(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
2007 年8 月16 日 (16.08.2007)

PCT

(10) 国際公開番号
WO 2007/091531 A2

(51) 国際特許分類:
G06F 17/10 (2006.01)

(21) 国際出願番号: PCT/JP2007/051959

(22) 国際出願日: 2007 年2 月6 日 (06.02.2007)

(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2006-032650 2006 年2 月9 日 (09.02.2006) JP

(71) 出願人 (米国を除く全ての指定国について): 日本電気株式会社 (**NEC CORPORATION**) [JP/JP]; 〒1088001 東京都港区芝五丁目７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): 寺西 勇 (**TERANISHI, Isamu**) [JP/JP]; 〒1088001 東京都港区芝五丁目７番１号 日本電気株式会社内 Tokyo (JP).

(74) 代理人: 宮崎 昭夫, 外(**MIYAZAKI, Teruo et al.**); 〒1070052 東京都港区赤坂１丁目９番２０号 第１６興和ビル８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, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, 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, HU, IE, IS, IT, LT, LU, LV, MC, NL, 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:
— 第１７条(2)(a)に基づく宣言；要約なし；国際調査機関により点検されていない発明の名称.

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

WO 2007/091531 A2

(54) **Title:** VERIFICATION CHECK SYSTEM, VERIFYING DEVICE, CHECK DEVICE, VERIFICATION CHECK METHOD, AND PROGRAM

(54) 発明の名称: 証明検証システム、証明装置、検証装置、証明検証方法、およびプログラム

(57) **Abstract:**

(57) 要約:

明 細 書

証明検証システム、証明装置、検証装置、証明検証方法、およびプログラム

技術分野

[0001] 本発明は、複数の情報のうちいくつかを知っていることの証明を可能にし、また、その証明の正当性の検証を可能にするための証明検証システム、証明装置および検証装置と、証明検証方法と、その方法をコンピュータに実行させるためのプログラムとに関する。

背景技術

[0002] 証明装置と呼ばれる情報処理装置が検証装置と呼ばれる他の情報処理装置に何らかの性質を満たす秘密情報を知っていることを証明するプロトコルを「知識の証明」という。検証装置は証明装置の行った証明の正当性を検証し、正当であるかそうでないかを判断する。

[0003] 知識の証明方式は様々な暗号プロトコルを作る際の部品として用いられている。直接的な応用例として認証方式や署名方式がある。認証方式も署名方式も、インターネット上で他の人に当人性を証明するために用いられる技術である。ある種の認証方式、署名方式はIPSec(Security Architecture for Internet Protocol)として実用化されており、産業上の利用価値が高い。

[0004] 文献(R.Cramer, I.Damgaard, and B.Schoenmakers., "Proofs of partial knowledge and simplified design of witness hiding protocols", In CRYPTO'94, LNCS 2139, Springer-Verlag, 1994, pp. 174-187:以下では、この文献を文献1と称する)において、Cramer等は、俗に「Orの証明」と呼ばれる知識の証明方式を提案した。nを自然数とし、kをn以下の自然数とすると、この方式では、証明装置がn個の秘密のうち少なくともk個を知っていることを検証者に証明できる。

[0005] 「Orの証明」はある種の匿名性を持つ認証方式を実現するのに必要で、これも様々な暗号プロトコルに応用されている。「Orの証明」は、例えば、文献(Isamu Teranishi, Jun Furukawa, and Kazue Sako., "k-Times Anonymous Authentication (Ext

ended Abstract)", In ASIACRYPT 2004, LNCS 3329, Springer-Verlag, 2004, pp. 308-322:以下では、この文献を文献2と称する)や文献(寺西勇,「離散対数問題にtight安全でかつ計算量が少ない署名方式」, In Proceedings of the 2005 Symposium on Cryptography and Information Security, 3E3-4, January 2005:以下では、この文献を文献3と称する)で利用されている。文献3は電子投票、電子現金、匿名視聴といった応用を持つ方式であり、これも産業上の利用価値が高い。

[0006] また、Cramer等は文献1で、次のような一般的な知識の証明方式も提案している。あらかじめ「アクセス構造」(Douglas R. Stinson,「暗号理論の基礎」, 共立出版株式会社, P355-356を参照)と呼ばれるものを決めておく。この方式では、証明装置はn個の秘密のうち複数個を知っていて、しかもその複数個は前述のアクセス構造を満たすものであることを証明できる。

n個の秘密をx__1, … ,x__nとする。

Cramer等は、x__1, … ,x__nが満たす性質が同じである場合に対して知識の証明方式を提案している。

Cramer等の知識の証明方式は、x__1の知識の証明方式、x__2の知識の証明方式、…、x__nの知識の証明方式を組み合わせることで、より一般の場合にも用いることができる。しかし、Cramer等の知識の証明方式を使うことができるのは、x__1の知識の証明方式、x__2の知識の証明方式、…、x__nの知識の証明方式が以下の条件1,2を満たす場合のみである。

条件1. x__1の知識の証明方式、x__2の知識の証明方式、…、x__nの知識の証明方式は3-ムーブのパブリックコイン証明方式である。

条件2. x__iの知識の証明方式でChallengeを選ぶ空間をChallengeSpace__iとするとき、ChallengeSpace__1= …= ChallengeSpace__nでなければならない。

なお、「3-ムーブのパブリックコイン証明方式」および「Challenge」等の言葉の説明は、後述の(定義)の章で行う。

発明の開示

[0007] Cramer等の知識の証明方式は、上述した2つの条件を満たす場合にしか使うことができない。また、Cramer等の知識の証明方式と同様な効果が得られ、かつ、必要と

される条件がより少ない方式を作る必要がある。

[0008] 本発明の目的は、上記問題点に鑑みてなされたものであり、秘密データ保持の証明をより速く検証可能にした証明検証システム、証明装置および検証装置と、証明検証方法と、その方法をコンピュータに実行させるためのプログラムとを提供することにある。

[0009] 上記目的を達成するための本発明の証明検証システムは、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータが格納された証明装置記憶部、ならびに、n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n−m)とし、証明装置記憶部から巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対してy__{j__v}とChallenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、Commit__{j__1}，…，Commit__{j__p}およびCommit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを外部に送信し、外部からチャレンジ値cを受信すると、チャレンジ値cおよびs__{j__1}，…，s__{j__p}から残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、y__i、Commit__iおよびChallenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを外部に送信する証明装置制御部を含む証明装置と、

証明装置と通信可能に接続され、複数の乱数および集合の元y__1，…，y__nが格納された検証装置記憶部、ならびに、証明装置からコミット値Commit__1,…,Commit__nを受信すると、複数の乱数から選択したcをチャレンジ値として証明装置に送信し、巡回群の元s__1，…，s__nおよびレスポンス値Response__1，…，Response__nを証明装置から受信すると、s__1，…，s__nがチャレンジ値cから正当な方法で生成

された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対してs__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文がy__iの正当な証明文であるか否かを検証し、正当である場合、証明装置による証明を受理し、正当でない場合、証明装置による証明を受理しない検証装置制御部を含む検証装置と、

を有する構成である。

[0010] また、本発明の証明検証システムは、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、n個の元を含む巡回群のデータおよび複数の元を含む群のデータが格納された証明装置記憶部、ならびに、n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n-m)とし、外部からコミット値Comを受信すると、コミット値Comを証明装置記憶部に格納し、証明装置記憶部から巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して元y__{j__v}とChallenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、Commit__{j__1}，…，Commit__{j__p}およびCommit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、複数の元を含む群からランダムに元c__2を選び、元c__2およびCommit__1，…，Commit__nを外部に送信し、コミット値Comを算出するための、複数の元を含む群の元c__1を外部から受信すると、コミット値Comが元c__1の正当なコミットメントであるか否かを検証し、正当でない場合、証明の続行を拒否し、正当である場合、c__1とc__2を乗算した値cを求め、s__{j__1}，…，s__{j__p}と値cから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、y__i、Commit__i、およびChallenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを検証装置に送信する証明装置制御部

を含む証明装置と、

証明装置と通信可能に接続され、複数の元を含む群のデータが格納された検証装置記憶部、ならびに、複数の元を含む群からランダムに元c__1を選び、元c__1のコミット値Comを算出して証明装置に送信し、複数の元を含む群の元c__2、およびコミット値Commit__1，…，Commit__nを証明装置から受信すると、元c__1を証明装置に送信し、巡回群の元s__1，…，s__n、およびレスポンス値Response__1，…，Response__nを証明装置から受信すると、c__1とc__2を乗算した値cを求め、s__1，…，s__nが値cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文がy__iの正当な証明文であるか否かを検証し、正当である場合、証明装置による証明を受理し、正当でない場合、証明装置による証明を受理しない検証装置制御部を含む検証装置と、

を有する構成である。

[0011] また、本発明の証明検証システムは、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータが格納された証明装置記憶部、ならびに、n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n-m)とし、証明装置記憶部から巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して元y__{j__v}とChallenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、Commit__{j__1}，…，Commit__{j__p}およびCommit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、ハッシュ値cとs__{j__1}，…，s__{j__p}とから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1，…，nに対して各s__iにそのハッシュ値をチャレンジ値Challenge

__iとし、y__i、Commit__iおよびChallenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを外部に送信する証明装置制御部を含む証明装置と、

証明装置と通信可能に接続され、集合の元y__1，…，y__nが格納された検証装置記憶部、ならびに、巡回群の元s__1，…，s__n、コミット値Commit__1，…，Commit__n、およびレスポンス値Response__1，…，Response__nを証明装置から受信すると、Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、s__1，…，s__nがハッシュ値cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文がy__iの正当な証明文であるか否かを検証し、正当である場合、証明装置による証明を受理し、正当でない場合、証明装置による証明を受理しない検証装置制御部を含む検証装置と、

を有する構成である。

[0012] 本発明では、証明装置からコミット値が検証装置に送信され、検証装置から証明装置に送信されたチャレンジ値に対して証明装置からレスポンス値が送信され、検証装置で証明装置の証明文が検証される。このようにして、証明装置が秘密データを正規に保持しているか否かが検証可能となる。よって、証明装置がn個の秘密データのうちm個を保持していることを検証装置に証明可能となる。2つの条件が必要な従来の知識の証明方式に対し、1つの条件を満たすことで検証できる。

図面の簡単な説明

[0013] [図1]図1は実施形態1における証明検証システムの一構成例を示すブロック図である。

[図2]図2は実施形態1における証明検証システムのデータ送受信の動作手順を示すフローチャートである。

[図3]図3は実施形態1における証明装置のコミット計算手段の動作手順を示すフローチャートである。

[図4]図4は実施形態1における検証装置のチャレンジ計算手段の動作手順を示すフローチャートである。

[図5]図5は実施形態1における証明装置のレスポンス計算手段の動作手順を示すフローチャートである。

[図6]図6は実施形態1における検証装置の検証手段の動作手順を示すフローチャートである。

[図7]図7は実施形態2における証明検証システムの一構成例を示すブロック図である。

[図8]図8は実施形態2における証明検証システムのデータ送受信の動作手順を示すフローチャートである。

[図9]図9は実施形態2における証明検証システムのデータ送受信の動作手順を示すフローチャートである。

[図10]図10は実施形態2における検証装置のチャレンジコミット計算手段の動作手順を示すフローチャートである。

[図11]図11は実施形態2における証明装置のコミット計算手段の動作手順を示すフローチャートである。

[図12]図12は実施形態2における検証装置のチャレンジ計算手段の動作手順を示すフローチャートである。

[図13]図13は実施形態2における証明装置のレスポンス計算手段の動作手順を示すフローチャートである。

[図14]図14は実施形態2における検証装置の検証手段の動作手順を示すフローチャートである。

[図15]図15は実施形態3における証明検証システムの一構成例を示すブロック図である。

[図16]図16は実施形態3における証明検証システムのデータ送受信の動作手順を示すフローチャートである。

[図17]図17は実施形態3における証明装置の証明文作成手段の動作手順を示すフローチャートである。

[図18]図18は実施形態3における証明装置のチャレンジ計算手段の動作手順を示すフローチャートである。

[図19]図19は実施形態3における検証装置の検証手段の動作手順を示すフローチャートである。

符号の説明

[0014] 100 証明装置

101 証明装置記憶部

102 証明装置制御部

200 検証装置

201 検証装置記憶部

202 検証装置制御部

発明を実施するための最良の形態

[0015] (定義)

[3-ムーブのパブリックコイン証明方式]

X,Yを集合とする。(Y, X, FuncCommit, Hash, FuncResponse, ChallengeSpace, Verify, Simulator)の証明文が次の性質1,…,7を満たすとき、(Y, X, FuncCommit, Hash, FuncResponse, Verify, Simulator)を3-ムーブのパブリックコイン証明方式という。

1, 組FuncCommit, Hash, FuncResponse, Simulator, Verifyは関数である。

2, ChallengeSpaceは集合である。

3, FuncCommitはY×Xの元を入力とし、データCommitとStateを出力する。

4, HashはChallengeSpaceに値を取るハッシュ関数である。

5, (y,x)をY×Xの元とし、ChallengeをChallengeSpaceの元とする。FuncResponseは、y, x, Commit, Challenge, Stateが入力されると、Responseを出力する。

6, Verifyは、y, Commit, Challenge, Responseが入力されると、AcceptもしくはRejectを出力する。

7, Simulatorは、Yの元が入力されると、組(Commit, Challenge, Response)を出力する。

[0016] なお、チャレンジ値Challengeは、通信相手の装置に対して、どのような回答をするかを試すための問いかけの情報であり、一般的には乱数が知られている。コミット値C

ommitは、自装置が保持する秘密データに関連しているが、その秘密データが直接わからないようにして、通信相手の装置に送る情報である。レスポンス値Responseは、チャレンジ値Challengeの送信元の装置に対して返信する、チャレンジ値の問いかけに対する回答の情報である。

[0017] [3-ムーブのパブリックコイン証明方式の例]

qを素数とし、Gを位数qの有限群とし、mを自然数とし、H= G^mとする。

(Y, X, FuncCommit, Hash, FuncResponse, ChallengeSpace, Verify, Simulator)を以下の1,…,8のように決めると、(Y, X, FuncCommit, Hash, FuncResponse, ChallengeSpace, Verify, Simulator)は3-ムーブのパブリックコイン証明方式である。

1. Y=H×Hとする。
2. Xを位数qの巡回群(Z/qZ)とする。
3. ChallengeSpace=(Z/qZ)とする。
4. y=((g__1, … ,g__m), (h__1, … ,h__m))をYの元とし、xをXの元とする。Y×Xの元(y, x)が入力されると、FuncCommitは、(Z/qZ)の元rをランダムに選んで、(C__1, … ,C__n)= (g__1^{r}, … ,g__m^{r})を計算し、Commit= (C__1, … ,C__n)、State= rを出力する。
5. HashはChallengeSpace=(Z/qZ)に値を取るハッシュ関数である。
6. cを(Z/qZ)の元とし、Challenge=cとする。Yの元y= ((g__1, … ,g__m), (h__1, … ,h__m))、Xの元x、Commit= (C__1, … ,C__n)、Challenge= cおよびState= rが入力されると、FuncResponseは、ρ= cx+ r mod qを計算し、Response= ρを出力する。
7. y= ((g__1, … ,g__m),(h__1, … ,h__m))、Commit= (C__1, … ,C__n)、Challenge= c、Response= ρが入力されると、Verifyは、(g__1^{ρ}, … ,g__m^{ρ})と(h__1^{c} C__1, … ,h__m^{c} C__m)を計算する。そして、(g__1^{ρ}, … ,g__m^{ρ})= (h__1^{c} C__1, … ,h__n^{c} C__n)が成立すれば「accept」を出力し、そうでなければ「reject」を出力する。
8. y=((g__1,…,g__m),(h__1,…,h__m))が入力されると、Simulatorは、ランダムにcとρを選び、(C__1, … ,C__m)= (g__1^{ρ} h__1^{-c}, … ,g__m^{ρ} h__m

^{-c})とし、Commit= (C__1, … ,C__m)、Challenge= c、Response= ρを出力する。

[0018] [Rに関する3-ムーブのパブリックコイン証明方式]

RをY×X上の関係式とする。3-ムーブのパブリックコイン証明方式(Y, X, FuncCommit, Hash, FuncResponse, Verify,Simulator)が次の性質1、2を満たすとき、(Y, X, FuncCommit, Hash, FuncResponse, Verify,Simulator)はRに関する3-ムーブのパブリックコイン証明方式であるという。

1. R(y,x)を満たす任意の(y,x)に対し、(Commit, State)= FuncCommit(y,x)、Challenge= Hash(y,Commit)、Response= FuncResponse(y, Commit, Challenge, State)とすると、Verify(y, Commit, Challenge, Response)= acceptとなる。

2. Yの任意の元yに対し、(Commit,Challenge,Response)=Simulator(y)とすると、Verify(y, Commit, Challenge, Response)= acceptとなる。

本発明は任意の3-ムーブのパブリックコイン証明方式に対して用いることができる。しかし、実用的には何らかの関係式Rに対する3-ムーブのパブリックコイン証明方式であることが望ましい。

[0019] [Rに関する3-ムーブのパブリックコイン証明方式の例]

(Y, X, FuncCommit, Hash, FuncResponse, ChallengeSpace, Verify, Simulator)を[3-ムーブのパブリックコイン証明方式の例]と同様に決める。Xの元xとYの元y= ((g__1, … ,g__m), (h__1, … ,h__m))に対し、R(y, x)を関係式「(h__1, … ,h__m)= (g__1^{x}, … ,g__m^{x})」とすると、(Y, X, FuncCommit, Hash, FuncResponse, ChallengeSpace, Verify, Simulator)はRに関する3-ムーブのパブリックコイン証明方式である。

[0020] [アクセス構造]

Sを整数の有限集合とする。Sの部分集合の族Γが次の性質1を満たすとき、ΓをS上のアクセス構造であるという。

1. Sの任意の部分集合A、Bに対しAがΓの元でかつAがBの部分集合ならBはΓの元である。

[アクセス構造の例1]

nを自然数とする。S＝{1,…,n}とし、Γ={S}の集合とする。このときΓはS上のアクセス構造である。

[アクセス構造の例2]

nを自然数とし、kをn以下の自然数とする。S＝{1,…,n}とし、ΓをSの部分集合でk個以上元を持つものの集合とする。このときΓはS上のアクセス構造である。

[0021] [秘密分散方式]

Sを有限集合とし、ΓをS上のアクセス構造とし、nをSの元の数とする。組(FuncShar, FuncReconstruct, SharSp, KeySp)が次の性質1,…,5を満たすとき、(FuncShar, FuncReconstruct, SharSp, KeySp)はアクセス構造Γを持つ秘密分散方式であるという。

1. FuncSharとFuncReconstructは関数である。
2. SharSpとKeySpは集合である。
3. KeySpの元cが入力されると、FuncSharは、SharSpの元s__1, … ,s__nを出力する。
4. 整数とSharSpの元との組(i__1,u__1),…,(i__m,u__m)が入力されると、FuncSharはKeySpの元を出力するか、または文字列「reject」を出力する。
5. cをKeySpの元とする。FuncSharにcを入力したときの出力をs__1,…,s__nとする。A = {i__1, … ,i__m}をΓの元とする。このとき(i__1, s__{i__1}), … ,(i__m, s__{i__m})をFuncSharに入力すると、FuncSharはcを出力する。

[0022] [秘密分散方式の例1]

n、S、Γを[アクセス構造の例1]で説明したように決める。さらに、qを自然数とする。(FuncShar, FuncReconstruct, SharSp, KeySp)を次の1,…,4のように決めると、(FuncShar, FuncReconstruct, SharSp, KeySp)は秘密分散方式である。

1. SharSpを巡回群(Z/qZ)とする。
2. KeySpを巡回群(Z/qZ)とする。
3. KeySpの元cが入力されると、FuncSharは、SharSpの元s__1,…,s__{n-1}をランダムに選び、s__n= c- s__1,…,s__{n-1} mod qを計算し、(s__1, … ,s__n)を出力する。

4. ((1,s__1),…,(n,s__n))が入力されると、FuncReconstructはs__1+ … +s__nを出力する。

[0023] [秘密分散方式の例2]

n、k、S、Γを[アクセス構造の例2]で説明したように決める。さらに、qを自然数とする。(FuncShar, FuncReconstruct, SharSp, KeySp)を次の1,…,4のように決めると、(FuncShar, FuncReconstruct, SharSp, KeySp)は秘密分散方式である。

1. SharSpを巡回群(Z/qZ)とする。

2. KeySpを巡回群(Z/qZ)とする。

3. KeySpの元cが入力されると、FuncSharは、(Z/qZ)の元a__1,…,a__{k-1}をランダムに選び、多項式f(u)をf(u)= c+ a__1x+ a__2x^2+ … +a__{k-1}x^{k-1} mod qと決め、s__1= f(1) mod q、 … 、s__n= f(n) mod qとし、(s__1, … ,s__n)を出力する。

4. ((i__1, s__{i__1}), … ,(i__k, s__{k}))が入力されると、FuncReconstructは多項式f(u)で、f(i__1)= s__{i__1} mod q、 … 、f(i__{k})= s__{i__k} mod qとなるものを選び、c= f(0) mod qとし、cを出力する。

[0024] [特別な秘密分散方式]

Sを有限集合とし、ΓをS上のアクセス構造とし、nをSの元の数とする。組(FuncShar, FuncReconstruct, FuncReShar, FuncVerShar, SharSp, KeySp)が次の性質1,…,4を満たすとき、(FuncShar, FuncReconstruct, FuncVerShar, SharSp, KeySp)はアクセス構造Γを持つ特別な秘密分散方式であるという。

1. (FuncShar, FuncReconstruct, SharSp, KeySp)は秘密分散である。

2. A={i__1,…,i__m}をΓの元とする。s__{i__1}, … ,s__{i__m}をSharSpの元とする。そしてl=n-mとする。SからAを除いた集合をB= {j__1, … ,j__l}とする。さらに、cをKeySpの元とする。このとき、cと(i__1, s__{i__1}), … ,(i__m, s__{i__m})をFuncReSharに入力すると、FuncReSharは、Sの元とSharSpの元の組(j__1, s__{j__1}), … ,(j__l, s__{j__l})を出力するか、または「reject」を出力する。

3. SharSpの元s__1, … ,s__nとKeySpの元cとが入力されると、FuncVerSharは文字列「accept」または文字列「reject」を出力する。

4. cをKeySpの元とする。cをFuncSharに入力したときの出力をs__1, … ,s__nとする。このときVerSharSp(s__1,…,s__n,c)=acceptが成立する。

[0025] [特別な秘密分散方式の例1]

n、S、Γを[アクセス構造の例1]で説明したように決める。さらに、qを自然数とする。(FuncShar, FuncReconstruct, SharSp, KeySp)を[秘密分散方式の例1]で説明したように決める。

1. A={i__1,…,i__m}をΓの元とする。s__{i__1}, … ,s__{i__m}をSharSpの元とする。そしてl=n-mとする。SからAを除いた集合をB= {j__1, … ,j__l}とする。さらに、cをKeySpの元とする。cと(i__1, s__{i__1}), … ,(i__m, s__{i__m})が入力されると、FuncReSharは、s__{j__1}, … ,s__{j__{l-1}}をランダムに選び、s__{j__l}= c-(s__1+ …+ s__{j__{l-1}}+ s__{j__{j+1}}+ …+ s__n) mod qとし、(j__1, s__{j__1}), … ,(j__l, s__{j__l})を出力する。

2. SharSpの元s__1, … ,s__nとKeySpの元cとが入力されると、FuncVerSharは、s__1+ …+ s__n mod qを計算し、c= s__1+ …+ s__n mod qなら「accept」を出力し、そうでないなら「reject」を出力する。

[0026] [特別な秘密分散方式の例2]

n、k、S、Γを[アクセス構造の例2]で説明したように決める。さらに、qを自然数とする。(FuncShar, FuncReconstruct, SharSp, KeySp)を[秘密分散方式の例2]で説明したように決める。(FuncReShar, FuncVerShar)を次の1,2のように決めると、(FuncShar, FuncReconstruct, FuncReShar, FuncVerShar, SharSp, KeySp)はアクセス構造Γを持つ特別な秘密分散方式である。

1. A={i__1,…,i__m}をΓの元とする。s__{i__1}, … ,s__{i__m}をSharSpの元とする。そしてl=n-mとする。SからAを除いた集合をB= {j__1, … ,j__l}とする。さらに、cをKeySpの元とする。cと(i__1, s__{i__1}), … ,(i__m, s__{i__m})が入力されると、FuncReSharは、多項式fでf(0)= c mod q, f(i__1)= s__{i__1} mod q, …, f(i__m)= s__{i__m} mod qとなるものを探す。

2. もしそのようなfが存在しなければ、FuncReSharは「reject」を出力する。そのようなfが存在すれば、FuncReSharは、s__{j__1}= f(j__1) mod q, …, s__{j__l}= f(j

__l) mod qを(j__1, s__{j__1}), … ,(j__l, s__{j__l})を出力する。

3. SharSpの元s__1, … ,s__nとKeySpの元cとが入力されると、FuncVerSharは、多項式fでf(0)= c mod q, f(1)= s__1 mod q, …, f(n)= s__n mod qとなるものを探す。そのようなfが存在すれば、FuncVerSharは「accept」を出力し、そうでなければ「reject」を出力する。

[0027] [コミットメント方式]

(C,FuncGen,FuncCom,FuncComVer)が次の1,…,6を満たすとき、(C, FuncGen, FuncCom, FuncComVer)をコミットメント方式という。

1. Cを集合とする。

2. FuncGen、FuncCom、FuncComVerは関数である。

3. FuncGenはDomParを出力する。

4. DomParとCの元cとが入力されると、FuncComはComとComStateを出力する。

5. DomPar、Cの元c、データCom'、およびデータComState'が入力されると、FuncComVerは文字列「accept」または文字列「reject」を出力する。

6. cをCの元とし、DomParをFuncGenの出力とする。DomParとcが入力したときのFuncComの出力をComとComStateとする。このときDomPar、c、Com、ComStateが入力されると、FuncComVerは「accept」を出力する。

[0028] [コミットメント方式の例]

qを素数とし、Gを位数qの有限群とする。(C, FuncGen, FuncCom, FuncComVer)を次のように決めると、(C, FuncGen, FuncCom, FuncComVer)はコミットメント方式になる。

1. C= (Z/qZ)とする。

2. FuncGenは、Gの2つの元g, hを選び、DomPar= (g, h)を出力する関数である。

3. DomPar= (g, h)とCの元cが入力されると、FuncComは、(Z/qZ)の元rをランダムに選び、Com= g^c h^rとComState= rを出力する。

4. DomPar= (g, h)とCの元cとComとComState=rが入力されると、FuncComVerは、g^c h^rを計算し、Com= g^c h^rなら「accept」を出力し、そうでなければ「reject」を出力する。

[0029] （実施形態1）

[装置構成]

本実施形態の証明検証システムの構成を説明する。

図1は本実施形態の証明検証システムの一構成例を示すブロック図である。

図1に示すように、証明検証システムは、証明を行うための証明装置100と、検証を行うための検証装置200とを有する。証明装置100および検証装置200は、通信ネットワーク（不図示）を介して通信可能に接続される。

証明装置100は、証明装置制御部102、証明装置記憶部101、および証明装置通信部（不図示）を有する。証明装置制御部102は、コミット計算手段111とレスポンス計算手段112とを有する。

検証装置200は、検証装置制御部202、検証装置記憶部201、および検証装置通信部（不図示）を有する。検証装置制御部202は、チャレンジ計算手段211と、レスポンス計算手段112とを有する。

[0030] [装置の実装]

各装置の制御部は、通信部の制御、記憶部の制御、およびデータの演算処理を行う。制御部には、プログラムにしたがって所定の処理を実行するCPU（Central Processing Unit）、およびプログラムを格納するためのメモリが設けられている。コミット計算手段111、レスポンス計算手段112、チャレンジ計算手段211および検証手段212は、CPUがプログラムを実行することで、コンピュータ内に仮想的に構成される。

通信ネットワークとして、例えば、インターネットやLAN（Local Area Network）を用いることができる。通信ネットワークは有線および無線のいずれでもよく、また有線と無線の組み合わせでもよい。図1では、装置間における情報の流れをわかりやすくするために、各装置の通信部を図に示すことを省略している。

記憶部にはハードディスクや半導体メモリなどを用いることが可能である。

[0031] [記号]

Gを加法群とし、HashをGに値を取るハッシュ関数とする。

i=1,…,nに対しR__iを関係式とし、y__iをデータとする。

i=1,…,nに対し（Y__i, X__i, FuncCommit__i, Hash__i, FuncResponse__i, Challe

ngeSpace__i, Verify__i, Simulator__i)を3-ムーブのパブリックコイン証明方式とする。

(FuncShar, FuncReconstruct, FuncReShar, FuncVerShar, SharSp, KeySp)を特別な秘密分散方式とする。

さらに、nを自然数とし、S={1,…,n}とする。ΓをS上のアクセス構造とし、A= {i__1, …, i__m}をΓの元とする。そして、p= n-mとする。SからAを除いた集合をB= {j__1, …, j__p}とする。

H__iをSharSpからChallengeSpace__iへのハッシュ関数とする。Simulator__2をゼロ知識証明シミュレータとする。ゼロ知識証明シミュレータは、従来技術と同様であるため、ここでは詳細な説明を省略する。

[0032] [実施形態1を使用する上での前提]

Xの元x__{i__1},…,x__{i__m}とYの元y__1,…,y__nが証明装置記憶部101に保存されている。このx__{i__1},…,x__{i__m}は、n個の秘密データのうちのm個の秘密データに相当する。

Yの元y__1,…,y__nが検証装置記憶部201に保存されている。

(Y__i, X__i, FuncCommit__i, Hash__i, FuncResponse__i, ChallengeSpace__i, Verify__i, Simulator__i)が関係式R__iに関する3ムーブのパブリックコイン証明方式である場合は、R(y__j, x__j)が満たされていることが望ましい。

[0033] [データ送受信]

証明装置100と検証装置200は、以下に説明するProt1-1, … ,Port1-10の順でデータ送受信を行う。図2はデータ送受信の動作を示すフローチャートである。

Prot1-1. 証明装置100はコミット計算手段111を行う(ステップ1101)。

Prot1-2. 証明装置100は証明装置通信部を使ってコミット計算手段111の出力を検証装置200に送信装置する(ステップ1102)。

Prot1-3. 検証装置200は検証装置通信部を使ってコミット計算手段111の出力を受信し、受信したデータを検証装置記憶部201に書き込む(ステップ1103)。

Prot1-4. 検証装置200はチャレンジ計算手段211を行う(ステップ1104)。

Prot1-5. 検証装置200は検証装置通信部を使ってチャレンジ計算手段211の出力

を証明装置100に送信する(ステップ1105)。

Prot1-6. 証明装置100は証明装置通信部を使ってチャレンジ計算手段211の出力を受信し、受信したデータを証明装置記憶部101に書き込む(ステップ1106)。

Prot1-7. 証明装置100はレスポンス計算手段112を行う(ステップ1107)。

Prot1-8. 証明装置100は証明装置通信部を使ってレスポンス計算手段112の出力を検証装置200に送信する(ステップ1108)。

Prot1-9. 検証装置200は検証装置通信部を使ってレスポンス計算手段112の出力を受信し、受信したデータを検証装置記憶部201に書き込む(ステップ1109)。

Prot1-10. 検証装置200は検証手段212を行う(ステップ1110)。

[0034] [コミット計算手段111]

証明装置100は、以下に説明する手段Com1-1, … ,Com1-8を順に行う。図3はコミット計算手段111の動作手順を示すフローチャートである。

Com1-1. 秘密データx__{i__1}, … ,x__{i__m}と集合Yの元y__1, … ,y__nを証明装置記憶部101から読み込む(ステップ1201)。

Com1-2. v=1,…,pに対してSharSpの元s__{j__v}をランダムに選びChallenge__{j__v}= H__i(s__{j__v})とする(ステップ1202)。

Com1-3. v=1,…,pに対してs__{j__v}を証明装置記憶部101に書き込む(ステップ1203)。

Com1-4. v=1,…,pに対して(Commit__{j__v}, Response__{j__v})= Simulator__2 (y__{j__v}, Challenge__{j__v})を計算する(ステップ1204)。

Com1-5. v=1,…,pに対して(Commit__{j__v}, Challenge__{j__v}, Response__{j__v})を証明装置記憶部101に書き込む(ステップ1205)。

Com1-6. u=1,…,mに対して(Commit__{i__u}, State__{i__u})= FuncCommit__{i__u} (x__{i__u}, y__{i__u})を計算する(ステップ1206)。

Com1-7. u=1,…,mに対してCommit__{i__u}, State__{i__u}を証明装置記憶部101に書き込む(ステップ1207)。

Com1-8. (Commit__1,…,Commit__n)を出力する(ステップ1208)。

[0035] [チャレンジ計算手段211]

検証装置200は、以下に説明する手段Cha1-1,…,Cha1-3を行う。図4はチャレンジ計算手段211の動作手順を示すフローチャートである。

Cha1-1. SharSpの元cをランダムに選ぶ(ステップ1301)。

Cha1-2. cを検証装置記憶部に書き込む(ステップ1302)。

Cha1-3. cを出力する(ステップ1303)。

[0036] [レスポンス計算手段112]

証明装置100は、以下に説明する手段Res1-1,…,Res1-7を順に行う。図5はレスポンス計算手段112の動作手順を示すフローチャートである。

Res1-1. Commit__1,…,Commit__nを証明装置記憶部101から読み込む(ステップ1401)。

Res1-2. cを証明装置記憶部101から読み込む(ステップ1402)。

Res1-3. v=1,…,pに対してState__{j__v}を証明装置記憶部101から読み込む(ステップ1403)。

Res1-4. FuncReSharにcと(j__1,s__{j__1}),…,(j__p,s__{j__p})を入力して出力(i__1, s__{i__1}), …, (i__m, s__{i__m})を得る(ステップ1404)。

Res1-5. v=1,…,pに対してChallenge__{i__v}=H__i(s__v)とする(ステップ1405)。

Res1-6. u=1,…,mに対してResponse__{i__u}= FuncResponse__{i__u} (y__{i__u}, Commit__{i__u}, Challenge__{i__u}, State__{i__u})とする(ステップ1406)。

Res1-7. s__1,…,s__nとResponse__1,…,Response__nを出力する(ステップ1407)。

[0037] [検証手段212]

検証装置200は、以下に説明する手段Ver1-1,…,Ver1-4を行う。図6は検証手段212の動作手順を示すフローチャートである。

Ver1-1. c,s__1,…,s__nとResponse__1,…,Response__nを検証装置記憶部201から読み込む(ステップ1501)。

Ver1-2. FuncVerShar(s__1,…,s__n,c)を計算し、FuncVerShar(s__1, …, s__n, c)= rejectならrejectを出力して終了する(ステップ1502)。

Ver1-3. i=1,…,nに対してChallenge__i=H__i(s__i)とする(ステップ1503)。

Ver1-4. i=1,…,nに対し、Verify(y__i, Commit__i, Challenge__i, Response__i)を

計算する。i=1，… ,nに対し、Verify(y__i，Commit__i，Challenge__i，Response__i)= acceptならacceptを出力して終了し、そうでなければrejectを出力して終了する(ステップ1504)。

[0038] 本実施形態の証明検証システムでは、上述のようにして、証明装置100がn個の秘密データのうちm個を保持していることを検証装置200に証明可能となる。そのため、文献1で開示された、2つの条件を必要とする知識の証明方式について、条件1を満たせばよい。

[0039] (実施形態2)

[装置構成]

本実施形態の証明検証システムの構成を説明する。なお、実施形態1と同様な構成についてはその詳細な説明を省略する。

図7は本実施形態の証明検証システムの一構成例を示すブロック図である。

図7に示すように、証明検証システムは、証明を行うための証明装置100と、検証を行うための検証装置200とを有する。証明装置100および検証装置200は、通信ネットワーク(不図示)を介して通信可能に接続される。

[0040] 証明装置100は、証明装置制御部102、証明装置記憶部101、および証明装置通信部(不図示)を有する。証明装置制御部102は、コミット計算手段114と、レスポンス計算手段115とを有する。

検証装置200は、検証装置制御部202、検証装置記憶部201、および検証装置通信部(不図示)を有する。検証装置制御部202は、チャレンジコミット計算手段210と、チャレンジ計算手段213と、検証手段214とを有する。

[0041] [装置の実装]

各装置の制御部には、プログラムにしたがって所定の処理を実行するCPUと、プログラムを格納するためのメモリとが設けられている。コミット計算手段114、レスポンス計算手段115、チャレンジコミット計算手段210、チャレンジ計算手段213および検証手段214は、CPUがプログラムを実行することで、コンピュータ内に仮想的に構成される。その他の構成については、実施形態1と同様であるため、詳細な説明を省略する。

[0042] [記号]

実施形態1と同じ記号を使う。C= ChallengeSpとし、(C, FuncGen, FuncCom, FuncComVer)をコミットメント方式とする。

[0043] [実施形態2を使用する上での前提]

実施形態1と同じ前提を課す。加えて次の前提1,2を課す。

1. ChallengeSpは群である。

2. FuncGenの出力DomParを証明装置100と検証装置200が事前に共有している。

[0044] [データ送受信]

証明装置100と検証装置200は、以下に説明するProt2-1,…,Prot2-13順でデータ送受信を行う。図8および図9は、データ送受信の動作手順を示すフローチャートである。

Prot2-1. 検証装置200はチャレンジコミット計算手段210を行う(ステップ2101)。

Prot2-2.検証装置200は検証装置通信部を使ってチャレンジコミット計算手段210の出力を証明装置100に送信する(ステップ2102)。

Prot2-3.証明装置100は証明装置通信部を使ってチャレンジコミット計算手段210の出力を受信し、受信したデータを証明装置記憶部101に書き込む(ステップ2103)。

Prot2-4.証明装置100はコミット計算手段114を行う(ステップ2104)。

Prot2-5.証明装置100は証明装置通信部を使ってコミット計算手段114の出力を検証装置200に送信装置する(ステップ2105)。

Prot2-6.検証装置200は検証装置通信部を使ってコミット計算手段114の出力を受信し、受信したデータを検証装置記憶部201に書き込む(ステップ2106)。

Prot2-7.検証装置200はチャレンジ計算手段213を行う(ステップ2107)。

Prot2-8.検証装置200は検証装置通信部を使ってチャレンジ計算手段213の出力を証明装置100に送信する(ステップ2108)。

Prot2－9. 証明装置100は証明装置通信部を使ってチャレンジ計算手段213の出力を受信し、受信したデータを証明装置記憶部101に書き込む(ステップ2109)。

Prot2-10.証明装置100はレスポンス計算手段115を行う(ステップ2110)。

Prot2-11.証明装置100は証明装置通信部を使ってレスポンス計算手段115の出力を検証装置200に送信する(ステップ2111)。

Prot2-12.検証装置200は検証装置通信部を使ってレスポンス計算手段115の出力を受信し、受信したデータを検証装置記憶部201に書き込む(ステップ2112)。

Prot2-13.検証装置200は検証手段214を行う(ステップ2113)。

[0045] [チャレンジコミット計算手段210]

検証装置200は、以下に説明する手段ChaCom2-1,…,ChaCom2-5を行う。図10はチャレンジコミット計算手段210の動作手順を示すフローチャートである。

ChaCom2-1. DomParを検証装置記憶部201から読み込む(ステップ2201)。

ChaCom2-2. ChallengeSpの元c__1をランダムに選ぶ(ステップ2202)。

ChaCom2-3. (Com,ComState)=FuncCom(DomPar,c__1)を計算する(ステップ2203)。

ChaCom2-4. c__1、Com、ComStateを検証装置記憶部201に書き込む(ステップ2204)。

ChaCom2-5. Comを出力する(ステップ2205)。

[0046] [コミット計算手段114]

証明装置100は、以下に説明する手段Com2-1,…,Com2-4を行う。図11はコミット計算手段114の動作手順を示すフローチャートである。

Com2-1. 実施形態1のCom1-1,…,Com1-7を行う(ステップ2301)。

Com2-2. ChallengeSpの元c__2をランダムに選ぶ(ステップ2302)。

Com2-3. c__2を証明装置記憶部101に書き込む(ステップ2303)。

Com2-4. (c__2,Commit__1,…,Commit__n)を出力する(ステップ2304)。

[0047] [チャレンジ計算手段213]

検証装置200は、以下に説明する手段Cha2-1,Cha2-2を行う。図12はチャレンジ計算手段213の動作手順を示すフローチャートである。

Cha2-1. 検証装置記憶部201からc__1、ComStateを読み込む(ステップ2401)。

Cha2-2. c__1、ComStateを出力する(ステップ2402)。

[0048] [レスポンス計算手段115]

証明装置100は、以下に説明する手段Res2-1,…,Res2-7を順に行う。図13はレスポンス計算手段115の動作手順を示すフローチャートである。

Res2-1. Commit__1,…,Commit__n, c__1,c__2,DomPar,Com,ComStateを証明装置記憶部101から読み込む(ステップ2501)。

Res2-2. FuncComVer(DomPar,c__1,Com,ComState)を計算し、FuncComVer (DomPar, c__1, Com, ComState)= rejectならrejectを出力して終了する(ステップ2502)。

Res2-3. c=c__1c__2を計算する(ステップ2503)。

Res2-4. cを用いて実施形態1のRes1-3,…,Res1-7を行う(ステップ2504)。

[0049] [検証手段214]

検証装置200は、以下に説明する手段Ver2-1,…,Ver2-5を行う。図14は検証手段214の動作手順を示すフローチャートである。

Ver2-1. c__1,c__2,s__1,…,s__nとResponse__1,…,Response__nを検証装置記憶部201から読み込む(ステップ2601)。

Ver2-2. c=c__1c__2を計算する(ステップ2602)。

Ver2-3. FuncVerShar(s__1,…,s__n,c)を計算し、FuncVerShar(s__1,…,s__n,c)=rejectならrejectを出力して終了する(ステップ2603)。

Ver2-4. i=1,…,nに対してChallenge__i=H__i(s__i)とする(ステップ2604)。

Ver2-5. i=1,…,nに対し、Verify(y__i, Commit__i, Challenge__i, Response__i)を計算する。i=1,…,nに対し、Verify(y__i, Commit__i, Challenge__i, Response__i)=acceptならacceptを出力して終了し、そうでなければrejectを出力して終了する(ステップ2605)。

[0050] 本実施形態では、実施形態1の効果の他に、検証装置200が証明装置100から情報を受信する前にチャレンジコミット計算手段210を実行することで、データ選択の際に証明装置100からの情報による影響が抑制され、証明装置100に格納された秘密情報の秘匿性が向上する。

[0051] (実施形態3)

[装置構成]

本実施形態の証明検証システムの構成を説明する。なお、実施形態1と同様な構成についてはその詳細な説明を省略する。

図15は本実施形態の証明検証システムの一構成例を示すブロック図である。

図15に示すように、証明検証システムは、証明を行うための証明装置100と、検証を行うための検証装置200とを有する。証明装置100および検証装置200は、通信ネットワーク(不図示)を介して通信可能に接続される。

[0052] 証明装置100は、証明装置制御部102、証明装置記憶部101、および証明装置通信部(不図示)を有する。証明装置制御部102には証明文作成手段110が設けられている。証明文作成手段110は、コミット計算手段116と、チャレンジ計算手段113と、レスポンス計算手段117とを有する。

検証装置200は、検証装置制御部202、検証装置記憶部201、および検証装置通信部(不図示)を有する。検証装置制御部202は、検証手段215を有する構成である。

[0053] [装置の実装]

各装置の制御部には、プログラムにしたがって所定の処理を実行するCPUと、プログラムを格納するためのメモリとが設けられている。コミット計算手段116、レスポンス計算手段117、チャレンジ計算手段113および検証手段215は、CPUがプログラムを実行することで、コンピュータ内に仮想的に構成される。その他の構成については、実施形態1と同様であるため、詳細な説明を省略する。

[0054] [記号]

実施形態1と同じ記号を使う。HashChaをSharSpに値を取るハッシュ関数とする。さらに、Mをビット列とする。証明装置100と検証装置200は事前にMを共有しているとする。どのような方法でMを共有したのかは問わない。Mは予備の入力で、目的に応じてMを様々なものにセットする。Mとして空列を選んでもよい。Mにセットするものとして、例えば、次の1,…,6があり得る。

1. (y__1,…,y__n)
2. 証明者のID
3. 検証者のID
4. 時刻
5. 何らかのメッセージ
6. 1,…,5の全てまたは一部を連結したもの

[実施形態3を使用する上での前提]

Xの元x__{i__1},…,x__{i__m}、Yの元y__1,…,y__n、およびMが証明装置記憶部101に保存されている。Yの元y__1, …, y__n、およびMが検証装置記憶部201に保存されている。

[0055] [データ送受信]

証明装置100と検証装置200は、以下に説明するProt3-1,…,Prot3-4の順でデータ送受信を行う。図16はデータ送受信の動作手順を示すフローチャートである。

Prot3-1. 証明装置100は証明文作成手段110を行う(ステップ3101)。

Prot3-2. 証明装置100は証明装置通信部を使って証明文作成手段110の出力を検証装置200に送信装置する(ステップ3102)。

Prot3-3. 検証装置200は検証装置通信部を使って証明文作成手段110の出力を受信し、受信したデータを検証装置記憶部201に書き込む(ステップ3103)。

Prot3-4. 検証装置200は検証手段215を行う(ステップ3104)。

[0056] [証明文作成手段110]

証明装置100は、以下に説明する手段GenPf3-1,GenPf3-2を順に行う。図17は証明文作成手段110の動作手順を示すフローチャートである。

GenPf3-1. [コミット計算手段116]、[チャレンジ計算手段113]、[レスポンス計算手段117]を順に行う(ステップ3201)。

GenPf3-2. Commit__1,…,Commit__n、s__1,…,s__n、Response__1,…,Response__nを出力する(ステップ3202)。

[コミット計算手段116]

証明装置100は、実施形態1のコミット計算手段111のCom1-1,…,Com1-7を行う。

[チャレンジ計算手段113]

証明装置100は以下の手段Cha3-1、Cha3-2を行う(図18)。

Cha3-1. Commit__1,…,Commit__n,Mを証明装置記憶部101から読み込む。(ステップ3301)

Cha3-2. c=HashCha(Commit__1,…,Commit__n,M)を計算し、cを証明装置記憶部101に書き込む。(ステップ3302)

[レスポンス計算手段117]

証明装置100は、実施形態1のレスポンス計算手段112のRes1-1,…,Res1-6を行う。

[0057] [検証手段215]

検証装置200は、以下に説明する手段Ver3-1,…,Ver3-5を行う。図19は検証手段215の動作手順を示すフローチャートである。

Ver3-1. s__1,…,s__n, Commit__1,…,Commit__n, M,Response__1,…,Response__nを検証装置記憶部201から読み込む(ステップ3401)。

Ver3-2. c=HashCha(Commit__1,…,Commit__n,M)を計算する(ステップ3402)。

Ver3-3. FuncVerShar(s__1,…,s__n,c)を計算し、FuncVerShar (s__1, …, s__n, c)= rejectならrejectを出力して終了する(ステップ3403)。

Ver3-4. i=1,…,nに対してChallenge__i=H__i(s__i)とする(ステップ3404)。

Ver3-5. i=1,…,nに対し、Verify(y__i, Commit__i, Challenge__i, Response__i)を計算する。i=1,…,nに対し、Verify(y__i, Commit__i, Challenge__i, Response__i)= acceptならacceptを出力して終了し、そうでなければrejectを出力して終了する(ステップ3405)。

[0058] 本実施形態では、証明装置100におけるチャレンジ計算手段113の動作がコミット計算手段116の影響を受けないように改善したので、実施形態1の効果の他に、証明装置100の秘密情報の秘匿性が向上するとともに、証明装置100と検証装置200との通信回数を低減できる。

[0059] なお、実施形態1から3で説明した証明検証方法を、コンピュータに実行させるためのプログラムに適用してもよい。

[0060] また、本発明の証明検証システム、証明装置および検証装置と、証明検証方法と、その方法をコンピュータに実行させるためのプログラムについて、知識の証明方式を様々な暗号プロトコルを作る際の部品として用いることができ、認証方式や署名方式に応用できる。

[0061] なお、本発明は上記実施形態に限定されることなく、発明の範囲内で種々の変形が可能であり、それらも本発明の範囲内に含まれることはいうまでもない。

請求の範囲

[1] n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータが格納された証明装置記憶部、ならびに、前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n−m)とし、前記証明装置記憶部から前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを外部に送信し、外部からチャレンジ値cを受信すると、該チャレンジ値cおよび前記s__{j__1}，…，s__{j__p}から残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを外部に送信する証明装置制御部を含む証明装置と、

前記証明装置と通信可能に接続され、複数の乱数および前記集合の元y__1，…，y__nが格納された検証装置記憶部、ならびに、前記証明装置からコミット値Commit__1,…,Commit__nを受信すると、前記複数の乱数から選択したcをチャレンジ値として該証明装置に送信し、前記巡回群の元s__1，…，s__nおよびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該s__1，…，s__nが前記チャレンジ値cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない検証装置制御部を含む検証装

置と、

を有する証明検証システム。

[2] n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、n個の元を含む巡回群のデータおよび複数の元を含む群のデータが格納された証明装置記憶部、ならびに、前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n-m)とし、外部からコミット値Comを受信すると、該コミット値Comを前記証明装置記憶部に格納し、該証明装置記憶部から前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、前記複数の元を含む群からランダムに元c__2を選び、該元c__2および該Commit__1，…，Commit__nを外部に送信し、前記コミット値Comを算出するための、前記複数の元を含む群の元c__1を外部から受信すると、該コミット値Comが該元c__1の正当なコミットメントであるか否かを検証し、正当でない場合、証明の続行を拒否し、正当である場合、前記c__1と前記c__2を乗算した値cを求め、前記s__{j__1}，…，s__{j__p}と該値cから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__i、および前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信する証明装置制御部を含む証明装置と、

前記証明装置と通信可能に接続され、複数の元を含む群のデータが格納された検証装置記憶部、ならびに、前記複数の元を含む群からランダムに元c__1を選び、該元c__1のコミット値Comを算出して前記証明装置に送信し、前記複数の元を含む群

の元c__2、およびコミット値Commit__1，…，Commit__nを前記証明装置から受信すると、前記元c__1を該証明装置に送信し、巡回群の元s__1，…，s__n、およびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該c__1と該c__2を乗算した値cを求め、該s__1，…，s__nが該値cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない検証装置制御部を含む検証装置と、

を有する証明検証システム。

[3] n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータが格納された証明装置記憶部、ならびに、前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n-m)とし、前記証明装置記憶部から前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、該Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、該ハッシュ値cと前記s__{j__1}，…，s__{j__p}とから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1，…，nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを外部に送信する証明装置制御部を含む証明装置と、

前記証明装置と通信可能に接続され、前記集合の元y__1，…，y__nが格納された検証装置記憶部、ならびに、前記巡回群の元s__1，…，s__n、コミット値Commit__1，…，Commit__n、およびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、該s__1，…，s__nが該ハッシュ値cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない検証装置制御部を含む検証装置と、

を有する証明検証システム。

[4] 検証装置と通信可能に接続され、該検証装置に対して秘密データの保持を証明する証明装置であって、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータが格納された記憶部と、

前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n-m)とし、前記記憶部から前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを前記検証装置に送信し、前記検証装置からチャレンジ値cを受信すると、該チャレンジ値cおよび前記s__{j__1}，…，s__{j__p}から残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ

値Challenge__iとし、前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信する制御部と、

を有する証明装置。

[5] 証明装置と通信可能に接続され、該証明装置が発行する証明文を検証する検証装置であって、

複数の乱数および集合の元y__1，…，y__nが格納された記憶部と、

前記証明装置からコミット値Commit__1,…,Commit__nを受信すると、前記複数の乱数から選択したcをチャレンジ値として該証明装置に送信し、巡回群の元s__1，…，s__nおよびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該s__1，…，s__nが前記チャレンジ値cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない制御部と、

を有する検証装置。

[6] 検証装置と通信可能に接続され、該検証装置に対して秘密データの保持を証明する証明装置であって、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、n個の元を含む巡回群のデータおよび複数の元を含む群のデータが格納された記憶部と、

前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n−m)とし、前記検証装置からコミット値Comを受信すると、該Comを前記記憶部に格納し、該記憶部から前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレ

スポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、前記複数の元を含む群からランダムに元c__2を選び、該c__2および該Commit__1，…，Commit__nを前記検証装置に送信し、前記Comを算出するための、前記複数の元を含む群の元c__1を該検証装置から受信すると、該コミット値Comが該c__1の正当なコミットメントであるか否かを検証し、正当でない場合、証明の続行を拒否し、正当である場合、前記c__1と前記c__2を乗算したcを求め、前記s__{j__1}，…，s__{j__p}と該cから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__i、および前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信する制御部と、

を有する証明装置。

[7] 証明装置と通信可能に接続され、該証明装置が発行する証明文を検証する検証装置であって、

複数の元を含む群のデータおよび集合の元y__1，…，y__nが格納された記憶部と、

前記複数の元を含む群からランダムに元c__1を選び、該元c__1のコミット値Comを算出して前記証明装置に送信し、前記複数の元を含む群の元c__2、およびコミット値Commit__1，…，Commit__nを前記証明装置から受信すると、前記c__1を該証明装置に送信し、巡回群の元s__1，…，s__n、およびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該c__1と該c__2を乗算したcを求め、該s__1，…，s__nが該cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を

受理し、正当でない場合、該証明装置による証明を受理しない制御部と、
を有する検証装置。

[8] 検証装置と通信可能に接続され、該検証装置に対して秘密データの保持を証明する証明装置であって、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータが格納された記憶部と、

前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n-m)とし、前記記憶部から前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、該Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、該cと前記s__{j__1}，…，s__{j__p}とから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1，…，nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信する制御部と、
を有する証明装置。

[9] 証明装置と通信可能に接続され、該証明装置が発行する証明文を検証する検証装置であって、

集合の元y__1，…，y__nが格納された記憶部と、

巡回群の元s__1，…，s__n、コミット値Commit__1，…，Commit__n、およびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該Commi

t__1，…，Commit__nを含むデータのハッシュ値をcとし、該s__1，…，s__nが該cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組（Commit__i，Challenge__i，Response__i）による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない制御部と、

を有する検証装置。

[10] 証明文を発行する証明装置と、該証明装置に通信可能に接続され、該証明文を検証する検証装置とによる証明検証方法であって、

前記証明装置が、n個の秘密データのうちm個（m＜n）の秘密データx__{i__1}，…,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータを格納し、

前記検証装置が、複数の乱数および前記集合の元y__1，…，y__nを格納し、

前記証明装置が、前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p（p= n−m）とし、前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組（Commit__{j__1}，Response__{j__1}），…，（Commit__{j__p}，Response__{j__p}）を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを前記検証装置に送信し、

前記検証装置が、前記証明装置からコミット値Commit__1,…,Commit__nを受信すると、前記複数の乱数から選択したcをチャレンジ値として該証明装置に送信し、

前記証明装置が、前記検証装置から前記チャレンジ値cを受信すると、該チャレンジ値cおよび前記s__{j__1}，…，s__{j__p}から残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし

、前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、s__1，…，s__nとResponse__1，…，Response__nを前記検証装置に送信し、

前記検証装置が、前記元s__1，…，s__nおよびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該s__1，…，s__nが前記チャレンジ値cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない、証明検証方法。

[11] 証明文を発行する証明装置と、該証明装置に通信可能に接続され、該証明文を検証する検証装置とによる証明検証方法であって、

前記証明装置が、n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，…,x__{i__m}、集合の元y__1，…，y__n、n個の元を含む巡回群のデータおよび複数の元を含む群のデータを格納し、

前記検証装置が、前記複数の元を含む群のデータを格納し、

前記検証装置が、前記複数の元を含む群からランダムに元c__1を選び、該元c__1のコミット値Comを算出して前記証明装置に送信し、

前記証明装置が、前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n−m)とし、前記検証装置から前記コミット値Comを受信すると、該Comを格納し、前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Com

mit__nを求め、前記複数の元を含む群からランダムに元c__2を選び、該c__2および該Commit__1，…，Commit__nを前記検証装置に送信し、

前記検証装置は、前記元c__2および前記コミット値Commit__1，…，Commit__nを前記証明装置から受信すると、前記c__1を該証明装置に送信し、

前記証明装置が、前記元c__1を前記検証装置から受信すると、前記コミット値Comが該c__1の正当なコミットメントであるか否かを検証し、正当でない場合、証明の続行を拒否し、正当である場合、前記c__1と前記c__2を乗算したcを求め、前記s__{j__1}，…，s__{j__p}と該cから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__i、および前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信し、

前記検証装置が、前記元s__1，…，s__n、および前記レスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該c__1と該c__2を乗算したcを求め、該s__1，…，s__nが該cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない、証明検証方法。

[12] 証明文を発行する証明装置と、該証明装置に通信可能に接続され、該証明文を検証する検証装置とによる証明検証方法であって、

前記証明装置が、n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，…,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータを格納し、

前記検証装置が、前記集合の元y__1，…，y__nを格納し、

前記証明装置が、前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n−m)とし、前記巡回群の元s__{j__1}，…，

s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、該Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、該cと前記s__{j__1}，…，s__{j__p}とから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1，…，nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信し、

前記検証装置は、前記元s__1，…，s__n、前記コミット値Commit__1，…，Commit__n、および前記レスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、該s__1，…，s__nが該cから正当な方法で生成された秘密分散であるか否かを検証し、正当であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない、証明検証方法。

[13] 検証装置と通信可能に接続され、該検証装置に対して秘密データの保持を証明するコンピュータに実行させるためのプログラムであって、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータを格納し、

前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n−m)とし、前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ

値Challenge__{j__v}とし、

v=1，…，pに対して前記y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、

u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを前記検証装置に送信し、

前記検証装置からチャレンジ値cを受信すると、該チャレンジ値cおよび前記s__{j__1}，…，s__{j__p}から残りの元s__{i__1}，…，s__{i__m}を生成し、

i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、

前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信する処理を前記コンピュータに実行させるためのプログラム。

[14] 証明装置と通信可能に接続され、該証明装置が発行する証明文を検証するコンピュータに実行させるためのプログラムであって、

複数の乱数および集合の元y__1，…，y__nを格納し、

前記証明装置からコミット値Commit__1,…,Commit__nを受信すると、前記複数の乱数から選択したcをチャレンジ値として該証明装置に送信し、

巡回群の元s__1，…，s__nおよびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該s__1，…，s__nが前記チャレンジ値cから正当な方法で生成された秘密分散であるか否かを検証し、

前記s__1，…，s__nが前記チャレンジ値cから正当な方法で生成された秘密分散であれば、i=1，…，nに対して前記s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない処理を前記コンピュータに実行させるためのプログラム。

[15] 検証装置と通信可能に接続され、該検証装置に対して秘密データの保持を証明するコンピュータに実行させるためのプログラムであって、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、n個の元を含む巡回群のデータおよび複数の元を含む群のデータを格納し、

前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n-m)とし、前記検証装置からコミット値Comを受信すると、該Comを格納し、

前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、

v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、

u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、前記複数の元を含む群からランダムに元c__2を選び、該c__2および該Commit__1，…，Commit__nを前記検証装置に送信し、

前記Comを算出するための、前記複数の元を含む群の元c__1を前記検証装置から受信すると、該コミット値Comが該c__1の正当なコミットメントであるか否かを検証し、

前記コミット値Comが前記c__1の正当なコミットメントでない場合、証明の続行を拒否し、正当なコミットメントである場合、前記c__1と前記c__2を乗算したcを求め、前記s__{j__1}，…，s__{j__p}と該cから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1,…,nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__i、および前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信する処理を前記コンピュータに実行させるためのプログラム。

[16] 証明装置と通信可能に接続され、該証明装置が発行する証明文を検証するコンピュータに実行させるためのプログラムであって、

複数の元を含む群のデータおよび集合の元y__1，…，y__nを格納し、

前記複数の元を含む群からランダムに元c__1を選び、該元c__1のコミット値Comを算出して前記証明装置に送信し、

前記複数の元を含む群の元c__2、およびコミット値Commit__1，…，Commit__nを前記証明装置から受信すると、前記c__1を該証明装置に送信し、

巡回群の元s__1，…，s__n、およびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該c__1と該c__2を乗算したcを求め、該s__1，…，s__nが該cから正当な方法で生成された秘密分散であるか否かを検証し、

前記s__1，…，s__nが前記cから正当な方法で生成された秘密分散であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない処理を前記コンピュータに実行させるためのプログラム。

[17] 検証装置と通信可能に接続され、該検証装置に対して秘密データの保持を証明するコンピュータに実行させるためのプログラムであって、

n個の秘密データのうちm個(m＜n)の秘密データx__{i__1}，… ,x__{i__m}、集合の元y__1，…，y__n、およびn個の元を含む巡回群のデータを格納し、

前記n個の秘密データに対する識別子i__1，…，i__mのいずれとも異なる識別子をj__1，…，j__p(p= n−m)とし、前記巡回群の元s__{j__1}，…，s__{j__p}をランダムに選び、v=1，…，pに対して各s__{j__v}のハッシュ関数による値をチャレンジ値Challenge__{j__v}とし、

v=1，…，pに対して前記元y__{j__v}と前記Challenge__{j__v}を入力とするゼロ知識証明のシミュレーションを行うことで、コミット値とレスポンス値の組(Commit__{j__1}，Response__{j__1})，…，(Commit__{j__p}，Response__{j__p})を生成し、

u=1，…，mに対してx__{i__u}とy__{i__u}を用いてコミット値Commit__{i__u}を生

成し、前記Commit__{j__1}，…，Commit__{j__p}および前記Commit__{i__1}，…，Commit__{i__m}からなるコミット値Commit__1，…，Commit__nを求め、該Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、前記cと前記s__{j__1}，…，s__{j__p}とから残りの元s__{i__1}，…，s__{i__m}を生成し、i=1，…，nに対して各s__iにそのハッシュ値をチャレンジ値Challenge__iとし、前記y__i、前記Commit__iおよび前記Challenge__iを使ってレスポンス値Response__iを算出し、元s__1，…，s__nとレスポンス値Response__1，…，Response__nを前記検証装置に送信する処理を前記コンピュータに実行させるためのプログラム。

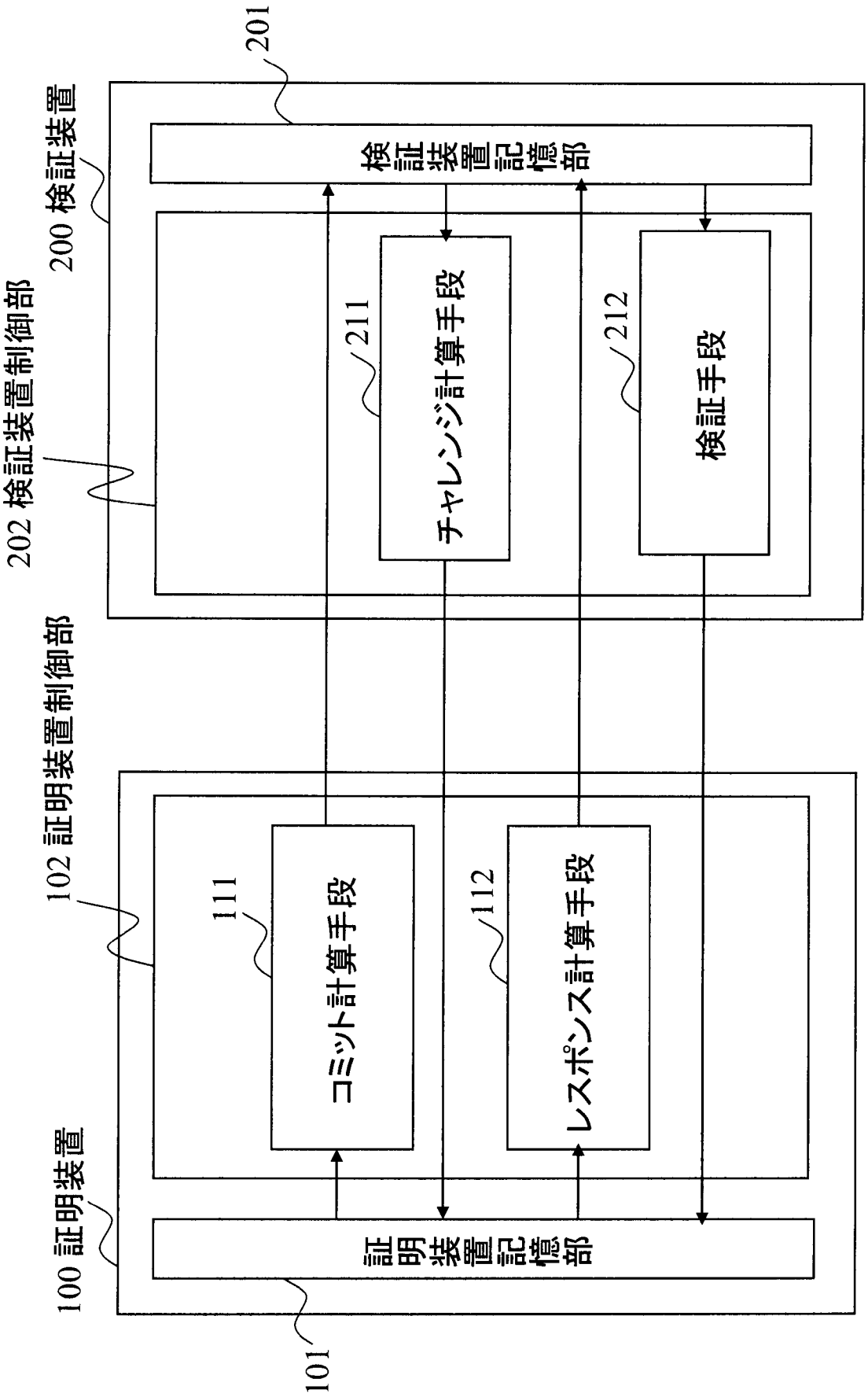
[18] 証明装置と通信可能に接続され、該証明装置が発行する証明文を検証するコンピュータに実行させるためのプログラムであって、

集合の元y__1，…，y__nを格納し、

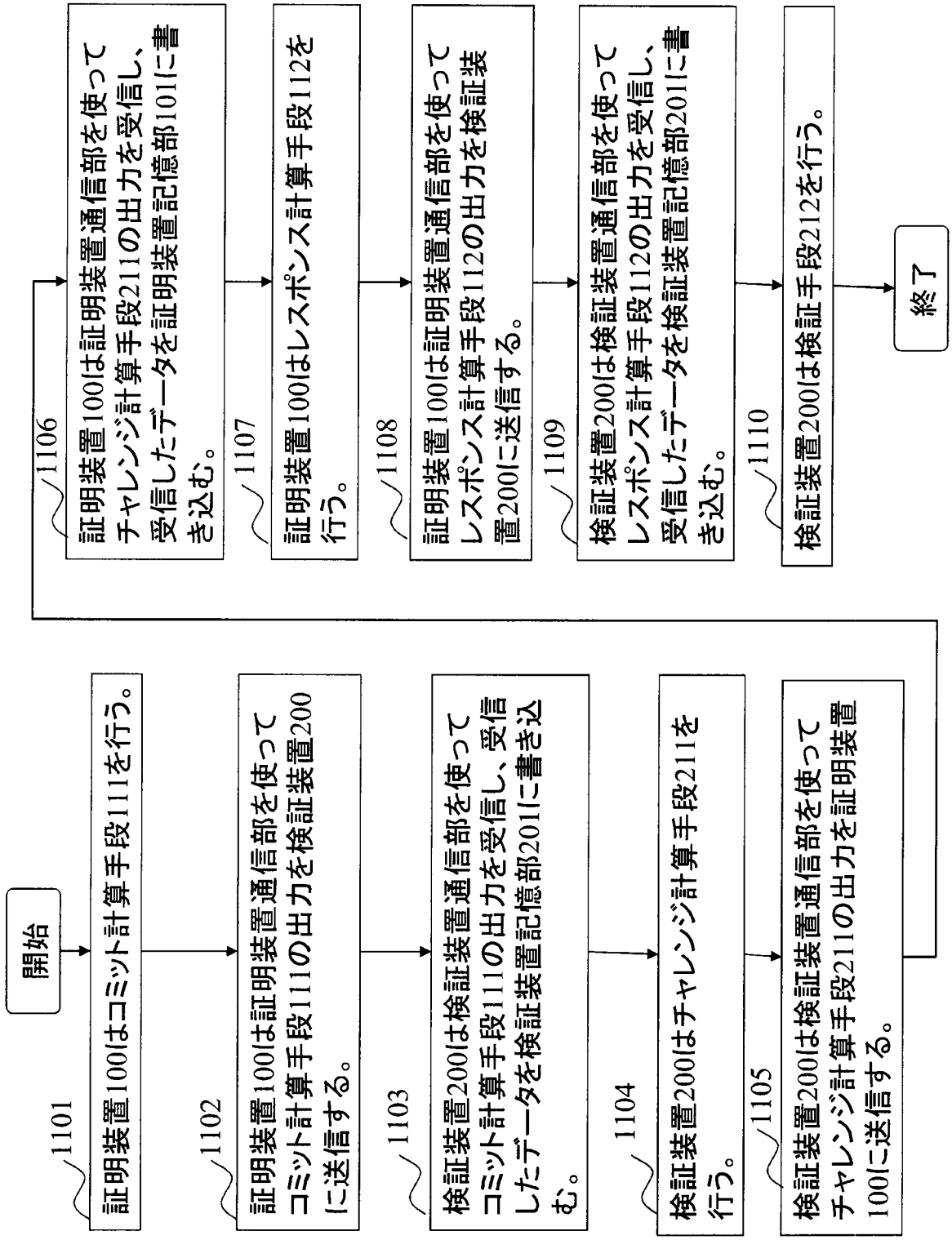
巡回群の元s__1，…，s__n、コミット値Commit__1，…，Commit__n、およびレスポンス値Response__1，…，Response__nを前記証明装置から受信すると、該Commit__1，…，Commit__nを含むデータのハッシュ値をcとし、該s__1，…，s__nが該cから正当な方法で生成された秘密分散であるか否かを検証し、

前記s__1，…，s__nが前記cから正当な方法で生成された秘密分散であれば、i=1，…，nに対して前記元s__iのハッシュ値をチャレンジ値Challenge__iとし、組(Commit__i，Challenge__i，Response__i)による証明文が前記y__iの正当な証明文であるか否かを検証し、正当である場合、前記証明装置による証明を受理し、正当でない場合、該証明装置による証明を受理しない処理を前記コンピュータに実行させるためのプログラム。

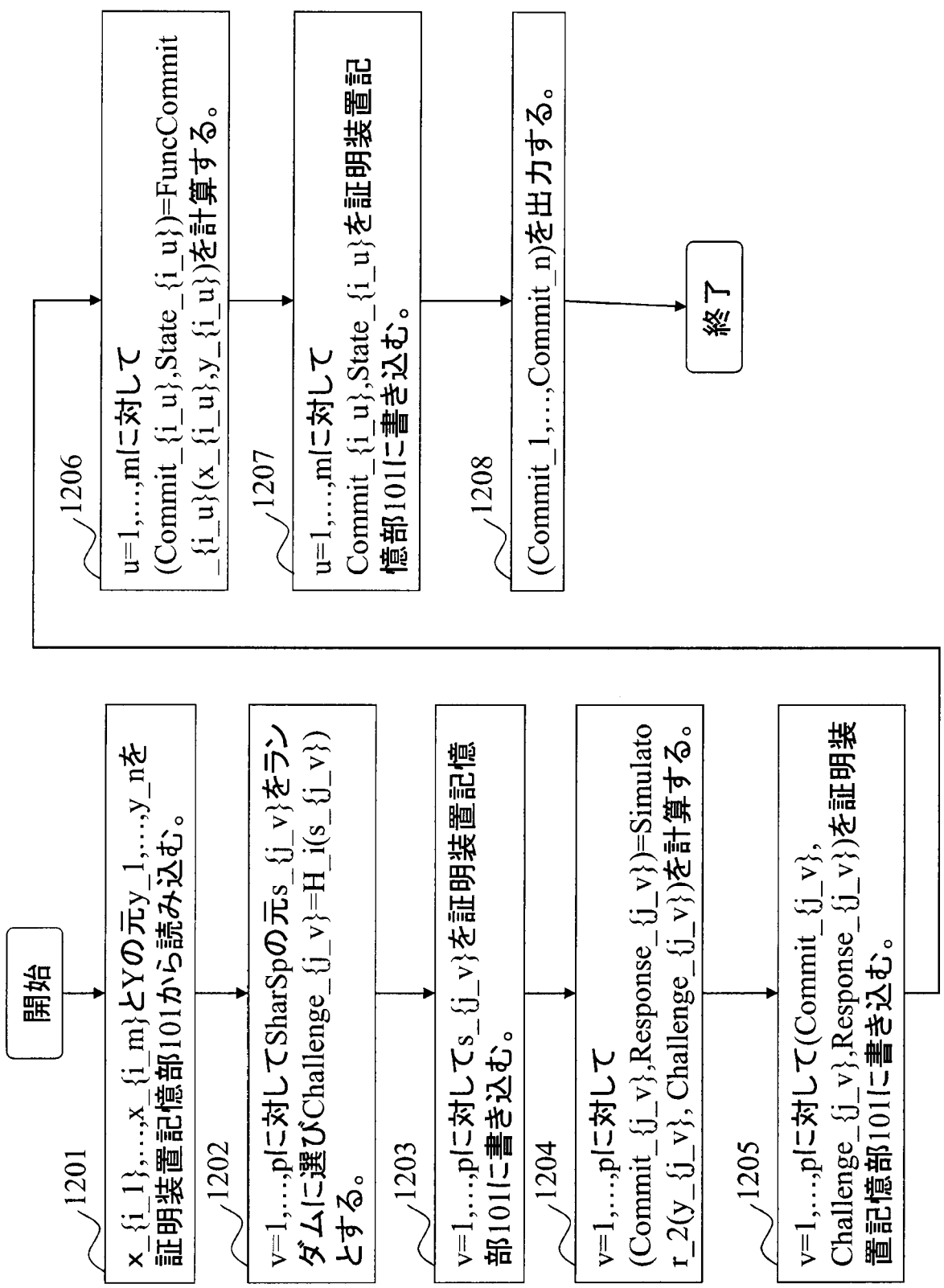
[図1]


201
200 検証装置
検証装置記憶部
202 検証装置制御部
211
チャレンジ計算手段
212
検証手段
102 証明装置制御部
111
コミット計算手段
112
レスポンス計算手段
100 証明装置
証明装置記憶部
101

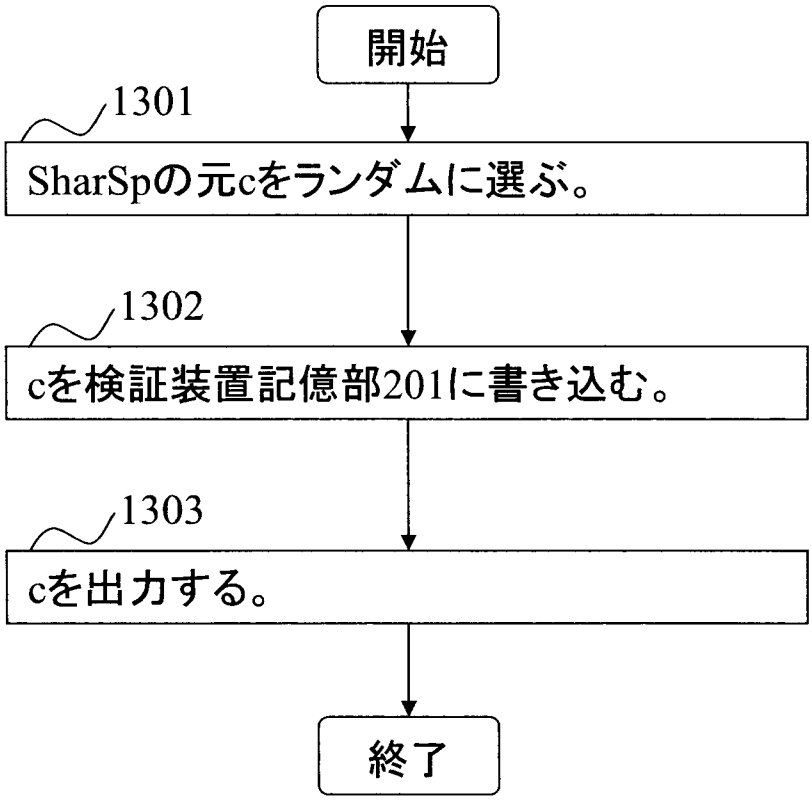
[図2]


開始
1101
証明装置100はコミット計算手段111を行う。
1102
証明装置100は証明装置通信部を使ってコミット計算手段111の出力を検証装置200に送信する。
1103
検証装置200は検証装置通信部を使ってコミット計算手段111の出力を受信し、受信したデータを検証装置記憶部201に書き込む。
1104
検証装置200はチャレンジ計算手段211を行う。
1105
検証装置200は検証装置通信部を使ってチャレンジ計算手段211の出力を証明装置100に送信する。
1106
証明装置100は証明装置通信部を使ってチャレンジ計算手段211の出力を受信し、受信したデータを証明装置記憶部101に書き込む。
1107
証明装置100はレスポンス計算手段112を行う。
1108
証明装置100は証明装置通信部を使ってレスポンス計算手段112の出力を検証装置200に送信する。
1109
検証装置200は検証装置通信部を使ってレスポンス計算手段112の出力を受信し、受信したデータを検証装置記憶部201に書き込む。
1110
検証装置200は検証手段212を行う。
終了

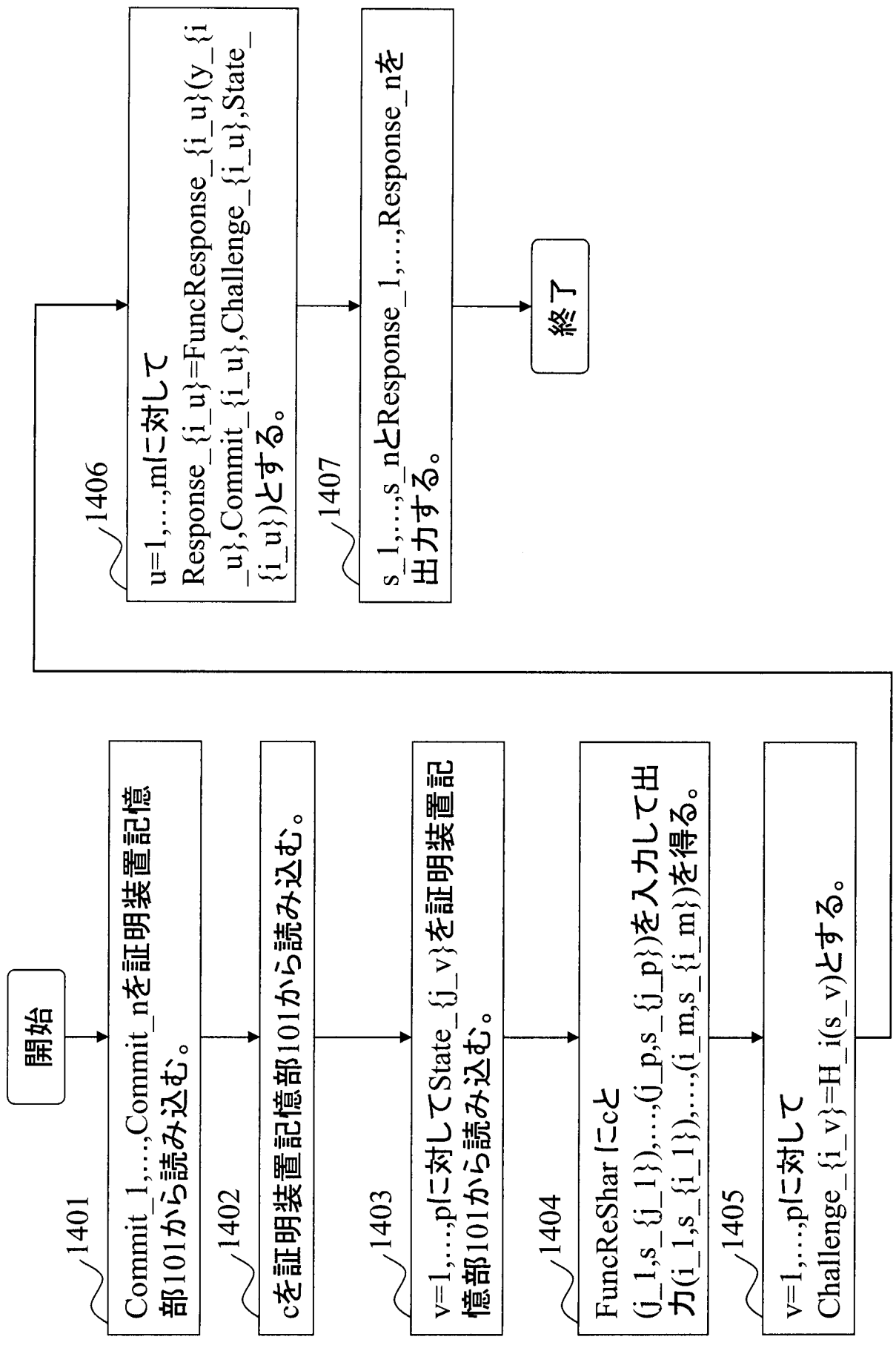
[図3]


開始
1201
x_{i_1},...,x_{i_m}とYの元y_1,...,y_nを証明装置記憶部101から読み込む。
1202
v=1,...,pに対してSharSpの元s_{j_v}をランダムに選びChallenge_{j_v}=H_i(s_{j_v})とする。
1203
v=1,...,pに対してs_{j_v}を証明装置記憶部101に書き込む。
1204
v=1,...,pに対して(Commit_{j_v},Response_{j_v})=Simulator_2(y_{j_v}, Challenge_{j_v})を計算する。
1205
v=1,...,pに対して(Commit_{j_v}, Challenge_{j_v},Response_{j_v})を証明装置記憶部101に書き込む。
1206
u=1,...,mに対して(Commit_{i_u},State_{i_u})=FuncCommit_{i_u}(x_{i_u},y_{i_u})を計算する。
1207
u=1,...,mに対してCommit_{i_u},State_{i_u}を証明装置記憶部101に書き込む。
1208
(Commit_1,...,Commit_n)を出力する。
終了

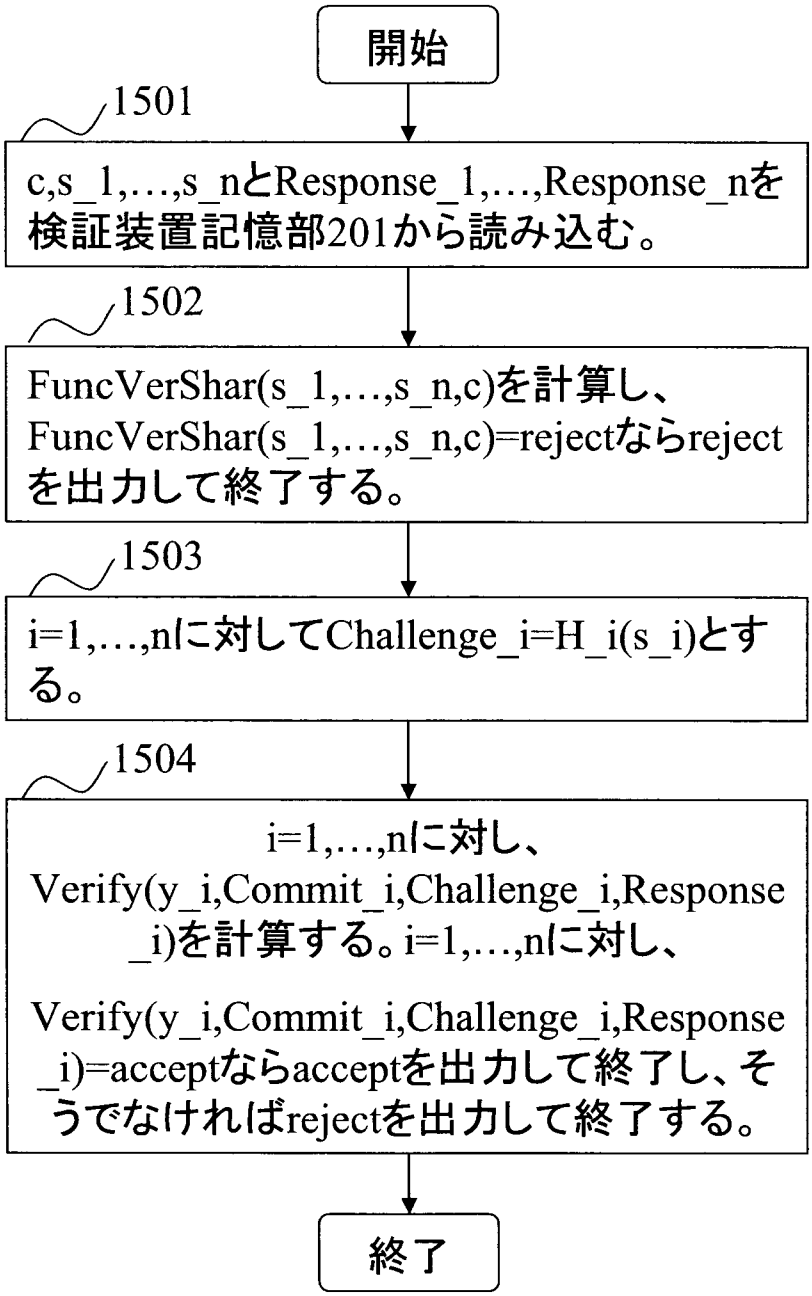
[図4]


開始
1301
SharSpの元cをランダムに選ぶ。
1302
cを検証装置記憶部201に書き込む。
1303
cを出力する。
終了

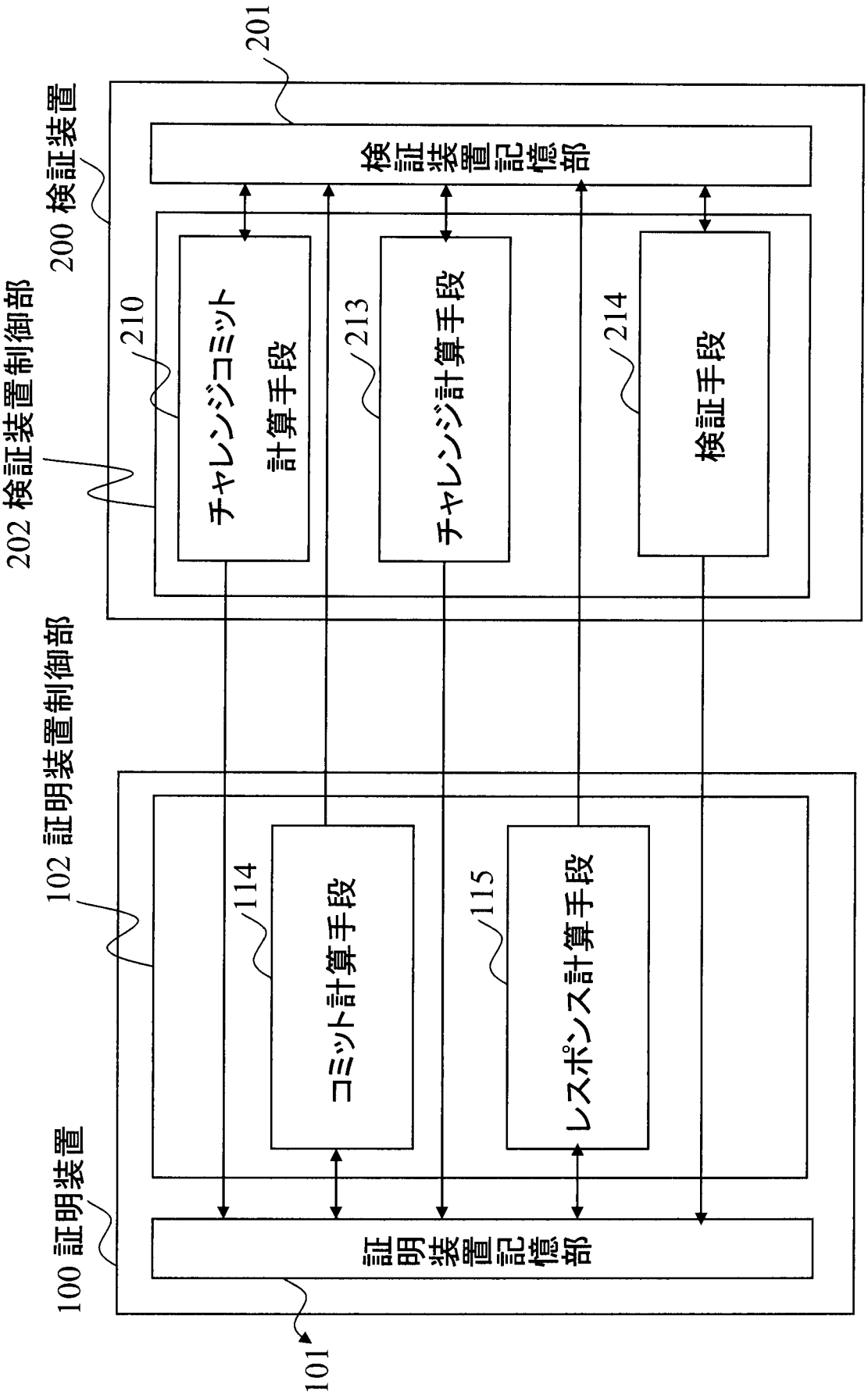
[図5]


開始
1401
Commit_1,...,Commit_nを証明装置記憶部101から読み込む。
1402
cを証明装置記憶部101から読み込む。
1403
v=1,...,pに対してState_{j_v}を証明装置記憶部101から読み込む。
1404
FuncReShar にcと(j_1,s_{j_1}),...,(j_p,s_{j_p})を入力して出力(i_1,s_{i_1}),...,(i_m,s_{i_m})を得る。
1405
v=1,...,pに対してChallenge_{i_v}=H_i(s_v)とする。
1406
u=1,...,mに対してResponse_{i_u}=FuncResponse_{i_u}(y_{i_u},Commit_{i_u},Challenge_{i_u},State_{i_u})とする。
1407
s_1,...,s_nとResponse_1,...,Response_nを出力する。
終了

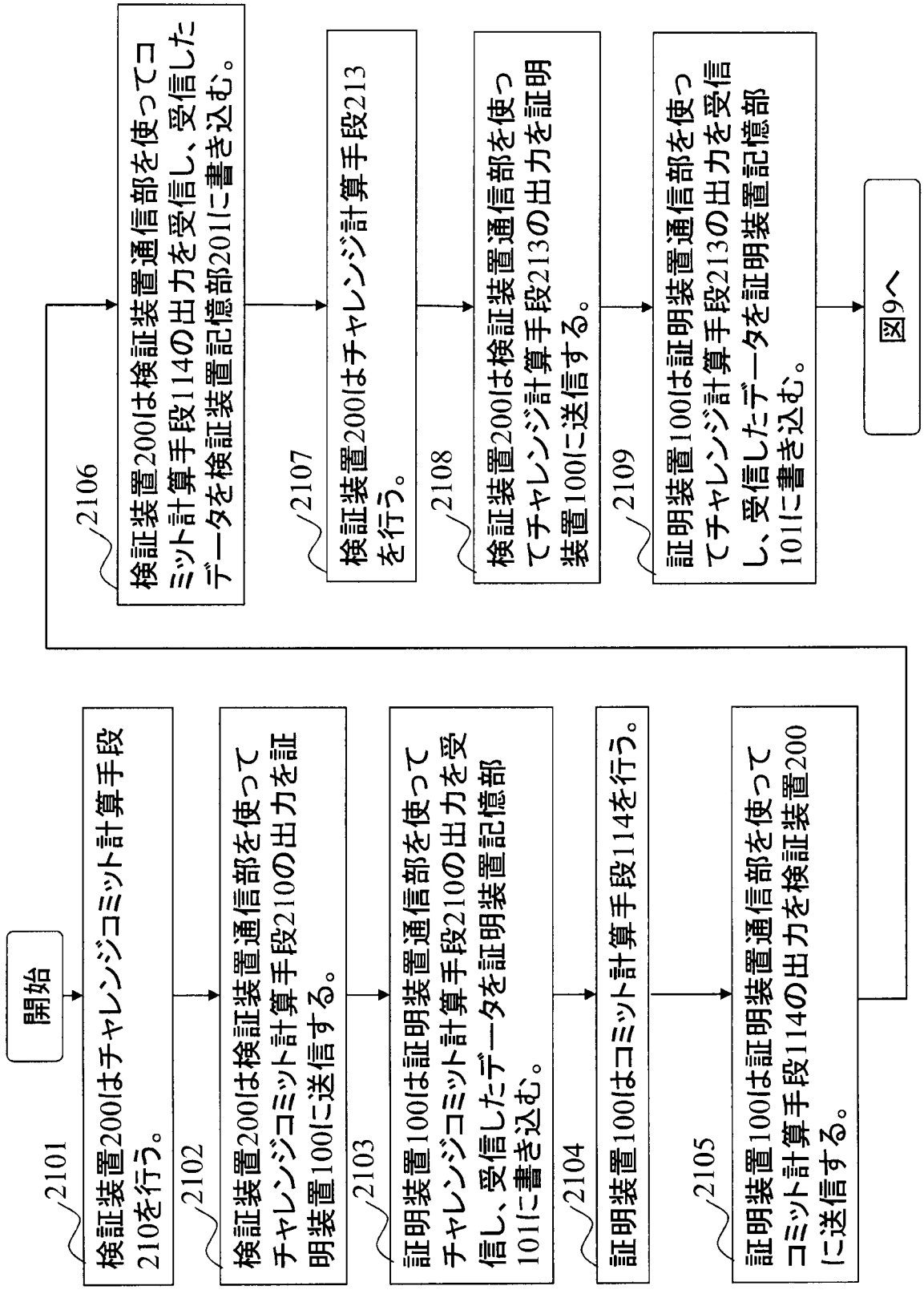
[図6]


開始
1501
c,s_1,…,s_nとResponse_1,…,Response_nを検証装置記憶部201から読み込む。
1502
FuncVerShar(s_1,…,s_n,c)を計算し、FuncVerShar(s_1,…,s_n,c)=rejectならrejectを出力して終了する。
1503
i=1,…,nに対してChallenge_i=H_i(s_i)とする。
1504
i=1,…,nに対し、Verify(y_i,Commit_i,Challenge_i,Response_i)を計算する。i=1,…,nに対し、Verify(y_i,Commit_i,Challenge_i,Response_i)=acceptならacceptを出力して終了し、そうでなければrejectを出力して終了する。
終了

[図7]


201
200 検証装置
検証装置記憶部
202 検証装置制御部
210
チャレンジコミット計算手段
213
チャレンジ計算手段
214
検証手段
102 証明装置制御部
114
コミット計算手段
115
レスポンス計算手段
100 証明装置
証明装置記憶部
101

[図8]


開始
2101
検証装置200はチャレンジコミット計算手段210を行う。
2102
検証装置200は検証装置通信部を使ってチャレンジコミット計算手段210の出力を証明装置100に送信する。
2103
証明装置100は証明装置通信部を使ってチャレンジコミット計算手段210の出力を受信し、受信したデータを証明装置記憶部101に書き込む。
2104
証明装置100はコミット計算手段114を行う。
2105
証明装置100は証明装置通信部を使ってコミット計算手段114の出力を検証装置200に送信する。
2106
検証装置200は検証装置通信部を使ってコミット計算手段114の出力を受信し、受信したデータを検証装置記憶部201に書き込む。
2107
検証装置200はチャレンジ計算手段213を行う。
2108
検証装置200は検証装置通信部を使ってチャレンジ計算手段213の出力を証明装置100に送信する。
2109
証明装置100は証明装置通信部を使ってチャレンジ計算手段213の出力を受信し、受信したデータを証明装置記憶部101に書き込む。
図9へ

[図9]

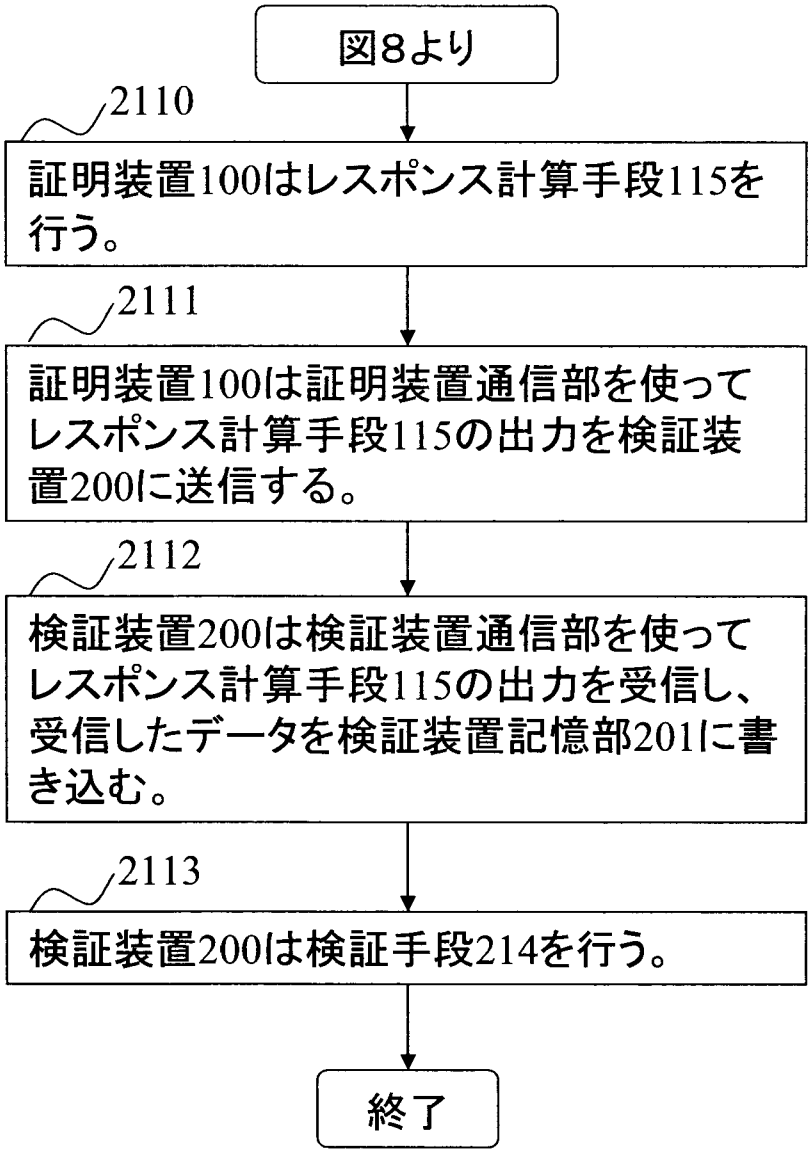
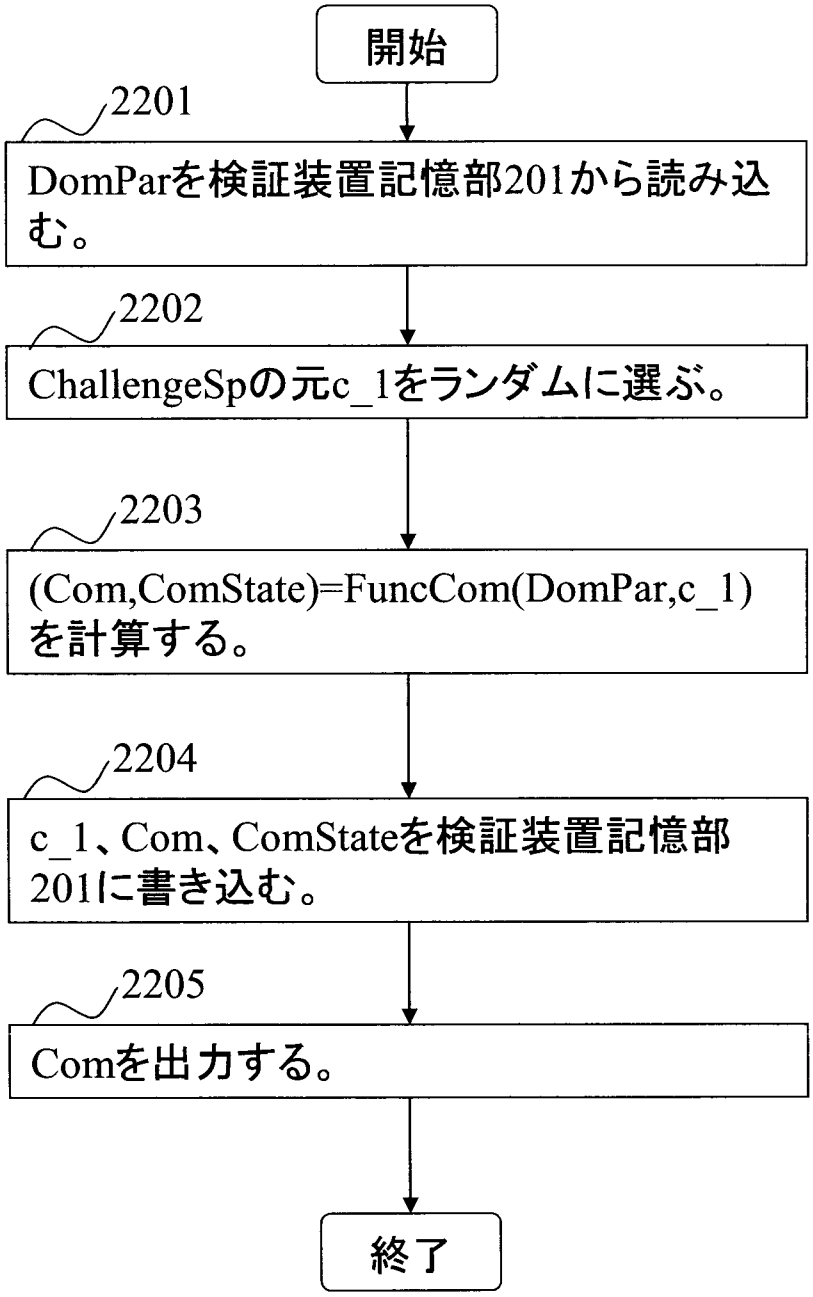
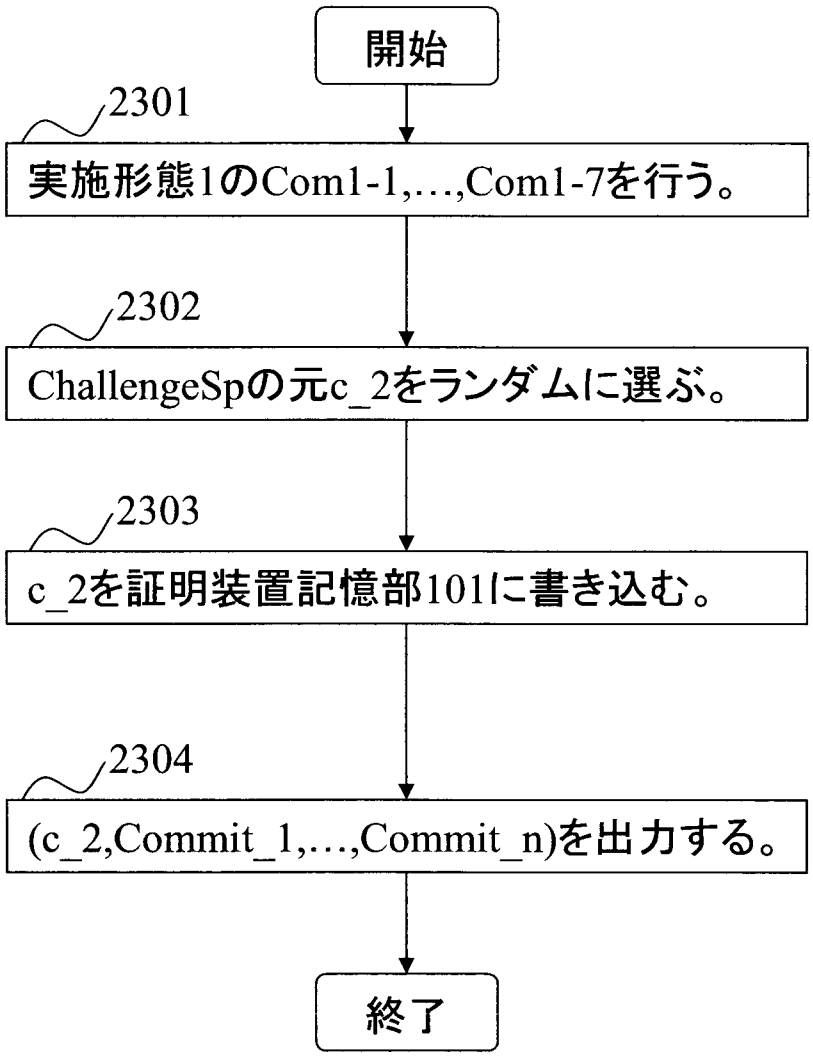


図8より
2110
証明装置100はレスポンス計算手段115を行う。
2111
証明装置100は証明装置通信部を使ってレスポンス計算手段115の出力を検証装置200に送信する。
2112
検証装置200は検証装置通信部を使ってレスポンス計算手段115の出力を受信し、受信したデータを検証装置記憶部201に書き込む。
2113
検証装置200は検証手段214を行う。
終了

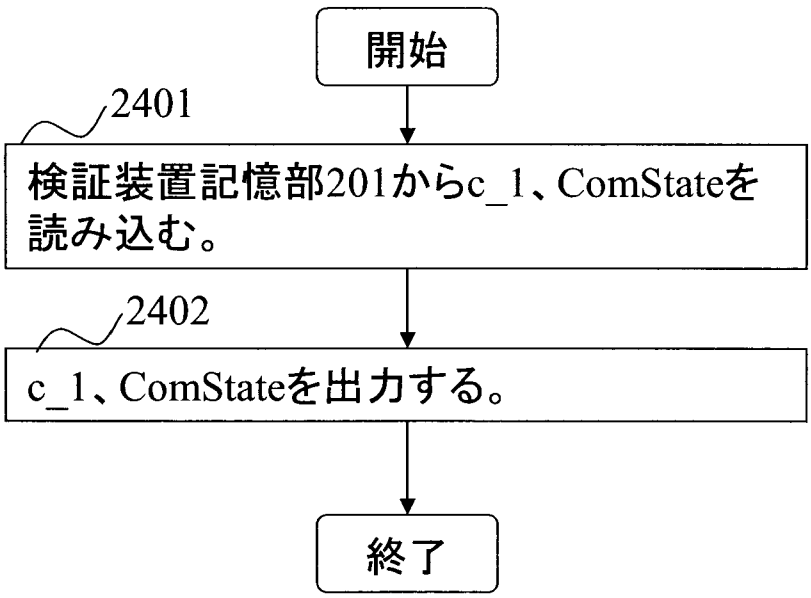
[図10]


開始
2201
DomParを検証装置記憶部201から読み込む。
2202
ChallengeSpの元c_1をランダムに選ぶ。
2203
(Com,ComState)=FuncCom(DomPar,c_1)を計算する。
2204
c_1、Com、ComStateを検証装置記憶部201に書き込む。
2205
Comを出力する。
終了

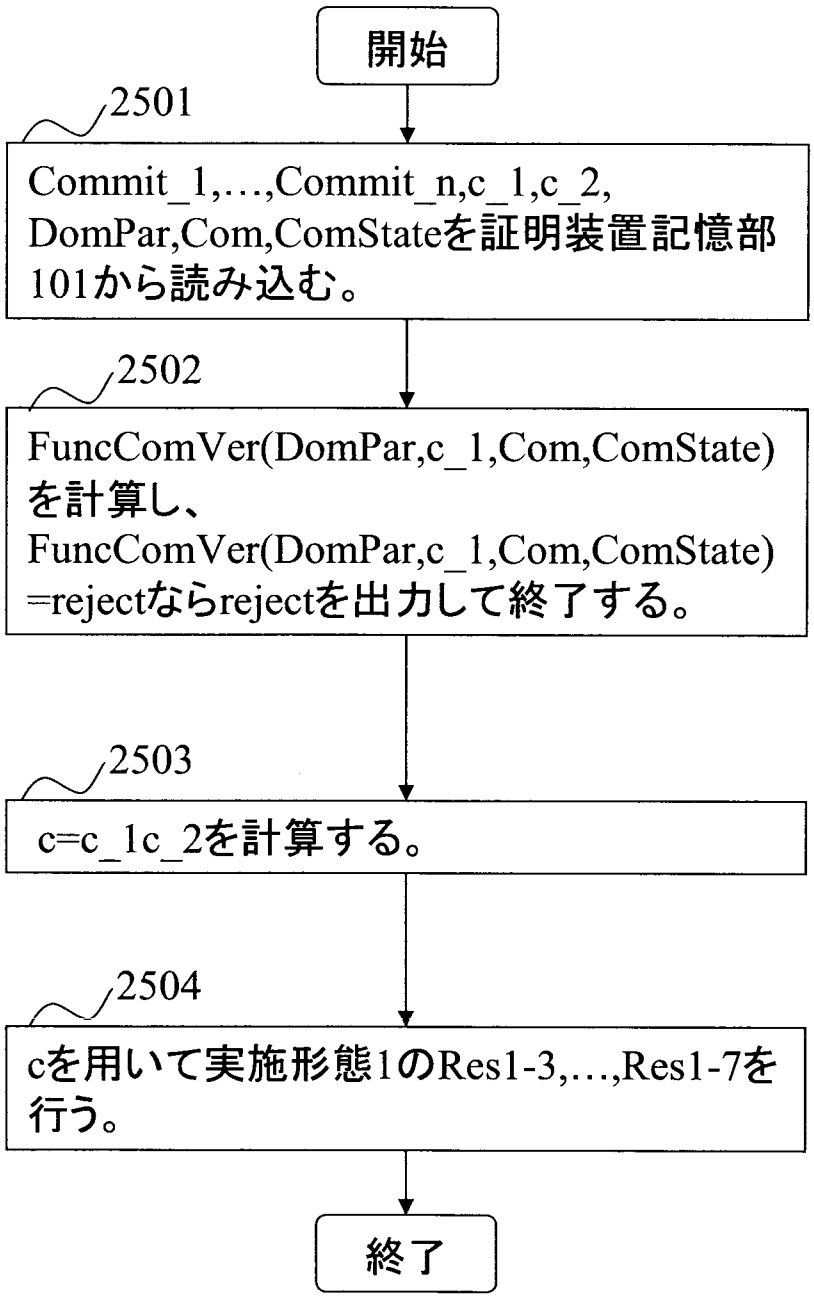
[図11]


開始
2301
実施形態1のCom1-1,...,Com1-7を行う。
2302
ChallengeSpの元c_2をランダムに選ぶ。
2303
c_2を証明装置記憶部101に書き込む。
2304
(c_2,Commit_1,...,Commit_n)を出力する。
終了

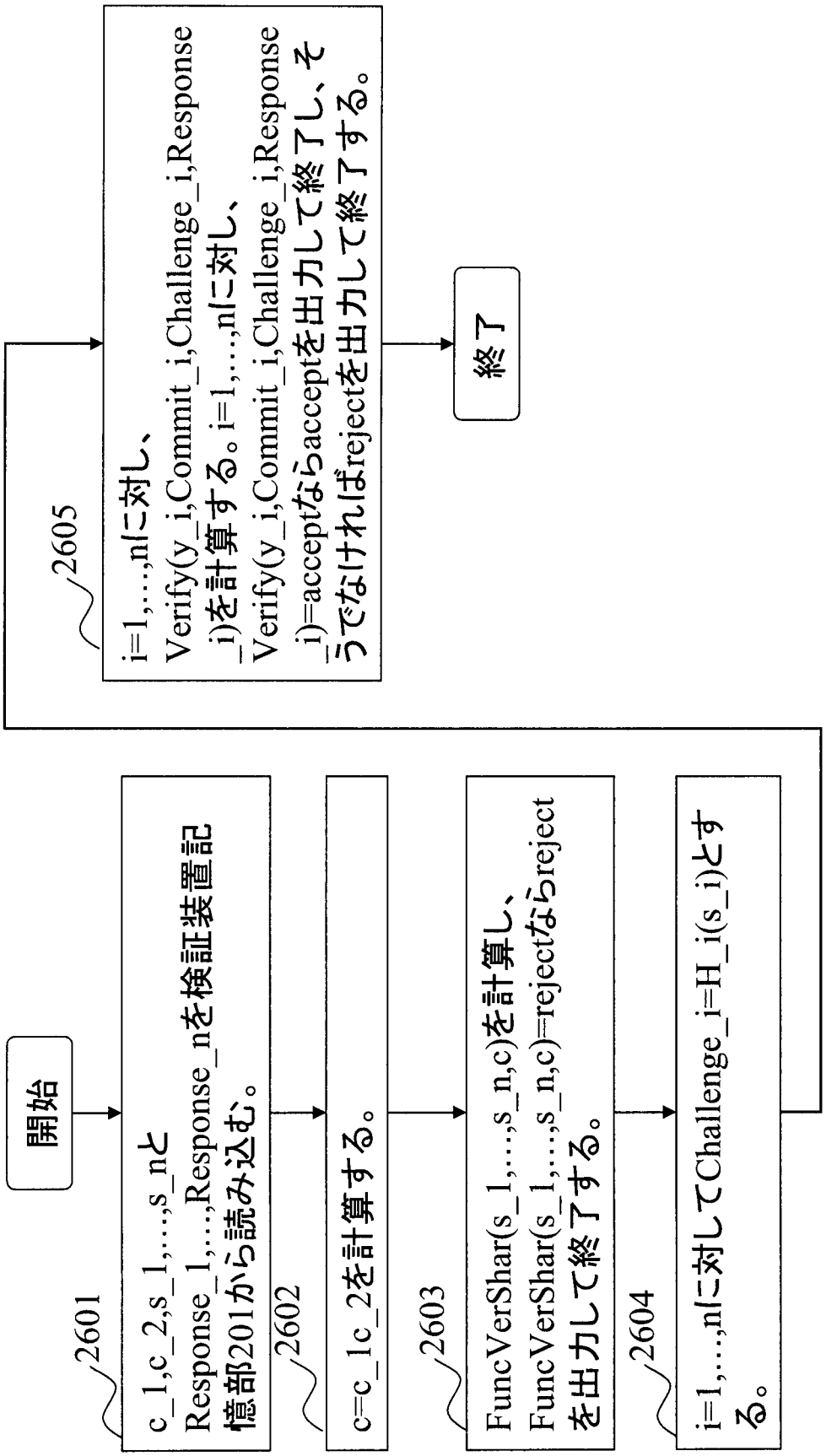
[図12]


開始
2401
検証装置記憶部201からc_1、ComStateを読み込む。
2402
c_1、ComStateを出力する。
終了

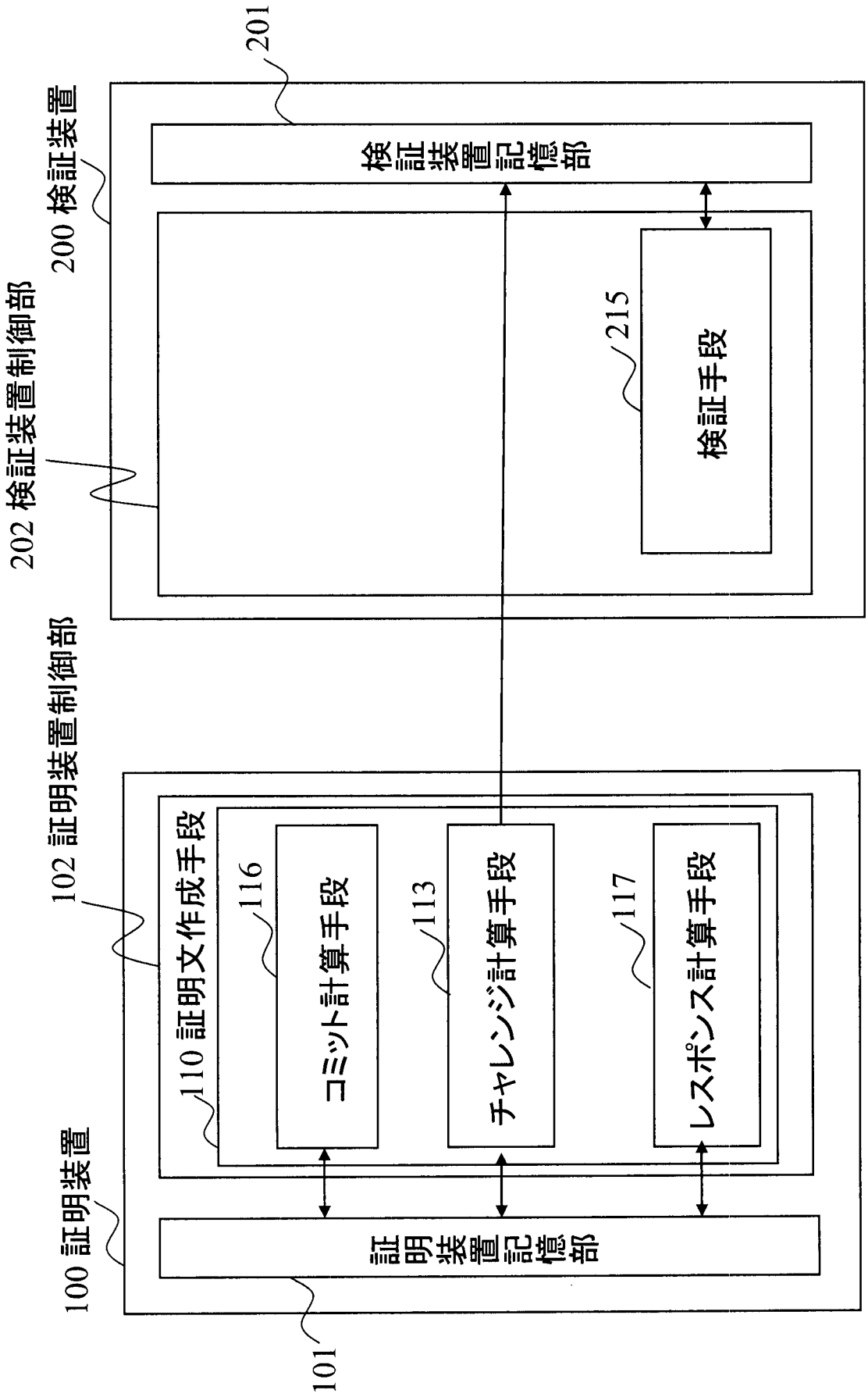
[図13]


開始
2501
Commit_1,…,Commit_n,c_1,c_2,
DomPar,Com,ComStateを証明装置記憶部
101から読み込む。
2502
FuncComVer(DomPar,c_1,Com,ComState)
を計算し、
FuncComVer(DomPar,c_1,Com,ComState)
=rejectならrejectを出力して終了する。
2503
c=c_1c_2を計算する。
2504
cを用いて実施形態1のRes1-3,…,Res1-7を
行う。
終了

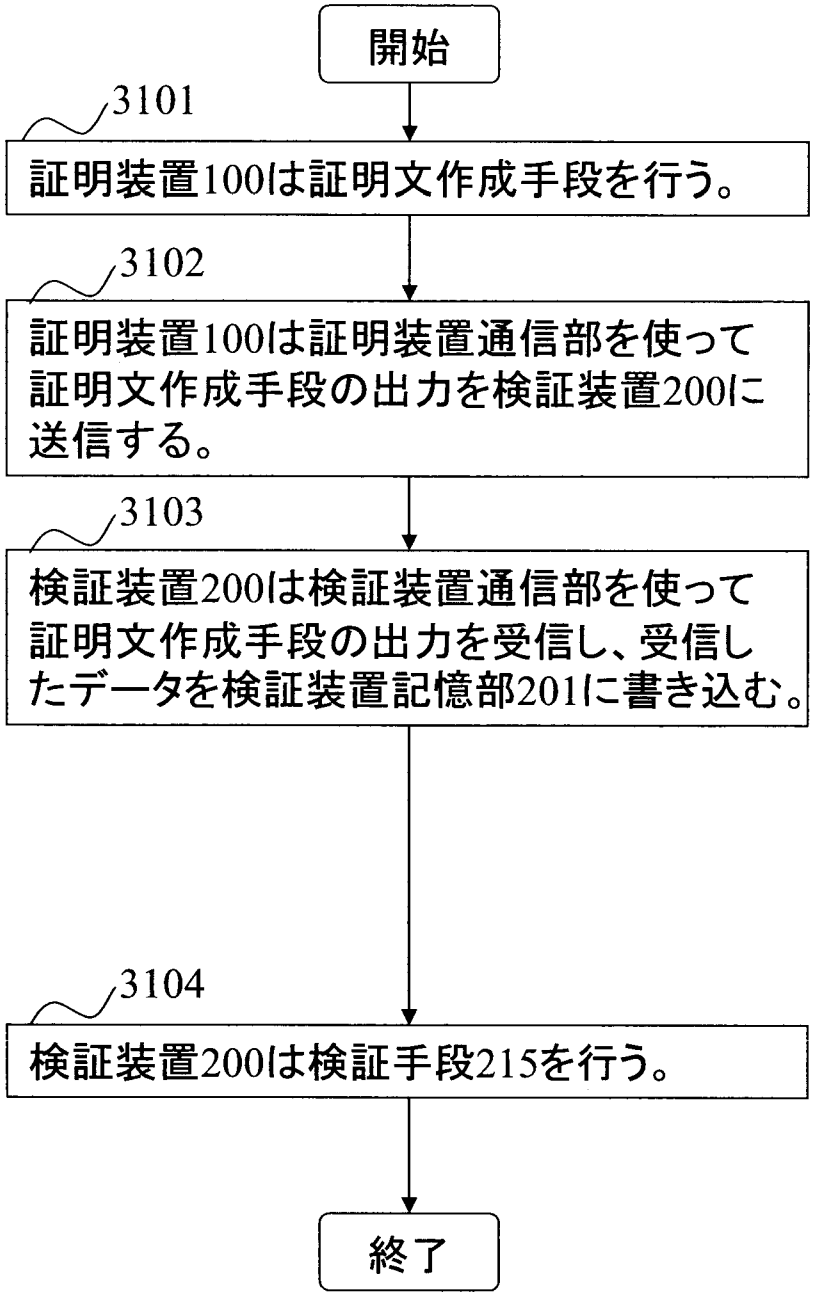
[図14]


開始
2601
c_1,c_2,s_1,...,s_nと
Response_1,...,Response_nを検証装置記憶部201から読み込む。
2602
c=c_1c_2を計算する。
2603
FuncVerShar(s_1,...,s_n,c)を計算し、
FuncVerShar(s_1,...,s_n,c)=rejectならrejectを出力して終了する。
2604
i=1,...,nに対してChallenge_i=H_i(s_i)とする。
2605
i=1,...,nに対し、
Verify(y_i,Commit_i,Challenge_i,Response_i)を計算する。i=1,...,nに対し、
Verify(y_i,Commit_i,Challenge_i,Response_i)=acceptならacceptを出力して終了し、そうでなければrejectを出力して終了する。
終了

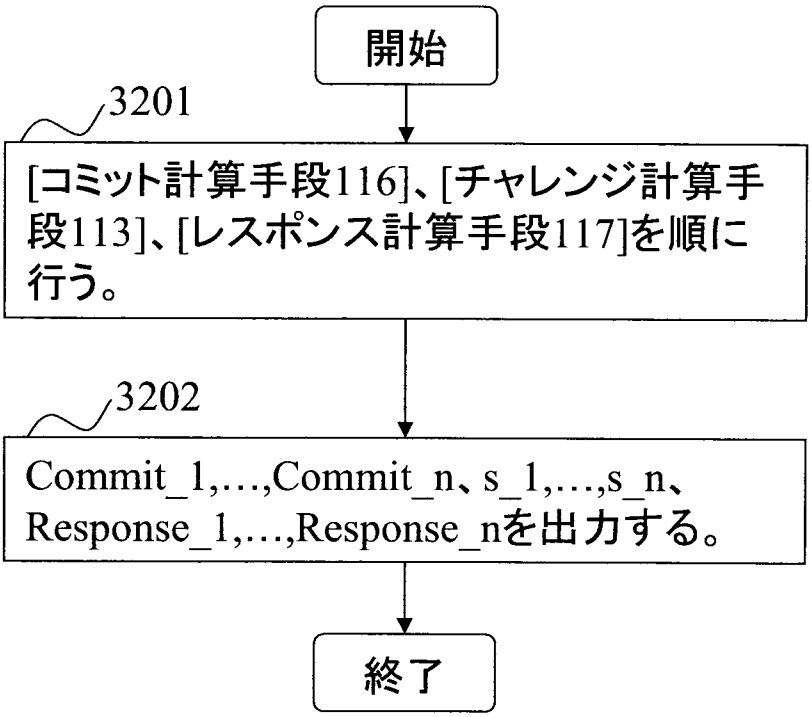
[図15]


201
200 検証装置
検証装置記憶部
202 検証装置制御部
215
検証手段
102 証明装置制御部
110 証明文作成手段
116
コミット計算手段
113
チャレンジ計算手段
117
レスポンス計算手段
100 証明装置
証明装置記憶部
101

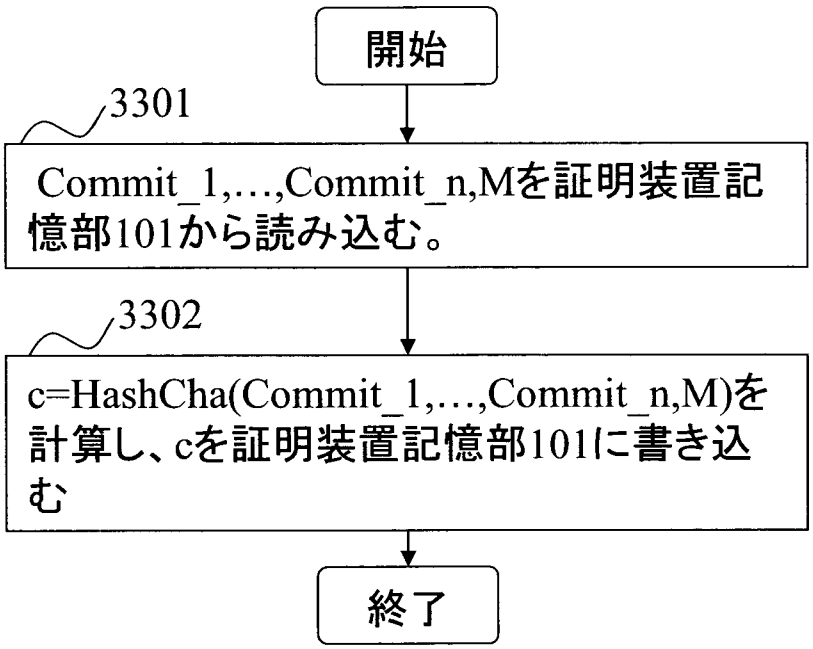
[図16]


開始
3101
証明装置100は証明文作成手段を行う。
3102
証明装置100は証明装置通信部を使って証明文作成手段の出力を検証装置200に送信する。
3103
検証装置200は検証装置通信部を使って証明文作成手段の出力を受信し、受信したデータを検証装置記憶部201に書き込む。
3104
検証装置200は検証手段215を行う。
終了

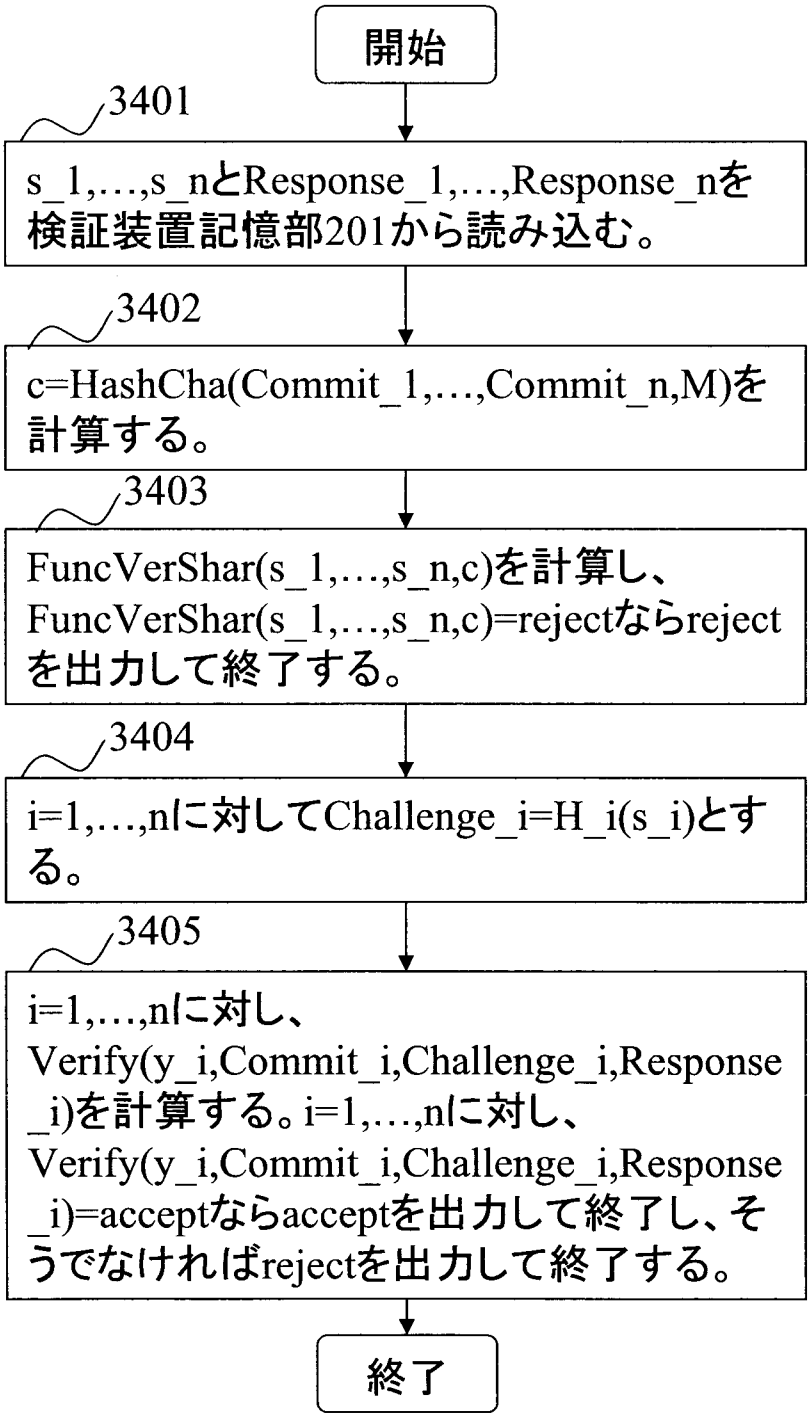
[図17]


開始
3201
[コミット計算手段116]、[チャレンジ計算手段113]、[レスポンス計算手段117]を順に行う。
3202
Commit_1,…,Commit_n、s_1,…,s_n、Response_1,…,Response_nを出力する。
終了

[図18]


開始
3301
Commit_1,…,Commit_n,Mを証明装置記憶部101から読み込む。
3302
c=HashCha(Commit_1,…,Commit_n,M)を計算し、cを証明装置記憶部101に書き込む
終了

[図19]


開始
3401
s_1,…,s_nとResponse_1,…,Response_nを検証装置記憶部201から読み込む。
3402
c=HashCha(Commit_1,…,Commit_n,M)を計算する。
3403
FuncVerShar(s_1,…,s_n,c)を計算し、FuncVerShar(s_1,…,s_n,c)=rejectならrejectを出力して終了する。
3404
i=1,…,nに対してChallenge_i=H_i(s_i)とする。
3405
i=1,…,nに対し、Verify(y_i,Commit_i,Challenge_i,Response_i)を計算する。i=1,…,nに対し、Verify(y_i,Commit_i,Challenge_i,Response_i)=acceptならacceptを出力して終了し、そうでなければrejectを出力して終了する。
終了

PATENT COOPERATION TREATY

PCT

DECLARATION OF NON-ESTABLISHMENT OF INTERNATIONAL SEARCH REPORT

(PCT Article 17(2)(a), Rules 13*ter*.1(c) and 39)

Applicant's or agent's file reference 06-0381-NEC	**IMPORTANT DECLARATION**	Date of mailing *(day/month/year)* 01 May, 2007 (01.05.07)
International application No. PCT/JP2007/ 051959	International filing date *(day/month/year)* 06 February, 2007 (06.02.07)	(Earliest) Priority Date *(day/month/year)* 09 February, 2006 (09.02.06)
International Patent Classification (IPC) or both national classification and IPC *G06F17/10* (2006.01)i		
Applicant NEC Corp.		

This International Searching Authority hereby declares, according to Article 17(2)(a), that **no international search report will be established** on the international application for the reasons indicated below.

1. ☒ The subject matter of the international application relates to:
   - a. ☐ scientific theories.
   - b. ☒ mathematical theories.
   - c. ☐ plant varieties.
   - d. ☐ animal varieties.
   - e. ☐ essentially biological processes for the production of plants and animals, other than microbiological processes and the products of such processes.
   - f. ☐ schemes, rules or methods of doing business.
   - g. ☐ schemes, rules or methods of performing purely mental acts.
   - h. ☐ schemes, rules or methods of playing games.
   - i. ☐ methods for treatment of the human body by surgery or therapy.
   - j. ☐ methods for treatment of the animal body by surgery or therapy.
   - k. ☐ diagnostic methods practised on the human or animal body.
   - l. ☐ mere presentations of information.
   - m. ☐ computer programs for which this International Searching Authority is not equipped to search prior art.
2. ☐ The failure of the following parts of the international application to comply with prescribed requirements prevents a meaningful search from being carried out:
   ☐ the description ☐ the claims ☐ the drawings
3. ☐ A meaningful search could not be carried out without the sequence listing; the applicant did not, within the prescribed time limit:
   - ☐ furnish a sequence listing on paper complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
   - ☐ furnish a sequence listing in electronic form complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
   - ☐ pay the required late furnishing fee for the furnishing of a sequence listing in response to an invitation under Rule 13ter.1(a) or (b).
4. ☐ A meaningful search could not be carried out without the tables related to the sequence listings; the applicant did not, within the prescribed time limit, furnish such tables in electronic form complying with the technical requirements provided for in Annex C-bis of the Administrative Instructions, and such tables were not available to the International Searching Authority in a form and manner acceptable to it.
5. Further comments:

Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

特 許 協 力 条 約

ＰＣＴ

国際調査報告を作成しない旨の決定

（法第８条第２項、法施行規則第42条、第50条の３第７項）
〔ＰＣＴ17条(2)(a)、ＰＣＴ規則13の3.1(c) 及び(d)、39〕

出願人又は代理人 の書類記号 06-0381-NEC	重要決定	発送日 （日.月.年）　０１．０５．２００７
国際出願番号 ＰＣＴ／ＪＰ２００７／０５１９５９	国際出願日 （日.月.年）　０６．０２．２００７	優先日 （日.月.年）　０９．０２．２００６
国際特許分類（ＩＰＣ）Int.Cl. G06F17/10(2006.01)i		
出願人（氏名又は名称） 日本電気株式会社		

この出願については、法第８条第２項（ＰＣＴ17条(2)(a)）の規定に基づき、次の理由により国際調査報告を作成しない旨の決定をする。

１．☑ この国際出願は、次の事項を内容としている。
- ａ．☐ 科学の理論
- ｂ．☑ 数学の理論
- ｃ．☐ 植物の品種
- ｄ．☐ 動物の品種
- ｅ．☐ 植物及び動物の生産の本質的に生物学的な方法（微生物学的方法による生産物及び微生物学的方法を除く。）
- ｆ．☐ 事業活動に関する計画、法則又は方法
- ｇ．☐ 純粋に精神的な行為の遂行に関する計画、法則又は方法
- ｈ．☐ 遊戯に関する計画、法則又は方法
- ｉ．☐ 人の身体の手術又は治療による処置方法
- ｊ．☐ 動物の身体の手術又は治療による処置方法
- ｋ．☐ 人又は動物の身体の診断方法
- ｌ．☐ 情報の単なる提示
- ｍ．☐ この国際調査機関が先行技術を調査できないコンピューター・プログラム

２．☐ この国際出願の次の部分が所定の要件を満たしていないので、有効な国際調査をすることができない。
☐ 明細書　☐ 請求の範囲　☐ 図面

３．☐ 入手可能な配列表が存在せず、有意義な調査を行うことができなかった。
出願人は所定の期間内に、
- ☐ 実施細則の附属書Ｃに定める基準を満たす紙形式の配列表を提出しなかったため、国際調査機関は、認められた形式及び方法で配列表を入手することができなかった。
- ☐ 実施細則の附属書Ｃに定める基準を満たす電子形式の配列表を提出しなかったため、国際調査機関は、認められた形式及び方法で配列表を入手することができなかった。
- ☐ ＰＣＴ規則13の3.1(a)又は(b)に基づく命令に応じた、要求された配列表の遅延提出手数料を支払わなかった。

４．☐ 入手可能な配列表に関連するテーブルが存在しないため、有意義な調査ができなかった。すなわち、出願人が、所定の期間内に、実施細則の附属書Ｃの２に定める技術的な要件を満たす電子形式のテーブルを提出しなかったため、国際調査機関は、認められた形式及び方法でテーブルを入手することができなかった。

５．附記

名称及びあて名 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	特許庁審査官（権限のある職員） 中里　裕正 電話番号　０３－３５８１－１１０１　内線　３５４６	５Ｓ　９３６４

様式ＰＣＴ／ＩＳＡ／２０３（２００５年４月）