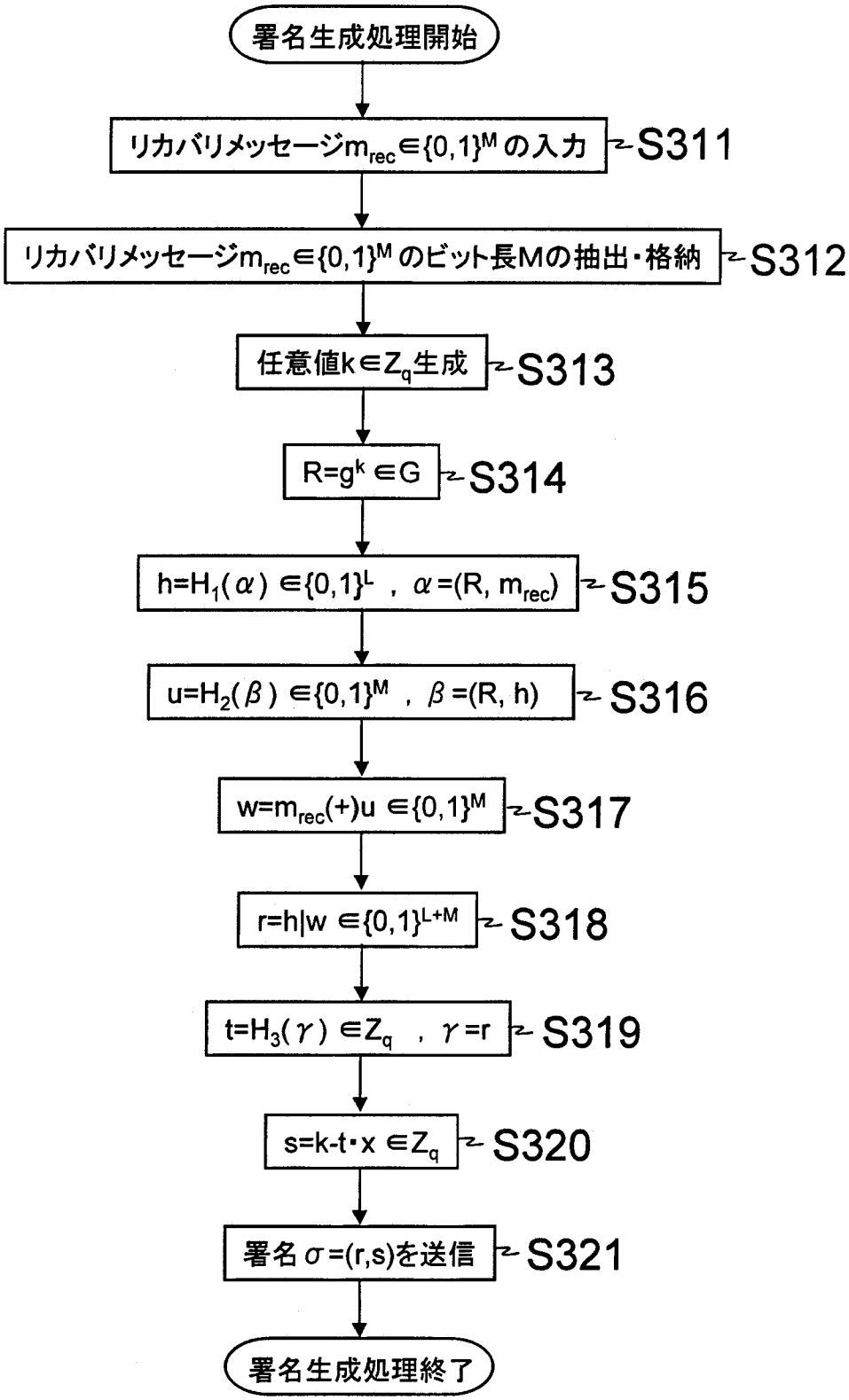

署名生成処理開始
リカバリメッセージ$m_{rec}\in\{0,1\}^M$の入力
S311
リカバリメッセージ$m_{rec}\in\{0,1\}^M$のビット長Mの抽出・格納
S312
任意値$k\in Z_q$生成
S313
$R=g^k\in G$
S314
$h=H_1(\alpha)\in\{0,1\}^L$, $\alpha=(R, m_{rec})$
S315
$u=H_2(\beta)\in\{0,1\}^M$, $\beta=(R, h)$
S316
$w=m_{rec}(+)u\in\{0,1\}^M$
S317
$r=h|w\in\{0,1\}^{L+M}$
S318
$t=H_3(\gamma)\in Z_q$, $\gamma=r$
S319
$s=k-t\cdot x\in Z_q$
S320
署名$\sigma=(r,s)$を送信
S321
署名生成処理終了

図20

[図21]

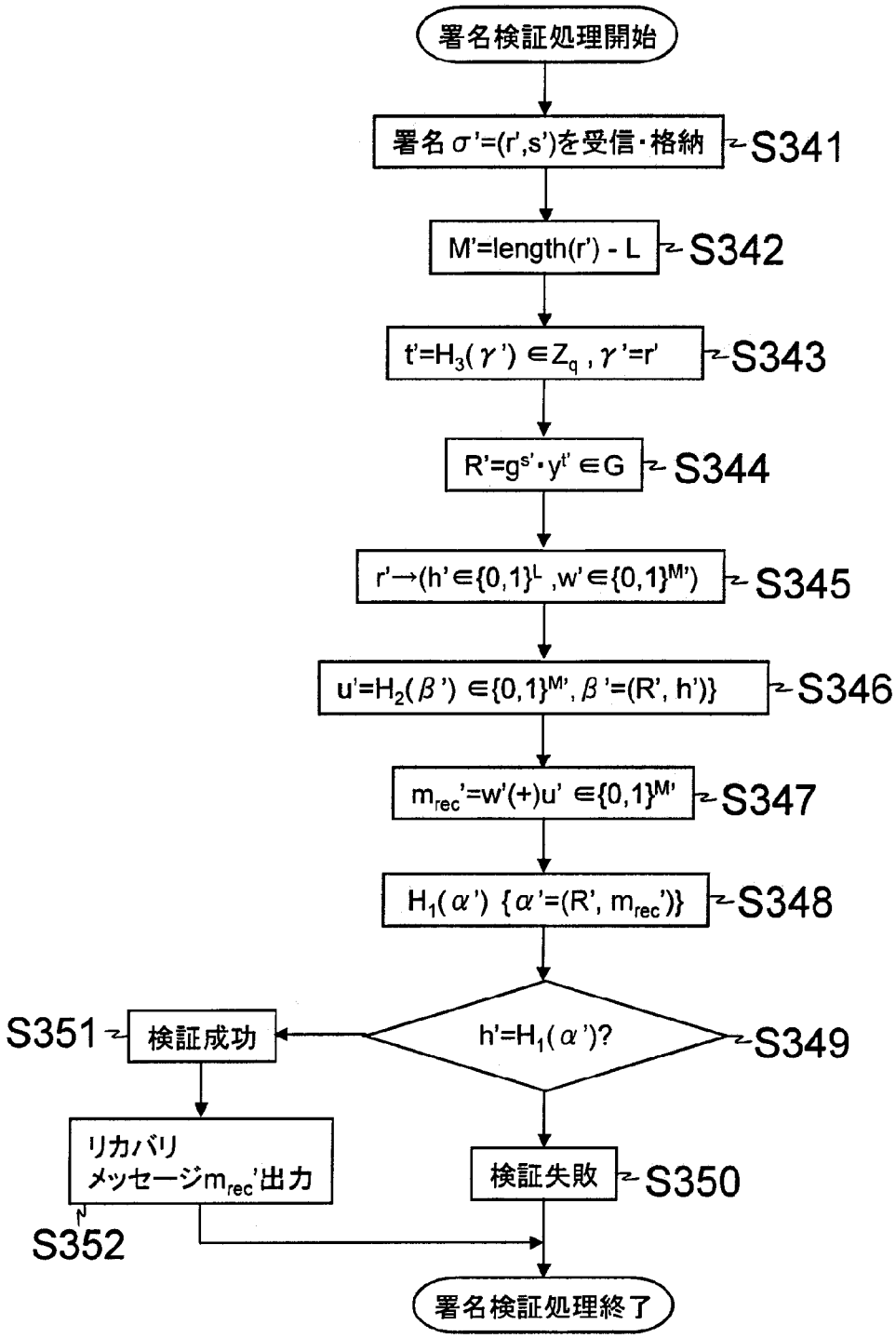

署名検証処理開始
署名σ'=(r',s')を受信・格納
S341
M'=length(r') - L
S342
t'=H3(γ') ∈Zq , γ'=r'
S343
R'=gs'・yt' ∈G
S344
r'→(h'∈{0,1}L ,w'∈{0,1}M')
S345
u'=H2(β') ∈{0,1}M', β'=(R', h')}
S346
mrec'=w'(+)u' ∈{0,1}M'
S347
H1(α') {α'=(R', mrec')}
S348
h'=H1(α')?
S349
S351
検証成功
リカバリ
メッセージmrec'出力
S352
検証失敗
S350
署名検証処理終了

図21

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2008/057962

A. CLASSIFICATION OF SUBJECT MATTER
G09C1/00(2006.01)i, *H04L9/32*(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G09C1/00, H04L9/32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2008
Kokai Jitsuyo Shinan Koho 1971-2008 Toroku Jitsuyo Shinan Koho 1994-2008

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Masayuki ABE and Tatsuaki OKAMOTO, "A Signature Scheme with Message Recovery as Secure as Discrete Logarithm", IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, 2001.01.01, VOL.E84-A,NO.1, p.197-204	1-18,25,26
A	Arsuko Miyaji, "A Message Recovery Signature Scheme Equivalent to DSA over Elliptic Curves", Lecture Notes in Computer Science, 1996.11.06 (received date), Vol.1163, p.1-14	1-18,25,26
A	Lecon A. Pintsov and Scott A. Vanstone, "Postal Revenue Collection in the Digital Age", http://citeseer.ist.psu.edu/339598.html, 2000	1-18,25,26

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 June, 2008 (24.06.08)	01 July, 2008 (01.07.08)
Name and mailing address of the ISA/ Japanese Patent Office	Authorized officer
Facsimile No.	Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2008/057962

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Naisa Nyberg and Rainer A. Rueppel, "Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem", Lecture Notes in Computer Science, 1995.10.16 (received date), Vol.950, p.182-193	1-18,25,26
A	Akihiro Mihara, Keisuke Tanaka, "Short Signatures with Message Recovery in the Random Oracle Model", 2004 Nen Symposium on Cryptography and Information Security (SCIS2004) Yokoshu CD-ROM, 2004.01.27, 2D4 Digital Shomei II, 2D4-4	1-18,25,26
A	Kefei Chen, "Signature with message recovery", Electronics Letters, 1998.10.01, Vol.34, No.20, p.1934	1-18,25,26
T,A	Koutarou Suzuki, Eiichiro Fujisaki, "A CDH-based Message Recovery Signature with Tight Security Reduction", 2008 Nen Symposium on Cryptography and Information Security Yokoshu, 22 January, 2008 (22.01.08), 3F2 Shomei (1), 3F2-5, pages 1 to 53F	1-18,25,26

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2008/057962

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. [X] Claims Nos.: 19-24
because they relate to subject matter not required to be searched by this Authority, namely:
See the "(extra sheet)".

2. [ ] Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. [ ] Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. [ ] As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. [ ] As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. [ ] As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. [ ] No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

the

[ ] The additional search fees were accompanied by the applicant's protest and, where applicable, payment of a protest fee.

[ ] The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

[ ] No protest accompanied the payment of additional search fees.

Continuation of Box No.II-1 of continuation of first sheet(2)

The inventions of claims 19-24 relate to an algorithm for expressing an artificial rule with scientific and mathematic theories and are scientific and mathematic theories. Therefore, the subject matter is not required to be searched by this International Searching Authority under PCT Article 17(2)(a)(i) and PCT Rule 39.1(v).

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００８／０５７９６２

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00(2006.01)i, H04L9/32(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00, H04L9/32

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００８年
日本国実用新案登録公報	１９９６－２００８年
日本国登録実用新案公報	１９９４－２００８年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	Masayuki ABE and Tatsuaki OKAMOTO, “A Signature Scheme with Message Recovery as Secure as Discrete Logarithm”, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, 2001.01.01, VOL.E84-A,NO.1, p.197-204	1-18,25,26
A	Arsuko Miyaji, “A Message Recovery Signature Scheme Equivalent to DSA over Elliptic Curves”, Lecture Notes in Computer Science, 1996.11.06（受入日）, Vol.1163, p.1-14	1-18,25,26

☑　Ｃ欄の続きにも文献が列挙されている。　　☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの
「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「Ｏ」口頭による開示、使用、展示等に言及する文献
「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
２４．０６．２００８	０１．０７．２００８

国際調査機関の名称及びあて先	特許庁審査官（権限のある職員）	５Ｓ	４２２９
日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	青木　重徳 電話番号　０３－３５８１－１１０１　内線　３５４６		

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
A	Lecon A. Pintsov and Scott A. Vanstone, “Postal Revenue Collection in the Digital Age”, http://citeseer.ist.psu.edu/339598.html, 2000	1-18,25,26
A	Naisa Nyberg and Rainer A. Rueppel, “Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem”, Lecture Notes in Computer Science, 1995.10.16（受入日）, Vol.950, p.182-193	1-18,25,26
A	Akihiro Mihara, Keisuke Tanaka, “Short Signatures with Message Recovery in the Random Oracle Model”, ２００４年暗号と情報セキュリティシンポジウム(SCIS2004)予稿集 CD-ROM, 2004.01.27, ２Ｄ４　ディジタル署名 II, ２Ｄ４－４	1-18,25,26
A	Kefei Chen, “Signature with message recovery”, Electronics Letters, 1998.10.01, Vol.34, No.20, p.1934	1-18,25,26
T,A	Koutarou Suzuki, Eiichiro Fujisaki, “A CDH-based Message Recovery Signature with Tight Security Reduction”, ２００８年暗号と情報セキュリティシンポジウム予稿集, 2008.01.22, ３Ｆ２　署名（１）, ３Ｆ２－５, p.1-5	1-18,25,26

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２００８／０５７９６２

第Ⅱ欄　請求の範囲の一部の調査ができないときの意見（第１ページの２の続き）

法第8条第3項（ＰＣＴ17条(2)(a)）の規定により、この国際調査報告は次の理由により請求の範囲の一部について作成しなかった。

１．☑ 請求の範囲 １９－２４ は、この国際調査機関が調査をすることを要しない対象に係るものである。
つまり、
「(特別ページ)」を参照。

２．☐ 請求の範囲＿＿＿＿＿＿＿＿ は、有意義な国際調査をすることができる程度まで所定の要件を満たしていない国際出願の部分に係るものである。つまり、

３．☐ 請求の範囲＿＿＿＿＿＿＿＿ は、従属請求の範囲であってＰＣＴ規則6.4(a)の第２文及び第3文の規定に従って記載されていない。

第Ⅲ欄　発明の単一性が欠如しているときの意見（第１ページの３の続き）

次に述べるようにこの国際出願に二以上の発明があるとこの国際調査機関は認めた。

１．☐ 出願人が必要な追加調査手数料をすべて期間内に納付したので、この国際調査報告は、すべての調査可能な請求の範囲について作成した。

２．☐ 追加調査手数料を要求するまでもなく、すべての調査可能な請求の範囲について調査することができたので、追加調査手数料の納付を求めなかった。

３．☐ 出願人が必要な追加調査手数料を一部のみしか期間内に納付しなかったので、この国際調査報告は、手数料の納付のあった次の請求の範囲のみについて作成した。

４．☐ 出願人が必要な追加調査手数料を期間内に納付しなかったので、この国際調査報告は、請求の範囲の最初に記載されている発明に係る次の請求の範囲について作成した。

追加調査手数料の異議の申立てに関する注意

☐ 追加調査手数料及び、該当する場合には、異議申立手数料の納付と共に、出願人から異議申立てがあった。

☐ 追加調査手数料の納付と共に出願人から異議申立てがあったが、異議申立手数料が納付命令書に示した期間内に支払われなかった。

☐ 追加調査手数料の納付はあったが、異議申立てはなかった。

請求の範囲１９－２４は、人為的な取り決めを科学及び数学の理論で表現したアルゴリズムであり、科学及び数学の理論に該当し、ＰＣＴ第１７条（２）（ａ）（ｉ）及びＰＣＴ規則３９．１（ｖ）の規定により、この国際調査機関が調査することを要しない対象に係るものである。