

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5241926号

(P5241926)

(45) 発行日 平成25年7月17日 (2013. 7. 17)

(24) 登録日 平成25年4月12日 (2013. 4. 12)

(51) Int. Cl.		F I			
H04L	9/32	(2006.01)	H04L	9/00	675A
G09C	1/00	(2006.01)	G09C	1/00	640E
G06F	13/00	(2006.01)	G06F	13/00	540A

請求項の数 64 (全 43 頁)

(21) 出願番号	特願2011-542559 (P2011-542559)	(73) 特許権者	507364838
(86) (22) 出願日	平成21年12月22日 (2009. 12. 22)		クアルコム, インコーポレイテッド
(65) 公表番号	特表2012-512617 (P2012-512617A)		アメリカ合衆国 カリフォルニア 921
(43) 公表日	平成24年5月31日 (2012. 5. 31)		21 サン ディエゴ モアハウス ドラ
(86) 国際出願番号	PCT/US2009/069113		イブ 5775
(87) 国際公開番号	W02010/075338	(74) 代理人	100108453
(87) 国際公開日	平成22年7月1日 (2010. 7. 1)		弁理士 村山 靖彦
審査請求日	平成23年6月16日 (2011. 6. 16)	(74) 代理人	100163522
(31) 優先権主張番号	12/342, 021		弁理士 黒田 晋平
(32) 優先日	平成20年12月22日 (2008. 12. 22)	(72) 発明者	ル・シャオ
(33) 優先権主張国	米国 (US)		アメリカ合衆国・カリフォルニア・921
			21・サン・ディエゴ・モアハウス・ドラ
			イブ・5775

最終頁に続く

(54) 【発明の名称】 ピアツーピア・ネットワークのための分散ハッシュテーブルにおけるセキュリティで保護されたノード識別子割当て

(57) 【特許請求の範囲】

【請求項 1】

ピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するために参加するノード上で動作する方法であって、

第1の寄与を生成するステップと、

参加要求を紹介ノードに送信するステップと、

前記紹介ノードから複数の寄与を受信するステップと、

前記第1の寄与および前記受信された寄与の関数として前記ノード識別子を生成するステップとを備え、

前記参加要求は、前記オーバーレイ・ネットワークの識別子空間内で前記ノード識別子を生成するための1または複数の寄与を取得することを求め、

前記寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、前記第1の寄与にも結び付けられる

ことを特徴とする方法。

【請求項 2】

前記第1の寄与に基づいて第1のハッシュ値を生成するステップをさらに備え、

前記参加要求は、検証署名を計算する際に複数の寄与ノードによって使用される前記第1のハッシュ値を含むことを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記紹介ノードから複数の検証署名を受信するステップをさらに備え、

10

20

各検証署名は、複数の寄与ノードによって生成された複数のコミットメント値の関数であり、前記コミットメント値は、前記寄与ノードによって供給される前記寄与の関数であることを特徴とする請求項 1 に記載の方法。

【請求項 4】

各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、前記コミットメント値と一緒に結び付け、その結果、前記寄与と一緒に結び付けることを特徴とする請求項 3 に記載の方法。

【請求項 5】

前記寄与は、ランダム値または擬似ランダム値のいずれかであることを特徴とする請求項 1 に記載の方法。

10

【請求項 6】

前記参加要求は、前記第1の寄与の第1のハッシュ値を含み、

前記第1のハッシュ値は、複数の寄与ノードに供給されて、前記検証署名を計算する際に使用されることを特徴とする請求項 1 に記載の方法。

【請求項 7】

前記ノード識別子を前記オーバーレイ・ネットワークにおける1または複数の隣接ノードに配信するステップをさらに備え、

前記ノード識別子の有効性は、前記1または複数の隣接ノードによって独立に検証され得ることを特徴とする請求項 1 に記載の方法。

【請求項 8】

20

第1の寄与を生成するための寄与ジェネレータと、

参加要求を紹介ノードに送信するための参加要求モジュールと、

前記紹介ノードから複数の寄与を受信するための寄与収集モジュールと、

前記第1の寄与および前記受信された寄与の関数として前記ノード識別子を生成するためのノード識別子ジェネレータとを備え、

前記参加要求は、オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するための1または複数の寄与を取得することを求め、

前記寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、前記第1の寄与にも結び付けられる

ことを特徴とするネットワークデバイス。

30

【請求項 9】

前記第1の寄与に基づいて第1のハッシュ値を生成するためのハッシュジェネレータをさらに備え、

前記参加要求は、検証署名を計算する際に複数の寄与ノードによって使用される前記第1のハッシュ値を含むことを特徴とする請求項 8 に記載のデバイス。

【請求項 10】

前記寄与収集モジュールは、前記紹介ノードから複数の検証署名を受信するようにさらに構成され、各検証署名は、複数の寄与ノードによって生成される複数のコミットメント値の関数であり、前記コミットメント値は、前記寄与ノードによって供給される前記寄与の関数であることを特徴とする請求項 8 に記載のデバイス。

40

【請求項 11】

各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、前記コミットメント値と一緒に結び付け、その結果、前記寄与と一緒に結び付けることを特徴とする請求項 10 に記載のデバイス。

【請求項 12】

前記寄与は、ランダム値または擬似ランダム値のいずれかであることを特徴とする請求項 8 に記載のデバイス。

【請求項 13】

前記参加要求は、前記第1の寄与の第1のハッシュ値を含み、前記第1のハッシュ値は、複数の寄与ノードに供給されて、前記検証署名を計算する際に使用されることを特徴とす

50

る請求項 8 に記載のデバイス。

【請求項 1 4】

前記ノード識別子を前記オーバーレイ・ネットワークにおける1または複数の隣接ノードに配信するためのノード識別子配信モジュールをさらに備え、

前記ノード識別子の有効性は、前記1または複数の隣接ノードによって独立に検証され得ることを特徴とする請求項 8 に記載のデバイス。

【請求項 1 5】

第1の寄与を生成するための手段と、

参加要求を紹介ノードに送信するための手段と、

前記紹介ノードから複数の寄与を受信するための手段と、

前記第1の寄与および前記受信された寄与の関数として前記ノード識別子を生成するための手段とを備え、

前記参加要求は、オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するための1または複数の寄与を取得することを求め、

前記寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、前記第1の寄与にも結び付けられる

ことを特徴とするネットワークデバイス。

【請求項 1 6】

前記紹介ノードから複数の検証署名を受信するための手段をさらに備え、

各検証署名は、複数の寄与ノードによって生成された複数のコミットメント値の関数であり、前記コミットメント値は、前記寄与ノードによって供給される前記寄与の関数であることを特徴とする請求項 1 5 に記載のデバイス。

【請求項 1 7】

前記ノード識別子を前記オーバーレイ・ネットワークにおける1または複数の隣接ノードに配信するための手段をさらに備え、

前記ノード識別子の有効性は、前記1または複数の隣接ノードによって独立に検証され得ることを特徴とする請求項 1 5 に記載のデバイス。

【請求項 1 8】

第1の寄与を生成し、

参加要求を紹介ノードに送信し、

前記紹介ノードから、複数の寄与を受信し、さらに

前記第1の寄与および前記受信された寄与の関数として前記ノード識別子を生成するように適合された処理回路を含み、

前記参加要求は、オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するための1または複数の寄与を取得することを求め、

前記寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、前記第1の寄与にも結び付けられる

ことを特徴とするプロセッサ。

【請求項 1 9】

前記処理回路は、前記紹介ノードから複数の検証署名を受信するようにさらに適合され、各検証署名は、複数の寄与ノードによって生成された複数のコミットメント値の関数であり、前記コミットメント値は、前記寄与ノードによって供給される前記寄与の関数であることを特徴とする請求項 1 8 に記載のプロセッサ。

【請求項 2 0】

前記処理回路は、前記ノード識別子を前記オーバーレイ・ネットワークにおける1または複数の隣接ノードに配信するようにさらに適合され、前記ノード識別子の有効性は、前記1または複数の隣接ノードによって独立に検証され得ることを特徴とする請求項 1 8 に記載のプロセッサ。

【請求項 2 1】

1または複数のプロセッサによって実行されると、前記プロセッサに、

第1の寄与を生成させ、

オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するための1または複数の寄与を取得することを求める参加要求を紹介ノードに送信させ、

前記紹介ノードから、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、前記第1の寄与にも結び付けられる複数の寄与を受信させ、さらに

前記第1の寄与および前記受信された寄与の関数として前記ノード識別子を生成させる前記オーバーレイ・ネットワーク内で前記ノード識別子を生成するためにノードにおいて動作する命令を備えることを特徴とするマシン可読媒体。

【請求項 2 2】

10

前記1または複数のプロセッサによって実行されると、前記プロセッサに、前記紹介ノードから複数の検証署名を受信させる1または複数の命令をさらに備え、各検証署名は、複数の寄与ノードによって生成された複数のコミットメント値の関数であり、前記コミットメント値は、前記寄与ノードによって供給される前記寄与の関数であることを特徴とする請求項 2 1 に記載のマシン可読媒体。

【請求項 2 3】

1または複数のプロセッサによって実行されると、前記プロセッサに、前記ノード識別子を前記オーバーレイ・ネットワークにおける1または複数の隣接ノードに配信させる1または複数の命令をさらに備え、前記ノード識別子の有効性は、前記1または複数の隣接ノードによって独立に検証され得る請求項 2 1 に記載のマシン可読媒体。

20

【請求項 2 4】

参加するノードがピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するのを支援するために紹介ノード上で動作する方法であって、

前記オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するための1または複数の寄与を取得することを求める参加要求を、前記参加するノードから受信するステップと、

前記オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信するステップと、

前記寄与ノードから複数のコミットメント値を受信するステップと、

前記複数の寄与値を前記寄与ノードに配信するステップと、

30

前記複数のコミットメント値を前記配信するステップに応答して、前記複数の寄与、および対応する検証署名を受信するステップと、

前記複数の寄与を前記参加するノードに送信するステップとを備え、

各コミットメント値は、前記寄与ノードのそれぞれにおいて生成された対応する寄与の関数であり、

各検証署名は、前記複数のコミットメント値の関数であり、

前記ノード識別子は、前記寄与の関数として前記参加するノードによって生成されることを特徴とする方法。

【請求項 2 5】

前記寄与は、ランダム値または擬似ランダム値のいずれかである請求項 2 4 に記載の方法。

40

【請求項 2 6】

前記参加要求は、前記第1の寄与の第1のハッシュ値を含み、前記第1のハッシュ値は、前記寄与ノードに供給されて、前記検証署名を計算する際に使用されることを特徴とする請求項 2 4 に記載の方法。

【請求項 2 7】

前記複数のコミットメント値は、前記寄与ノードによって他の寄与ノードに直接に配信されることを特徴とする請求項 2 4 に記載の方法。

【請求項 2 8】

寄与-検証署名ペアが、前記複数の寄与ノードのそれぞれから受信されることを特徴と

50

する請求項 2 4 に記載の方法。

【請求項 2 9】

各検証署名は、前記参加するノードからの第1の値の関数でもあることを特徴とする請求項 2 4 に記載の方法。

【請求項 3 0】

各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、前記コミットメント値と一緒に結び付け、その結果、前記寄与と一緒に結び付けることを特徴とする請求項 2 9 に記載の方法。

【請求項 3 1】

前記寄与を前記参加するノードに前記送信するステップに先立って、前記コミットメント値および前記検証署名を検証するステップをさらに備えることを特徴とする請求項 2 4 に記載の方法。

10

【請求項 3 2】

前記検証署名を前記参加するノードに送信するステップをさらに備えることを特徴とする請求項 2 4 に記載の方法。

【請求項 3 3】

前記寄与要求が送信される寄与ノードの数は、前記オーバーレイ・ネットワークにおける敵対的ノードの数より少なくとも1つ多いことを特徴とする請求項 2 4 に記載の方法。

【請求項 3 4】

前記寄与要求が送信される前記寄与ノードは、ランダムに選択されるか、または擬似ランダムに選択されるかのいずれかであることを特徴とする請求項 2 4 に記載の方法。

20

【請求項 3 5】

前記複数の寄与要求が送信される寄与ノードの数を減らすステップをさらに備え、前記寄与ノードは、前記ピアツーピア・オーバーレイ・ネットワークにおけるノードのより大きいセットからのノードの縮小されたセットであることを特徴とする請求項 2 4 に記載の方法。

【請求項 3 6】

ノードの前記縮小されたセットは、ノード間の結託の可能性を回避するようにランダムに選択されることを特徴とする請求項 3 5 に記載の方法。

【請求項 3 7】

寄与ノードの前記ランダムな選択は、前記参加するノードに関する公開鍵に基づくことを特徴とする請求項 3 5 に記載の方法。

30

【請求項 3 8】

寄与ノードの前記ランダムな選択は、前記紹介ノードに関する公開鍵に基づくことを特徴とする請求項 3 5 に記載の方法。

【請求項 3 9】

値xが、前記紹介ノードに関する前記公開鍵に基づいて生成され、この値xは、前記オーバーレイ・ネットワークにおける値xに関するそのポイントの責任を負うノード n_C を見出すのに使用され、次に、前記ノード n_C が、それぞれ、前記オーバーレイ・ネットワークにおける前記ノード n_C の先行ノードおよび後続ノードである、前のノード n_{C-Prev} および次のノード n_{C-Next} を特定するのに使用され、前記前のノードおよび前記次のノードは、寄与ノードとして利用されることを特徴とする請求項 3 5 に記載の方法。

40

【請求項 4 0】

値xが、前記紹介ノードに関する前記公開鍵、および第1の文字列に基づいて生成され、値yが、前記紹介ノードに関する前記公開鍵、および第2の文字列に基づいて生成され、値zが、前記紹介ノードに関する前記公開鍵、および第3の文字列に基づいて生成され、次に、前記値x、前記値y、および前記値zは、前記オーバーレイ・ネットワークにおける対応する値x、y、およびzの責任を負う複数のノード n_X 、 n_Y 、および n_Z を見出すのに使用され、これらの複数のノード n_X 、 n_Y 、および n_Z は、前記寄与ノードとして使用される請求項 3 5 に記載の方法。

50

【請求項 4 1】

オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するための1または複数の寄与を取得することを求める、参加するノードからの参加要求を受信するための要求レシーバと、

前記オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信するための寄与要求モジュールと、

前記寄与ノードから、それぞれが前記寄与ノードのそれぞれにおいて生成される対応する寄与の関数である複数のコミットメント値を受信するためのコミットメントレシーバと、

前記複数のコミットメント値を前記寄与ノードに配信するためのコミットメントディストリビュータと、

前記複数のコミットメント値が前記配信されることに応答して、前記複数の寄与、およびそれぞれが前記複数のコミットメント値の関数である対応する検証署名を受信するための寄与収集モジュールと、

前記複数の寄与を前記参加するノードに送信するための寄与配信モジュールとを備え、ノード識別子が、前記寄与の関数として、前記参加するノードによって生成されることを特徴とするネットワークデバイス。

【請求項 4 2】

前記寄与は、ランダム値または擬似ランダム値のいずれかである請求項 4 1 に記載のデバイス。

【請求項 4 3】

前記参加要求は、前記参加するノードからの第1の寄与の第1のハッシュ値を含み、前記第1のハッシュ値は、前記寄与ノードに供給されて、前記検証署名を計算する際に使用されることを特徴とする請求項 4 1 に記載のデバイス。

【請求項 4 4】

前記複数のコミットメント値は、前記寄与ノードによって他の寄与ノードに直接に配信されることを特徴とする請求項 4 1 に記載のデバイス。

【請求項 4 5】

各検証署名は、前記参加するノードからの第1の値の関数でもあることを特徴とする請求項 4 1 に記載のデバイス。

【請求項 4 6】

各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、前記コミットメント値と一緒に結び付け、その結果、前記寄与と一緒に結び付けることを特徴とする請求項 4 5 に記載のデバイス。

【請求項 4 7】

前記寄与を前記参加するノードに前記送信するのに先立って、前記コミットメント値および前記検証署名を検証するための検証モジュールをさらに備えることを特徴とする請求項 4 1 に記載のデバイス。

【請求項 4 8】

前記寄与ディストリビュータは、前記検証署名を前記参加するノードにさらに送信することを特徴とする請求項 4 1 に記載のデバイス。

【請求項 4 9】

前記寄与要求が送信される寄与ノードの数は、前記オーバーレイ・ネットワークにおける敵対的ノードの数より少なくとも1つ多いことを特徴とする請求項 4 1 に記載のデバイス。

【請求項 5 0】

前記寄与要求が送信される前記寄与ノードは、ランダムに選択されるか、または擬似ランダムに選択されるかのいずれかであることを特徴とする請求項 4 1 に記載のデバイス。

【請求項 5 1】

オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するた

10

20

30

40

50

めの1または複数の寄与を取得することを求める、参加するノードからの参加要求を受信するための手段と、

前記オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信するための手段と、

前記寄与ノードから、それぞれが前記寄与ノードのそれぞれにおいて生成される対応する寄与の関数である複数のコミットメント値を受信するための手段と、

前記複数のコミットメント値を前記寄与ノードに配信するための手段と、

前記複数のコミットメント値が前記配信されることに応答して、前記複数の寄与、およびそれぞれが前記複数のコミットメント値の関数である対応する検証署名を受信するための手段と、

前記複数の寄与を前記参加するノードに送信するための手段とを備え、

ノード識別子が、前記寄与の関数として、前記参加するノードによって生成されることを特徴とするネットワークデバイス。

【請求項52】

各検証署名は、各寄与ノードがそのノードの寄与を実際に表示するのに先立って、前記コミットメント値と一緒に結び付け、その結果、前記寄与と一緒に結び付けることを特徴とする請求項51に記載のデバイス。

【請求項53】

前記寄与を前記参加するノードに前記送信するのに先立って、前記コミットメント値および前記検証署名を検証するための手段をさらに備えることを特徴とする請求項51に記載のデバイス。

【請求項54】

オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するための1または複数の寄与を取得することを求める、参加するノードからの参加要求を受信し、

前記オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信し、

前記寄与ノードから、それぞれが前記寄与ノードのそれぞれにおいて生成される対応する寄与の関数である複数のコミットメント値を受信し、

前記複数のコミットメント値を前記寄与ノードに配信し、

前記複数のコミットメント値が前記配信されることに応答して、前記複数の寄与、およびそれぞれが前記複数のコミットメント値の関数である対応する検証署名を受信し、さらに

前記複数の寄与を前記参加するノードに送信するように適合された処理回路を含み、

ノード識別子が、前記寄与の関数として、前記参加するノードによって生成されることを特徴とするプロセッサ。

【請求項55】

1または複数のプロセッサによって実行されると、前記プロセッサに、

オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するための1または複数の寄与を取得することを求める、参加するノードからの参加要求を受信させ、

前記オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信させ、

前記寄与ノードから、それぞれが前記寄与ノードのそれぞれにおいて生成される対応する寄与の関数である複数のコミットメント値を受信させ、

前記複数のコミットメント値を前記寄与ノードに配信させ、

前記複数のコミットメント値が前記配信されることに応答して、前記複数の寄与、およびそれぞれが前記複数のコミットメント値の関数である対応する検証署名を受信させ、さらに

前記複数の寄与を前記参加するノードに送信させ、前記参加するノードがピアツーピア

10

20

30

40

50

・オーバーレイ・ネットワークに関する前記ノード識別子を生成するのを支援するためにノードにおいて動作する命令を備え、
ノード識別子が、前記寄与の関数として、前記参加するノードによって生成されることを特徴とするマシン可読媒体。

【請求項 56】

参加するノードがピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するのを支援するために寄与ノード上で動作する方法であって、

前記参加するノードからのハッシュされた初期寄与を含む、寄与を求める寄与要求を受信するステップと、

第1の寄与を取得するステップと、

前記第1の寄与からコミットメント値を生成するステップと、

前記コミットメント値を少なくとも1つの別の寄与ノードに送信するステップと、

前記少なくとも1つの別の寄与ノードから他のコミットメント値を受信するステップと、

前記受信されたコミットメント値および前記ハッシュされた初期寄与に基づいて検証署名を生成するステップと、

前記他のコミットメント値を前記受信するステップの後、前記第1の寄与および前記検証署名を前記参加するノードに送信するステップとを備え、

前記ノード識別子が、少なくとも前記第1の寄与および前記初期寄与の関数として、前記参加するノードによって生成される

ことを特徴とする方法。

【請求項 57】

前記第1の寄与は、ランダム値または擬似ランダム値のいずれかであることを特徴とする請求項 56 に記載の方法。

【請求項 58】

前記検証署名は、前記コミットメント値および前記ハッシュされた初期寄与を結び付けて、他のノードが、その後、前記ノード識別子の真正性を検証することを許すことを特徴とする請求項 56 に記載の方法。

【請求項 59】

前記コミットメント値はまず、前記コミットメント値を収集して、後に、そのコミットメント値を前記寄与ノードに配信する紹介ノードに送信されることを特徴とする請求項 56 に記載の方法。

【請求項 60】

参加するノードからのハッシュされた初期寄与を含む、寄与を求める寄与要求を受信するための寄与要求レシーバと、

第1の寄与を取得するためのナンバジェネレータと、

前記第1の寄与からコミットメント値を生成するためのコミットメントジェネレータと、

前記コミットメント値を少なくとも1つの別の寄与ノードに送信するためのコミットメントディストリビュータと、

前記少なくとも1つの別の寄与ノードから他のコミットメント値を受信するための寄与収集モジュールと、

前記受信されたコミットメント値および前記ハッシュされた初期寄与に基づいて検証署名を生成するための検証署名ジェネレータと、

前記他のコミットメント値を前記受信した後、前記第1の寄与および前記検証署名を前記参加するノードに送信するための寄与配信モジュールとを備え、

前記ノード識別子が、少なくとも前記第1の寄与および前記初期寄与の関数として、前記参加するノードによって生成される

ことを特徴とする寄与ノードデバイス。

【請求項 61】

前記検証署名は、前記コミットメント値および前記ハッシュされた初期寄与を結び付けて、他のノードが、その後、前記ノード識別子の真正性を検証することを許すことを特徴とする請求項60に記載のデバイス。

【請求項62】

参加するノードからのハッシュされた初期寄与を含む、寄与を求める寄与要求を受信するための手段と、

第1の寄与を取得するための手段と、

前記第1の寄与からコミットメント値を生成するための手段と、

前記コミットメント値を少なくとも1つの別の寄与ノードに送信するための手段と、

前記少なくとも1つの別の寄与ノードから他のコミットメント値を受信するための手段と、

前記受信されたコミットメント値および前記ハッシュされた初期寄与に基づいて検証署名を生成するための手段と、

前記他のコミットメント値を前記受信した後、前記第1の寄与および前記検証署名を前記参加するノードに送信するための手段とを備え、

前記ノード識別子が、少なくとも前記第1の寄与および前記初期寄与の関数として、前記参加するノードによって生成される

ことを特徴とする寄与ノードデバイス。

【請求項63】

参加するノードからのハッシュされた初期寄与を含む、寄与を求める寄与要求を受信し、

第1の寄与を取得し、

前記第1の寄与からコミットメント値を生成し、

前記コミットメント値を少なくとも1つの別の寄与ノードに送信し、

前記少なくとも1つの別の寄与ノードから他のコミットメント値を受信し、

前記受信されたコミットメント値および前記ハッシュされた初期寄与に基づいて検証署名を生成し、さらに

前記他のコミットメント値を前記受信した後、前記第1の寄与および前記検証署名を前記参加するノードに送信するように適合された処理回路を含み、

前記ノード識別子が、少なくとも前記第1の寄与および前記初期寄与の関数として、前記参加するノードによって生成される

ことを特徴とするプロセッサ。

【請求項64】

1または複数のプロセッサによって実行されると、前記プロセッサに、

参加するノードからのハッシュされた初期寄与を含む、寄与を求める寄与要求を受信させ、

第1の寄与を取得させ、

前記第1の寄与からコミットメント値を生成させ、

前記コミットメント値を少なくとも1つの別の寄与ノードに送信させ、

前記少なくとも1つの別の寄与ノードから他のコミットメント値を受信させ、

前記受信されたコミットメント値および前記ハッシュされた初期寄与に基づいて検証署名を生成させ、さらに

前記他のコミットメント値を前記受信した後、前記第1の寄与および前記検証署名を前記参加するノードに送信させる、前記参加するノードがピアツーピア・オーバーレイ・ネットワークに関する前記ノード識別子を生成するのを支援するためにノードにおいて動作する命令を備え、

前記ノード識別子が、少なくとも前記第1の寄与および前記初期寄与の関数として、前記参加するノードによって生成される

ことを特徴とするマシン可読媒体。

【発明の詳細な説明】

10

20

30

40

50

【技術分野】

【0001】

様々な特徴は、無線通信システムに関する。少なくとも1つの態様は、ピアツーピア・オーバーレイ・ネットワークにおいてノード識別子を生成するため、および/または割り当てるための方法に関する。

【背景技術】

【0002】

構造化されたピアツーピア・ネットワークは、スケーラビリティ、およびクエリの決定論的ルーティングのために分散ハッシュテーブル(DHT)を使用する。分散ハッシュテーブル(DHT)は、ハッシュテーブルと同様のルックアップサービスを提供する或るクラスの非集中型の分散されたシステムであり、(ノード名, 識別子値)ペアが、DHTの中に格納され、いずれの参加するノードも、所与のノード名に関連する識別子値を効率的に取り出すことができる。各ノードが、DHTの中に一意のIDを有し、同様に、ネットワークに格納された各サービスまたは各オブジェクトが、一意のIDを有する。すべてのIDは、衝突を回避するように通常は非常に大きい同一の識別子空間の一部である。IDを維持すること(例えば、ノード名から識別子値IDにマップすること)の責任は、参加するノードのセットの変更が最小限の混乱しかもたらさないように、ノードの間で分散される。このことは、DHTが、極めて大きい数のノードにまで拡大されることを許すとともに、絶え間ないノード出現(arrival)、ノード離脱(departure)、およびノード障害を扱うことを許す。DHTは、分散ファイルシステム、ピアツーピアファイル共有-コンテンツ配信システム、協調的ウェブキャッシング、マルチキャスト、ユニキャスト、ドメイン名サービス、およびインスタントメッセージングなどの、より複雑なサービスを構築するのに使用され得るインフラストラクチャを形成する。

【0003】

しかし、1つの弱点は、DHTがアクセス制御のための組み込まれた機構を全く有さないことである。任意のノードが、複数のIDを取得すること(すなわち、ノードがネットワークのセキュリティを転覆させるように複数のIDを偽造することが可能なシビル(Sybil)攻撃)、複数の場所でネットワークに参加すること、および/またはオーバーレイルーティングサービス、ストレージサービス、およびその他のサービスを混乱させる可能性がある。したがって、悪意のあるノードが、DHTネットワークトポロジの効率的な動作を混乱させる可能性がある。悪意のあるノードの割合が f である場合、オーバーレイ・ネットワークにおける所与のパスに悪意のあるノードが無い確率は、 $(1-f)^h$ であり、ただし、 h は、パス長である。この確率は、パス長 h が増加するにつれ、 f (すなわち、悪意のあるノードの割合)が非常に小さい場合でさえ、急速に低下する。例えば、 2^{20} 個のノード(約100万個のノード)のネットワークにおいて、3パーセントのノードに悪意がある場合、20ホップの所与のパスに悪意のあるノードが無い確率は、約54パーセントである。

【0004】

敵対者が任意のノードIDの所有権を主張することが十分あり得る。このため、ノードIDは、公平な第三者によって独立に検証可能でなければならない。

【0005】

さらに、DHTは選択ロケーション攻撃(chosen location attack)に対しても脆弱である。ノードIDは、通常、以下のとおり生成される。すなわち、ノードID=hash(ランダムな入力)である。敵対者は、その入力の中のビットを注意深く選択することによって、わずかな計算作業でオーバーレイ・ネットワーク内の特定の領域に加わることを選択することが可能である。サイズ 2^{160} のオーバーレイ・ネットワークにおけるサイズ 2^x の領域内のいずれの場所に加わるのにも、敵対者は、 $O(2^{160-x})$ 回のオフライン計算を実行し、ただし、 $O(f(x))$ は、 $f(x)$ のオーダの計算複雑度を表す。同様に、オーバーレイ・ネットワーク内のノードの様な配置が想定される場合、 2^m 個のノードを含むオーバーレイ・ネットワーク内で選択された2つのノード間に着地するために必要なオフライン計算作業は、 $O(2^m)$ である。つまり、100万ノードネットワークの場合でさえ、選択された2つのノード間に着

地するために必要な計算作業は、手が出ないことは全くない、 2^{20} 回の動作である。暗号対策を使用する際の現行の想定は、 $O(2^{128})$ 回のオフライン計算、および $O(2^{80})$ 回のオンライン計算を実行することができる敵対者から保護することである。

【0006】

このため、選択ロケーション攻撃は完全に実行可能である。選択ロケーションは、敵対者が、2つのノード間の或る領域、またはネットワーク内の或る領域を支配することを許す。すると、敵対者は、選択されたノードの通信を混乱させること、またはそのロケーションにマップされるオブジェクトおよびサービスへのアクセスを制御することができる。つまり、選択ロケーション攻撃の影響も大きい。これらの攻撃の実行可能性および潜在的な影響からして、我々が、これらの攻撃に対して防御することは絶対に必要である。

10

【0007】

選択ロケーション攻撃において、敵対者は、所望されるノードIDを取得するようにランダムな入力の中のビットを注意深く選択する。このため、ハッシュ関数への入力のランダム性も検証可能でなければならない。

【発明の概要】

【発明が解決しようとする課題】

【0008】

したがって、ノードID生成および/またはノードID割当てを保護することによって、オーバーレイ・ネットワークにおける敵対的ノード攻撃を防止する、または最小限に抑える方法が必要とされる。

20

【課題を解決するための手段】

【0009】

参加するノードが、その参加するノードのノード識別子を生成するのに、ピアツーピア・オーバーレイ・ネットワークにおける寄与(contributor)ノードによって供給される寄与を使用する、マルチパーティコミットメント方法が提供される。参加するノードが、第1の寄与を生成し、参加要求を紹介(introducer)ノード(または複数の寄与ノード)に送信し、ただし、参加要求は、オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するための1または複数の寄与を取得することを求める。第1の寄与のハッシュが、参加要求の一部として含められることが可能である。これに応答して、参加するノードは、複数の寄与を受信することが可能であり、これらの寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、第1の寄与にも結び付けられる。参加するノードは、次に、第1の寄与および受信された寄与の関数として、参加するノードのノード識別子を生成する。このようにして、ノード間の結託が防止され得る。

30

【0010】

1つの特徴によれば、ピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するための、参加するノード上で動作する方法が提供される。第1の寄与が、参加するノードによって生成される。第1のハッシュが、第1の寄与に基づいて参加するノードによって生成されることが可能であり、ただし、参加要求は、検証署名を計算する際に複数の寄与ノードによって使用される第1のハッシュ値を含む。参加するノードは、次に、紹介ノードに参加要求を送信し、ただし、参加要求は、オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するための1または複数の寄与を取得することを求める。この参加要求は、第1の寄与の第1のハッシュ値を含むことが可能であり、第1のハッシュ値は、その後、複数の寄与ノードに供給されて、検証署名を計算する際に使用されることに留意されたい。これに応答して、参加するノードは、紹介ノードから複数の寄与を受信することが可能であり、これらの寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、第1の寄与にも結び付けられる。これらの寄与は、ランダム値または擬似ランダム値のいずれかであることが可能である。さらに、参加するノードは、紹介ノードから複数の検証署名を受信することも可能であり、各検証署名は、複数の寄与ノードによって生成された複数のコミットメント値の関数であり、これらのコミットメント値は、寄与ノードによって供給される寄与の関数である。各検証署

40

50

名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、これらのコミットメント値と一緒に結び付けることが可能であり、その結果、寄与と一緒に結び付けることが可能である。参加するノードは、次に、第1の寄与および受信された寄与の関数としてノード識別子を生成することが可能である。このノード識別子は、オーバーレイ・ネットワークにおける1または複数の隣接ノードに配信されることが可能であり、このノード識別子の有効性は、その1または複数の隣接ノードによって独立に検証され得る。

【0011】

一例において、参加するノードは、寄与ジェネレータ、参加要求モジュール、寄与収集モジュール、ノード識別子ジェネレータ、ハッシュジェネレータ、および/またはノード識別子配信モジュールを備えるネットワークデバイスであることが可能である。寄与ジェネレータは、第1の寄与を生成するように構成されることが可能である。ハッシュジェネレータは、第1の寄与に基づいて第1のハッシュ値を生成するように適合されることが可能であり、ただし、参加要求は、検証署名を計算する際に複数の寄与ノードによって使用される第1のハッシュ値を含む。参加要求モジュールは、参加要求を紹介ノードに送信するように適合されることが可能であり、ただし、この参加要求は、オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するための1または複数の寄与を取得することを求める。この参加要求は、第1の寄与の第1のハッシュ値を含むことが可能であり、ただし、第1のハッシュ値は、複数の寄与ノードに供給されて、検証署名を計算する際に使用される。寄与収集モジュールは、紹介ノードから複数の寄与を受信するように適合されることが可能であり、これらの寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、第1の寄与にも結び付けられる。これらの寄与は、ランダム値または擬似ランダム値のいずれか1つであることが可能である。寄与収集モジュールは、紹介ノードから複数の検証署名を受信するようにさらに構成されることが可能であり、各検証署名は、複数の寄与ノードによって生成される複数のコミットメント値の関数であり、これらのコミットメント値は、寄与ノードによって供給される寄与の関数である。各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、これらのコミットメント値と一緒に結び付けることが可能であり、その結果、寄与と一緒に結び付けることが可能である。ノード識別子ジェネレータは、第1の寄与および受信された寄与の関数としてノード識別子を生成するように適合されることが可能である。ノード識別子配信モジュールは、オーバーレイ・ネットワークにおける1または複数の隣接ノードにノード識別子を配信するように適合されることが可能であり、ノード識別子の有効性は、その1または複数の隣接ノードによって独立に検証され得る。

【0012】

別の特徴によれば、参加するノードがピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するのを支援するための、紹介ノード上で動作する方法が提供される。オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するための1または複数の寄与を取得することを求める参加要求が、参加するノードから受信される。この参加要求は、第1の寄与の第1のハッシュ値を含むことが可能であり、ただし、第1のハッシュ値は、寄与ノードに供給されて、検証署名を計算する際に使用される。紹介ノードは、オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信することが可能である。一例において、寄与要求が送信される寄与ノードの数は、オーバーレイ・ネットワークにおける敵対的ノードの数より少なくとも1つ多いことが可能である。寄与要求が送信される寄与ノードは、ランダムに選択されるか、または擬似ランダムに選択されるかのいずれかである。これに応答して、紹介ノードは、寄与ノードから複数のコミットメント値を受信することが可能であり、各コミットメント値は、寄与ノードのそれぞれにおいて生成される対応する寄与の関数である。次に、紹介ノードは、その複数のコミットメント値を寄与ノードに配信することが可能である。紹介ノードは、その複数のコミットメント値を配信することに応答して、複数の寄与、および対応する検証署名を受信することが可能であり、各検証署名は、その複数のコミットメント値の関数である。一例において、寄与と検証署名のペアが、複数の寄与ノードのそれぞれから受信

10

20

30

40

50

されることが可能である。寄与は、ランダム値または擬似ランダム値のいずれか1つであることが可能である。各検証署名は、参加するノードからの第1の値の関数であることも可能である。各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、これらのコミットメント値と一緒に結び付けることが可能であり、その結果、寄与と一緒に結び付けることが可能である。複数の寄与は、次に、参加するノードに送信されることが可能であり、ただし、ノード識別子は、それらの寄与の関数として、参加するノードによって生成される。コミットメント値および検証署名は、参加するノードに寄与を送信することに先立って検証されることが可能であることに留意されたい。また、検証署名は、参加するノードに送信されることが可能である。複数のコミットメント値は、寄与ノードによってその他の寄与ノードに直接に配信されることが可能である。

10

【0013】

さらに、複数の寄与要求が送信される寄与ノードの数は、減らされることが可能であり、寄与ノードは、ピアツーピア・オーバーレイ・ネットワークにおけるノードのより大きいセットからのノードの縮小されたセットである。ノードの縮小されたセットは、ノード間の結託の可能性を回避するようにランダムに選択されることが可能である。一例において、寄与ノードのランダムな選択は、参加するノードに関する公開鍵に基づくことが可能である。別の例において、寄与ノードのランダムな選択は、紹介ノードに関する公開鍵に基づくことが可能である。1つの実例において、値 x が、紹介ノードに関する公開鍵に基づいて生成され、この値 x は、オーバーレイ・ネットワークにおける値 x に関するそのポイントの責任を負うノード n_c を見出すのに使用され、次に、ノード n_c が、それぞれ、オーバーレイ・ネットワークにおけるノード n_c の先行ノードおよび後続ノードである、前のノード n_{c-prev} および次のノード n_{c-next} を特定するのに使用され、ただし、前のノードおよび次のノードは、寄与ノードとして利用される。別の実例において、値 x が、紹介ノードに関する公開鍵、および第1の文字列に基づいて生成され、値 y が、紹介ノードに関する公開鍵、および第2の文字列に基づいて生成され、値 z が、紹介ノードに関する公開鍵、および第3の文字列に基づいて生成され、次に、値 x 、 y 、および z が、オーバーレイ・ネットワークにおける対応する値 x 、 y 、および z の責任を負う複数のノード n_x 、 n_y 、および n_z を見出すのに使用され、これらの複数のノード n_x 、 n_y 、および n_z が、寄与ノードとして使用される。

20

【0014】

一例において、紹介ノードは、要求レシーバ、寄与要求モジュール、コミットメントレシーバ、コミットメント配信モジュール、検証モジュール、寄与収集モジュール、および/または寄与配信モジュールを備えるネットワークデバイスであることが可能である。要求レシーバは、オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するための1または複数の寄与を取得することを求める、参加するノードからの参加要求を受信するように適合されることが可能である。参加要求は、参加するノードからの第1の寄与の第1のハッシュ値を含むことが可能であり、ただし、第1のハッシュ値は、寄与ノードに供給されて、検証署名を計算する際に使用される。寄与要求モジュールは、オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信するように適合されることが可能である。一例において、寄与要求が送信される寄与ノードの数は、オーバーレイ・ネットワークにおける敵対的ノードの数より少なくとも1つ多いことが可能である。別の例において、寄与要求が送信される寄与ノードは、ランダムに選択されるか、または擬似ランダムに選択されるかのいずれかである。コミットメントレシーバは、寄与ノードから複数のコミットメント値を受信するように適合されることが可能であり、各コミットメント値は、寄与ノードのそれぞれにおいて生成される対応する寄与の関数である。コミットメント配信モジュールが、複数のコミットメント値を寄与ノードに配信するように適合されることが可能である。代替として、複数のコミットメント値は、寄与ノードによって他の寄与ノードに直接に配信されてもよい。寄与収集モジュールは、複数のコミットメント値が配信されることに応答して、複数の寄与、および対応する検証署名を受信するように適合されることが可能であり、各検証署名は、複数のコミットメント値

30

40

50

の関数である。また、寄与収集モジュールは、寄与ノードのそれぞれから検証署名を受信するように適合されることも可能であり、各検証署名は、コミットメント値の関数である。また、各検証署名は、参加するノードからの第1の値の関数でもある。各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、これらのコミットメント値と一緒に結び付けることが可能であり、その結果、寄与と一緒に結び付けることが可能である。検証モジュールは、寄与に参加するノードに送信するのに先立って、これらのコミットメント値および検証署名を検証するように適合されることが可能である。寄与配信モジュールは、複数の寄与に参加するノードに送信するように適合されることが可能であり、ただし、ノード識別子は、それらの寄与の関数として、参加するノードによって生成される。寄与は、ランダム値または擬似ランダム値のいずれかであることが可能である。寄与配信モジュールは、参加するノードに検証署名をさらに送信することが可能である。

10

【0015】

さらに別の特徴によれば、参加するノードがピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するのを支援するための、寄与ノード上で動作する方法が提供される。寄与ノードは、寄与を求める寄与要求を受信することが可能であり、ただし、この寄与要求は、参加するノードからのハッシュされた初期寄与を含むことが可能である。次に、第1の寄与が、寄与ノードによって取得される、または生成される。第1の寄与は、ランダム値または擬似ランダム値のいずれかであることが可能である。次に、寄与ノードが、第1の寄与からコミットメント値を生成することが可能である。コミットメント値は、寄与ノードによって少なくとも1つの別の寄与ノードに送信されることが可能である。一例において、コミットメント値はまず、コミットメント値を収集し、その後、コミットメント値を寄与ノードに配信する紹介ノードに送信される。他のコミットメント値が、その少なくとも1つの別の寄与ノードから受信されることが可能である。検証署名が、受信されたコミットメント値、およびハッシュされた初期寄与に基づいて寄与ノードによって生成されることが可能である。検証署名は、コミットメント値とハッシュされた初期寄与を結び付けて、他のノードが、その後、ノード識別子の真正性を検証することを許す。次に、寄与ノードが、その他のコミットメント値を受信した後、第1の寄与および/または検証署名に参加するノードに送信することが可能であり、ただし、ノード識別子は、少なくとも第1の寄与および初期寄与の関数として、参加するノードによって生成される。

20

30

【0016】

一例において、寄与ノードは、寄与要求レシーバ、ナンバジェネレータ、コミットメントジェネレータ、コミットメントディストリビュータ、寄与収集モジュール、検証署名ジェネレータ、および/または寄与配信モジュールを備えるネットワークデバイスであることが可能である。寄与要求レシーバは、寄与を求める寄与要求を受信するように適合されることが可能であり、寄与要求は、参加するノードからのハッシュされた初期寄与を含む。ナンバジェネレータは、第1の寄与を取得するように適合されることが可能である。コミットメントジェネレータは、第1の寄与からコミットメント値を生成するように適合されることが可能である。コミットメントディストリビュータは、少なくとも1つの別の寄与ノードにコミットメント値を送信するように適合されることが可能である。寄与収集モジュールは、その少なくとも1つの別の寄与ノードから他のコミットメント値を受信するように適合されることが可能である。検証署名ジェネレータは、受信されたコミットメント値、およびハッシュされた初期寄与に基づいて、検証署名を生成するように適合されることが可能である。検証署名は、コミットメント値とハッシュされた初期寄与を結び付けて、他のノードが、その後、ノード識別子の真正性を検証することを許すことが可能である。その他のコミットメント値を受信した後、第1の寄与、および検証署名に参加するノードに送信するための寄与配信モジュール、ただし、ノード識別子は、少なくとも第1の寄与および初期寄与の関数として、参加するノードによって生成される。

40

【0017】

本発明の態様の特徴、性質、および利点は、すべての図面にわたって同様の参照符号が

50

それに相応した識別をする図面と併せて解釈されると、後段で示される詳細な説明から、より明白となることが可能である。

【図面の簡単な説明】

【0018】

【図1】DHTがどのように、ピアツーピア・オーバーレイ・ネットワークの動作を円滑にするように機能することが可能であることを示す図である。

【図2】集中型レジストラ(信頼される機関)がどのように、オーバーレイ・ネットワーク内の複数のノードにノードIDを割り当てる役割をすることが可能であることを示すブロック図である。

【図3】集中型レジストラによるオーバーレイ・ネットワークのDHTに関するノードIDの生成および割当てを示す図である。 10

【図4】オーバーレイ・ネットワークに参加するノードによる、ただし、集中型レジストラからの入力を用いた、ノードIDの生成および割当てを示す図である。

【図5】参加するノードがオーバーレイ・ネットワークのDHTに参加することを許すのに先立つノードIDの検証を示す図である。

【図6】分散型レジストラによるオーバーレイ・ネットワークのDHTに関するノードIDの生成および割当てを示す図である。

【図7】ノードIDが、オーバーレイ・ネットワークにおける他の1または複数のノードからの寄与または支援を用いて、参加するノードによって生成されることが可能な、マルチパーティコミットメントモデルを示す図である。 20

【図8】ノードIDが、オーバーレイ・ネットワークにおける他の1または複数のノードからの寄与または支援を用いて、参加するノードによって生成されることが可能な、マルチパーティコミットメント方法を示す図である。

【図9】マルチパーティコミットメントモデルにおけるノードIDの検証を示す図である。

【図10】ピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するために、参加するノードにおいて動作する方法の例を示す図である。

【図11】オーバーレイ・ネットワークのノードとして動作することを求めていることが可能なデバイスの例を示すブロック図である。

【図12】参加するノードの機能構成要素の例を示すブロック図である。

【図13】参加するノードがピアツーピア・オーバーレイ・ネットワークにおいてノード識別子を取得するのを円滑にするように、紹介ノードにおいて動作する方法の例を示す図である。 30

【図14】参加するノードによるノード識別子の生成を円滑にするようにオーバーレイ・ネットワークの紹介ノードとして動作することが可能なデバイスの例を示すブロック図である。

【図15】紹介ノードの機能構成要素の例を示すブロック図である。

【図16】参加するノードがピアツーピア・オーバーレイ・ネットワークにおいてノード識別子を取得するのを円滑にするように、寄与ノードにおいて動作する方法の例を示す図である。

【図17】参加するノードによるノード識別子の生成を円滑にするようにオーバーレイ・ネットワークにおける寄与ノードとして動作することが可能なデバイスの例を示すブロック図である。 40

【図18】寄与ノードの機能構成要素の例を示すブロック図である。

【発明を実施するための形態】

【0019】

以下の説明において、実施形態の完全な理解をもたらすように特定の詳細が与えられる。しかし、これらの実施形態は、これらの特定の詳細なしに実施されることも可能であることが、当業者には理解されよう。例えば、回路は、不必要な詳細で、これらの実施形態を不明瞭にしないようにブロック図で示されることが可能である。その他、よく知られた回路、構造、および技術は、これらの実施形態を不明瞭にしないように詳細には示されな 50

い可能性がある。

【0020】

本明細書で使用される「識別」、「識別子」、および「ID」という用語は、特定のノードを識別する役割をする値を指すように互いに区別なく使用されることが可能である。

【0021】

(概観)

1つの特徴は、参加するノードが、ネットワークにおける寄与ノードによって供給されたランダムな入力を使用して、参加するノードのIDを生成するマルチパーティコミットメント方法を提供する。参加するノードとその他のノードの間の結託を防止するのに、紹介ノードが、いずれのノードが参加するノードに関する寄与ノードの役割をすべきかをランダムに選択する。寄与ノードによって供給されたランダムな入力は、ノードIDへの入力寄与のすべてが一緒に結び付けられて、敵対的な参加するノードがそのノード自らのノードIDを選ぼうと試みることを防止するように、他のすべての寄与ノードに供給されることが可能である。その後、参加するノードがオーバーレイ・ネットワークに加えられることを求めると、検証ノードが、寄与ノードからの入力寄与を使用して、参加者のノードIDが有効であるかどうかを検証することができる。

【0022】

(DHTオーバーレイ・ネットワーク)

通信ネットワーク内で、各ネットワークデバイスまたはノードが、そのネットワークデバイスに、および/またはそのネットワークデバイスからメッセージをルーティングするのに使用されるIP(インターネットプロトコル)アドレスを有する。そのようなネットワークIP層の上で、各ノードのノード識別子(すなわち、ノードID)を使用するオーバーレイ・ネットワークを実施することが可能である。したがって、ノードのIPアドレスが変わったとした場合でも、そのノードのノードIDは、オーバーレイ・ネットワークに関して同一のままである。このため、オーバーレイ・ネットワークは、IP層インフラストラクチャに依拠せず、代わりに、オーバーレイ・ネットワークのノード間のピアツーピア通信のためにノードIDを使用する。さらに、そのようなオーバーレイ・ネットワークは、オーバーレイ・ネットワークがいずれの単一の障害ポイントによっても損なわれることがないように、オーバーレイ・ネットワークのノードにわたる分散ハッシュテーブルを実施することが可能である。

【0023】

図1は、DHTがどのように、ピアツーピア・オーバーレイ・ネットワークの動作を円滑にするように機能することが可能であることを示す図である。この例は、オーバーレイ・ネットワークをリングとして図示するが、このことは、簡明のために過ぎず、他のオーバーレイ・ネットワークアーキテクチャも可能であり、企図される。ピアツーピア・オーバーレイ・ネットワークA100は、複数のノードA102、B104、C106、D108、E110、および/またはF112によって形成されることが可能である。各ノードは、オーバーレイ・ネットワークA100に関するDHTの一部分(例えば、ノード名、ノード識別子)を維持することが可能である。各ノードは、識別子空間から選択された、オーバーレイ・ネットワーク100のDHT114における一意のノード識別子(ノードID)を有する。同様に、ネットワーク100に格納された各サービスまたは各オブジェクトが、やはり識別子空間から選択された一意の識別子を有することが可能である。すべてのノードIDは、衝突を回避するように、通常、非常に大きい、同一の識別子空間の一部であることが可能である。ノードIDを維持すること(例えば、ノード名からノード識別子にマップすること)の責任は、参加するノードのセットの変更が最小限の混乱しかもたらさないように、ノードの間で分散される。各ノードは、通常、そのノード自らのノードIDから始まり、そのノードの隣接ノードのうち1または複数のノードのノードIDに至る、識別子空間の一部分の責任を負うことが可能である。ノードは、そのノードが責任を負う範囲内の任意のロケーションにマップされるオブジェクトを格納すること、および/またはそのようなオブジェクトに関するクエリに応答することの責任を負うことが可能である。各ノードは、他のノード(そのノードの隣接ノードまたはルー

10

20

30

40

50

ティングテーブル)へのリンクのセットを維持することが可能である。一緒になって、これらのリンクは、オーバーレイ・ネットワークを形成する。ノードは、ネットワークのトポロジと呼ばれる或る構造に従って、そのノードの隣接ノードを選ぶ。

【0024】

暗号ハッシュ関数が、オーバーレイ・ネットワーク100内のノード、オブジェクト、およびサービスのインスタンスにID(識別子)を割り当てるのに使用されることが可能である。ほとんどのDHTは、ハッシュ鍵(すなわち、識別子)をノードおよび/またはサービスにマップする整合性のあるハッシングの何らかの変種を使用する。この技術は、ノード間の地理的距離またはネットワーク待ち時間と無関係である、鍵 k_1 から鍵 k_2 までの距離の抽象的概念を定義する関数 $\delta(k_1, k_2)$ を使用する。各ノードには、そのノードの識別子(すなわち、ノードID)と呼ばれる単一の鍵が割り当てられる。識別子値(ノードID) i を有するノードは、識別子 i が δ に従って測定された最も近いIDである、すべての鍵(または識別子)を所有する(またはそのような鍵の責任を負う)。DHTネットワークトポロジは、任意の鍵 k に関して、ノードが、 k を所有する、または上記に定義される鍵空間距離の点で k により近いノードに至るリンクを有するという特性の何らかの変種を共有する。すると、貪欲アルゴリズムを使用して任意の鍵 k (すなわち、識別子)の所有者にメッセージをルーティングすることが、比較的簡単である。例えば、各ノードにおいて、メッセージは、ノードIDが鍵 k に最も近い隣接ノードに転送される。そのような隣接ノードが全く存在しない(すなわち、他のいずれの隣接ノードも、より近いノードIDを有さない)場合、そのメッセージは、上記に定義される鍵 k の所有者である、最も近いノードに着信しているものと想定される。このスタイルのルーティングは、ときとして、鍵ベースのルーティングと呼ばれる。一例において、DHTは、円の上のポイント(図1に示される)を鍵として扱うことが可能であり、 $\delta(k_1, k_2)$ は、この円を k_1 から k_2 まで時計回りに移動する距離である。このため、この円形の鍵空間は、エンドポイントがノード識別子(ノードID)である隣接するセグメントに分割される。 i_1 と i_2 が隣り合う2つのIDである場合、 ID_{i_2} を有するノードは、 i_1 と i_2 の間に入るすべての鍵を所有する。このネットワークトポロジを実施する際の効率上の考慮のため、任意の経路におけるホップの最大数(経路長)が小さいことを保証して、要求が迅速に完了するようにし、さらに任意のノードの隣接ノードの最大数(最大ノード度)が小さいことを保証して、メンテナンスオーバーヘッドが過度にならないようにすることが賢明である可能性がある。1つのトレードオフは、より短い経路が、各ノードに関していくつかの隣接ノード(すなわち、より高い最大ノード度)を要求することであることに留意されたい。

【0025】

一例において、ノードC106は、ノードC106が1または複数の先行ノード、および1または複数の後続ノードに関するノードIDを保持するDHT114を維持することが可能である。DHT114は、すべてのノードIDのサブセット、すなわち、ノードC106のノードIDに近接したノードIDだけを維持することが可能である。例えば、ノードC116に関するDHT114は、ノードA102およびB104に関する先行するノードID、ならびにノードD108、E110、およびF112に関する後続のノードIDを維持することが可能である。DHTは、極めて大きい数のノード(例えば、ネットワーク100における 2^{16} 個のノード)まで拡大されることが可能であり、絶え間ないノード出現、ノード離脱、およびノード障害を扱うことができる。一例において、DHTは、ノードIDを円の上のポイントとして扱うことが可能であり、さらに $\delta(\text{ノードID}_{i_1}, \text{ノードID}_{i_2})$ が、この円を時計回りに移動する距離(例えば、ノードCから何らかのノードID x までの)である。このため、この円形の識別子空間は、エンドポイントがノード識別子(ノードID)である隣接するセグメントに分割される。 i_1 と i_2 が隣り合う2つのノードIDである場合、 ID_{i_2} を有するノードは、 i_1 と i_2 の間に入るすべての鍵を所有する。したがって、DHTは、より複雑なサービス(例えば、分散型ファイルシステム、ピアツーピアファイル共有-コンテンツ配信システム、協調的ウェブキャッシング、マルチキャスト、ユニキャスト、ドメイン名サービス、およびインスタントメッセージング)を構築するのに使用され得るインフラストラクチャを形成することができ、ただし、そのような各サービ

スは円形の識別子空間において識別子と関連付けられることが可能である。或る特定のサービス識別子に値が最も近いノードIDを有するノードが、そのようなサービスをホストする(例えば、或る特定のファイルを格納する、またはその「サービス」識別子によって識別されるサービスを提供するなど)ものと想定されることが可能である。例えば、DHT114を参照すると、或るサービスが「79」という識別子値を有する場合、この値は、ノードC 106に関連するノードID「73」に最も近く、したがって、そのようなサービスを求めるいずれの要求も、最終的にノードC 106に行き着く。

【0026】

(セキュリティで保護されたDHTノードID生成)

従来のオーバーレイ・ネットワークがさらされやすい可能性がある様々なタイプの攻撃から保護するDHTノードID生成のための様々な方法が本明細書で説明される。一般に、本明細書で説明されるDHTノードID生成スキームは、(a)ノードIDの一樣な分布、(b)ノードIDの独立した検証、(c)ノードIDを生成する際のランダム性、および/または(d)ノードIDを生成する際に行われた作業の証拠を要求することによってそのような攻撃を妨げようと、または防止しようと試みる。

【0027】

ノードIDの一樣な分布 第1に、ノードIDが、DHT内でノードIDが一樣に分布するように理想的に生成されることが可能である。暗号ハッシュ関数を使用して生成されたノードIDが、この基準を満たす。例えば、SHA-1などの暗号ハッシュ関数の使用が、入力がランダムであるものと想定すると、DHTを巡るノードのランダムな分布を実現するのを助ける。一樣な分布は、ノードが、オーバーレイ・ネットワーク内でID空間のほぼ同等の所有権を有することを確実にするのに望ましい。しかし、ノード離脱(例えば、移動ノードが異なる領域に移動した際の)が、DHTにおけるノードIDの一樣な分布を乱す可能性があり、その結果、この特性は、常に一貫して保たれるわけではない可能性がある。

【0028】

ノードIDの独立した検証 第2に、ノードIDが独立して検証されることが可能である。つまり、ノードIDが、識別子を主張するノードに実際に属することを第三者が独立に検証することができなければならない。さらに、任意の第三者がノードID導出関数への入力からノードIDを導き出すことが可能でなければならない。このことは、第三者が、ノードIDを導き出すのに使用されたとされる入力の、そのノードIDを主張するノードに対する関連を検証することを許す。そのようなノードID検証を実現するのに、ノードは、公開鍵と秘密鍵とを備える非対称鍵ペアを生成することが可能である。次に、ノードは、そのノードの公開鍵をハッシュ関数への入力として使用して、そのノードのノードIDを生成することができる。ノードは、そのノードのノードID上の署名を、そのノードIDを検証するための証拠として提示することができる。そのような署名は、ノードの秘密鍵に基づくことが可能である。別のノードが同一のノードIDを主張するのに、そのようなノードは、ハッシュされて同一のノードIDになる公開鍵を有する別の非対称鍵ペアを生成するという困難な課題を有することになる。

【0029】

予測不能にランダムなノードID 第3に、ノードIDは、予測不能にランダムであることが可能である。このことは、ノードが、オーバーレイ・ネットワークに参加するのに先立って、DHT内でそのノードのIDを予測するのが計算上、困難であるべきことを暗示する。ノードIDのランダム性のため、ハッシュ関数への入力またはノードID自体が、ランダムまたは擬似ランダムでなければならない。そのような問題解決法のために2つのカテゴリが存在する。すなわち、(a)信頼される機関によるID割当ておよび/またはランダムな入力寄与、ならびに/あるいは(b)オーバーレイ・ネットワーク内のノードによる分散ノードID生成または分散ランダム入力寄与である。

【0030】

ノードIDを生成する際の作業の証拠 第4に、ノードIDを提示するノードが、そのノードIDを生成する際に行われた作業の証拠を提供することが可能である。作業の証拠は、計

10

20

30

40

50

算に関するものであることが可能である。代替として、作業の証拠は、攻撃する相手またはノードがDHT内で複数のノードIDを生成するのをより困難にするように、現実世界の資格証明または関係に関連付けられることが可能である。

【0031】

(信頼される機関(レジストラ)によるノードID割当て)

DHTノードID割当ての、このモデルでは、信頼される機関が、DHTオーバーレイ・ネットワーク内のノードIDのすべてを発行する。すべてのノードが信頼される機関と接触して、認定されたノードIDを得る。検証者(DHTのノード)は、信頼される機関、または信頼される機関の代理によって署名されたノードIDの主張だけしか受け付けない。信頼される機関は、シビル攻撃を弱める、または減速させるようにノードID割当てに対してアクセス制御機能および/またはレート制限機能を実行することが可能である。信頼される機関は、集中型レジストラであっても、分散型レジストラのセットであってもよい。

10

【0032】

(集中型レジストラ)

図2は、集中型レジストラ(信頼される機関)がどのように、オーバーレイ・ネットワーク内の複数のノードにノードIDを割り当てる役割をすることが可能であることを示すブロック図である。オーバーレイ・ネットワーク100に関して、集中型レジストラ202は、複数のネットワークノード102、104、106、108、110、112、116、および/または118にノードIDを与えて、DHT204を形成する役割をすることが可能である。ネットワークノードは、ノードIDを取得することを求めるたびに、ノード識別子の割当てを要求するメッセージを集中型レジストラ202に送信する。すると、集中型レジストラ202は、要求するノードに関するノードIDを計算し、要求するノードにそのノードIDを割り当て、要求するノードにそのノードIDを送信する。

20

【0033】

一部の実施形態において、集中型レジストラ202は、いずれのオーバーレイ・ネットワーク動作にも参加しない可能性があり、このため、他のノードによるノードID検証のために到達可能でさえない可能性がある。そのような集中型レジストラ202の一例は、要求するノードから公開鍵を受信し、ノードIDを割り当て、ノードIDのマッピングをノードにアサートする証明機関(CA)を含むことが可能である。例えば、証明機関(CA)は、要求するノードの公開鍵から導き出されたノードIDのマッピングに署名することが可能である。すると、要求するノードは、認定されたノードIDをその他のノードに提示し、証明書の中に含まれる公開鍵の所有を証明することが可能である。ノードは、伝送された或るコンテンツに、このノードの秘密鍵で署名することによって、この所有を証明することができる。その他のノードは、もたらされる署名されたコンテンツを、このノードの証明書によって認定された公開鍵を使用して検証することができる。オーバーレイ・ネットワーク固有の集中型レジストラが、所与のオーバーレイ・ネットワークの作成者、またはそのようなオーバーレイ・ネットワークの採用を推進することに関心のあるエンティティによって展開されることが可能であり、あるいは、そのようなレジストラは、ネットワーク事業者によって提供されることも可能である。

30

【0034】

集中型レジストラを使用することの1つの弱点は、ピアツーピア・オーバーレイ・ネットワークの潜在的なすべてのメンバ(例えば、ノード)が、集中型レジストラを所有し、運用する単一の機関を信頼することが困難であり得ることである。一部の実施形態において、オーバーレイ・ネットワークノードが、ノードID割当てのために外部の機関(すなわち、集中型レジストラ)に完全に頼るのではなく、代わりに、集中型レジストラ202が、ランダムな入力をハッシュ関数に与え、ノード自体が別のランダムな入力をハッシュ関数に与えることが可能である。これら2つの入力のそれぞれは、ノードIDが、そのノードIDを主張するノードに属することを検証する際に、検証ノードによって使用される。

40

【0035】

集中型レジストラは、単一のエンティティまたはノードが取得することができるノード

50

IDの数を制限することが可能である。例えば、事業者によって所有される集中型レジストラの場合に、ノードIDは、サービス契約に結び付けられることが可能である。代替として、ノードIDは、クレジットカード番号などの何らかの価値のあるものに結び付けられてもよい。集中型レジストラが、いずれの価値のあるものにもノードIDを結び付けることなく、ノードIDを割り当てる場合、またはノードIDが貴重でない情報(例えば、電子メールアドレス、IPアドレス、またはMACアドレスなど)だけにしか結び付けられない場合、攻撃者がいくつかのノードIDを取得することがより容易になる。このことは、シビル攻撃を助長し、所望されるノードIDが取得されるまで、攻撃者がいくつかのノードIDを取得することを許す。

【0036】

10

(集中型レジストラによるノードID生成)

集中型レジストラがノードIDを生成して、オーバーレイ・ネットワークに参加するノードに割り当てることのできる、いくつかの方法が存在する。

【0037】

図3は、集中型レジストラによるオーバーレイ・ネットワークのDHTに関するノードIDの生成および割当てを示す図である。一部の事例において、ノードIDは、契約、クレジットカード番号もしくは他のレガシーID、または対称鍵ベースの資格証明に結び付けられることが可能である。他の事例において、ノードIDは、自己署名された証明書に結び付けられることが可能である。すべての事例において、集中型レジストラは、集中型レジストラが、各ノード(資格証明、クレジットカード番号、および/または自己署名された証明書を使用して識別された)にポリシーによって許されるより多くのノードIDを発行することがないことを確実にするように、ノードID割当てのレコードを保持するものと想定される。

20

【0038】

ノードIDを取得するのに、参加するノード302が、ランダムな第1のナンス(Nonce-J)を有するノードIDを求める要求を集中型レジストラ304に送信する。この要求を受信したことに応答して、集中型レジストラ304は、第2のナンス(Nonce-CR)308を生成し、受信された第1のナンス、および生成された第2のナンスを有するノードIDチャレンジ310に参加するノード302に送信する。一部の実施形態において、ノードIDチャレンジ310は、集中型レジストラ304に関する公開鍵証明書(CERT-CR)も含むことが可能である。本明細書全体にわたって、オプションの内容または動作は、角括弧(すなわち、「[...]」などの)に入れられる可能性があることに留意されたい。

30

【0039】

次に、参加するノード302は、第1のナンスおよび第2のナンス(Nonce-JおよびNonce-CR)、所有の証拠(POP)ペイロード(POP-J)、中央機関によって署名され得る、または自己署名され得るノード証明書(CERT-J)、ならびにメッセージ全体にわたるメッセージ署名(SIG-J)を含むチャレンジ応答メッセージ312を計算し、送信する。2つのナンスにわたるメッセージ署名(SIG-J)が、参加するノード302がオンライン計算またはライブの計算を行ったこと(その結果、オフライン計算を使用する攻撃を防止していること)を集中型レジストラ304に証明する。POPペイロード(POP-J)は、オプションであるが、参加するノード302によって、共有鍵に基づいて契約の所有を証明するのに、またはクレジットカード情報を提示するのに使用され得る。共有される秘密の所有を証明するのに、所有の証明ペイロード(POP-J)は、その前に、参加するノードと集中型レジストラの間の共有される秘密、例えば、製造中にプロビジョニングされたPIN(個人識別番号)、またはデフィーヘルマン鍵交換などの公開鍵暗号交換を介してネゴシエートされた対称鍵を使用する。参加するノード302は、集中型レジストラに所有を証明するために、共有される秘密を使用してノードIDチャレンジ応答にわたるメッセージ認証符号(MAC)を計算することが可能である。クレジットカード情報を送信するのに、参加するノード302は、集中型レジストラの公開鍵(公開鍵交換において前もって交換されていることが可能な)を使用して、その情報を暗号化することが可能である。また、メッセージ署名(SIG-J)が、メッセージ312の完全性を証明することも可能である。

40

50

【0040】

次に、集中型レジストラ304が、チャレンジ応答314を検証する。つまり、集中型レジストラ304は、例えば、参加するノード証明書CERT-J、所有の証拠POP-J、および/またはメッセージ署名SIG-Jを検証することによって、チャレンジ応答312を検証することが可能である。検証314が成功した場合、集中型レジストラ304は、ノードID316を生成して、第1のナンスおよび第2のナンス(Nonce-JおよびNonce-CR)、新たなノード識別子(Node ID)、ノードIDおよび参加するノードの公開鍵にわたる証明書(Node-CERT)、ならびにメッセージ全体にわたる署名(SIG-CR)を含むID応答メッセージ318を送信することが可能である。署名(SIG-CR)は、ノードIDを生成する際に集中型レジストラ304がオンラインまたはライブであったことを証明し、さらに参加するノード302に対してメッセージ318の出所を確立する。

10

【0041】

(集中型レジストラによるランダムな入力寄与)

代替の例において、ノードが集中型レジストラを信頼しない場合、ノードは、代わりに、2パーティコミットメントスキームを使用して、ノード自らのノードIDを生成するが、集中型レジストラからの入力を用いることが可能である。図4は、オーバーレイ・ネットワークに参加するノードによる、ただし、集中型レジストラからの入力を用いたノードIDの生成および割当てを示す図である。このモデルでは、いずれのパーティも(ノードも、集中型レジストラも)、ノードID生成関数へのランダムな入力を支配していない。参加するノード402が、関連する秘密鍵PrK-Jおよび公開鍵PuK-Jを有するのに対して、集中型レジストラ404もやはり、関連する秘密鍵PrK-CRおよび公開鍵PuK-CRを有する。ノードID生成関数は、以下のとおりであることが可能である。すなわち、 $\text{ノードID} = \text{hash}(\text{PuK-J} \parallel \text{Rand-J} \parallel \text{Rand-CR})$ 、ただし、PuK-Jは、参加するノードに関する公開鍵であり、Rand-Jは、参加するノードの入力/寄与であり、さらにRand-CRは、集中型レジストラの入力/寄与である。Rand-Jは、コミットメントおよび否認防止のために使用され、ただし、 $\text{hash}(\text{Rand-J})$ には、集中型レジストラによって、集中型レジストラが実際のRand-J値を知る前に、署名が行われる。このようにして、参加するノードは、集中型レジストラが $\text{hash}(\text{Rand-J})$ に署名した後に、Rand-Jを変更することができない。Nonce-JおよびRand-Jが、参加するデバイスによって生成されるのに対して、Nonce-CRおよびRand-CRは、集中型レジストラによって生成される。 $\text{hash}(\text{Rand-J})$ およびRand-CRが、ノード証明書CERT-Jの中に埋め込まれ、ノード証明書CERT-Jには、集中型レジストラの公開鍵PuK-CRを使用した他者による将来の検証のために、集中型レジストラによって、集中型レジストラの秘密鍵PrK-CRを使用して署名が行われる。

20

30

【0042】

ノードIDを取得するのに、参加するノード402は、第1のナンス(Nonce-J)405を生成して、第1のナンス(Nonce-J)を有するID寄与要求406を集中型レジストラ404に送信する。要求される寄与要求406を受信したことに応答して、集中型レジストラ404は、第2のナンス(Nonce-CR)408を生成することができ、受信された第1のナンス(Nonce-J)、および生成された第2のナンス(Nonce-CR)を有するノードIDチャレンジ410に参加するノード402に送信する。一部の実施形態において、ノードIDチャレンジ410は、集中型レジストラ404に関する公開鍵PuK-CRを含有する、集中型レジストラ404に関する公開鍵証明書(CERT-CR)を含むことも可能である。

40

【0043】

次に、参加するノード402が、乱数、値、入力、または寄与(RAND-J)411を生成して、第1のナンスおよび第2のナンス(Nonce-JおよびNonce-CR)、所有の証拠(POP)ペイロード(POP-J)、ランダムな入力/寄与のハッシュ($\text{hash}(\text{Rand-J})$)、参加するノードの公開鍵PuK-Jを含む参加ノード事前証明書(Pre-CERT-J)、ならびにメッセージ署名(SIG-J)を含むIDチャレンジ応答メッセージ412を送信する。Pre-CERT-Jは(オプションとして)、参加するノードによって、例えば、参加するノードの秘密鍵PrK-Jを使用して署名されることが可能である。2つのナンス(Nonce-JおよびNonce-CR)にわたる署名SIG-Jが、SIG-Jが生成された際

50

に参加するノード402がオンラインまたはライブであったことを集中型レジストラ404に証明する。POPペイロード(POP-J)は、オプションであるが、参加するノード402によって、共有鍵に基づいて契約の所有を証明するのに、またはクレジットカード情報を提示するのに使用され得る。共有される秘密の所有を証明するのに、参加するノード402は、共有される秘密を使用して、IDチャレンジ応答412にわたるMACを計算することが可能である。クレジットカード情報を送信するのに、参加するノード402は、集中型レジストラの公開鍵(集中型レジストラ404によって前もって供給されていることが可能な)を使用して、その情報を暗号化することが可能である。また、参加ノード署名(SIG-J)が、メッセージ412の完全性を証明することも可能である。

【0044】

次に、集中型レジストラ404が、チャレンジ応答413を検証する。例えば、集中型レジストラ404は、参加するノードの証明書Pre-CERT-Jおよび/または署名SIG-Jを検証することが可能である。検証413が成功した場合、集中型レジストラ404は、次に、乱数、値、入力、または寄与(Rand-CR)414、ならびにhash(Rand-J)およびRand-CRを含むことが可能であり、さらに集中型レジストラの秘密鍵PrK-CRで署名された参加ノード証明書(CERT-J)415を生成する。第1のナンスおよび第2のナンス(Nonce-JおよびNonce-CR)、hash(Rand-J)および/またはRand-CRにわたる署名された参加ノード証明書(CERT-J)、ならびにメッセージ全体にわたる署名(SIG-CR)を含むID応答メッセージ416が送信される。署名(SIG-CR)は、集中型レジストラ404がオンラインまたはライブであることを証明し、さらに参加するノード402に対してメッセージ416の出所を確立する。次に、参加するノード402が、第1の乱数Rand-Jおよび第2の乱数Rand-CR、ならびにノード402の公開鍵PuK-Jの関数として、ノード402のノードID418を生成する(例えば、ノードID=hash(PuK-J || Rand-J || Rand-CR))。つまり、参加するノード402は、受信された参加ノード証明書CERT-J上で集中型レジストラの公開鍵Puk-CRを使用することによって、集中型レジストラ404からのランダムな入力/寄与Rand-CRにアクセスすることができる。

【0045】

(ノードID検証)

ノードは、ノードIDを取得すると、オーバーレイ・ネットワークに参加することができる。そうするのに、ノードは、そのノードの隣接ノードに1または複数の要求を送信することが可能である。そのような要求は、要求するノードが参加することを許すのに先立って、隣接ノードによって検証される。

【0046】

図5は、参加するノードがオーバーレイ・ネットワークのDHTに参加することを許すのに先立つノードIDの検証を示す図である。参加するノード502が、検証者504の隣接ノードになる要求506を送信する。「検証者」は、オーバーレイ・ネットワークにおける別のノードである。通常、そのような検証は、或るノードが別のノードを相手にDHT隣接関係を形成しようと試みると、トリガされる。要求506の一環として、参加するノードは、第1のナンス(Nonce-J)を送信することが可能である。参加するノードは、そのノードの証明書(CERT-J)も送信することが可能であるが、検証者504は、この段階ではステートレス(stateless)であることを所望する可能性がある。これに応答して、検証者504は、受信された第1のナンス(Nonce-J)に加えて、検証者504自らの第2のナンス(Nonce-V)および検証者証明書(CERT-V)を有するID検証要求510を送信する。次に、参加するノード502が、第1のナンスおよび第2のナンス(Nonce-JおよびNonce-V)、参加するノードのID(Node-ID-J)、対応する参加するノードの証明書(CERT-J)、ならびにメッセージ512全体の署名(SIG-J)を送信する。第1のナンスおよび第2のナンス(Nonce-JおよびNonce-V)が署名(SIG-J)に含まれることは、参加するノード502がライブであることを証明し、さらに署名(SIG-J)自体は、メッセージ512が参加するノード502を発信元とすることを証明する。参加するノードの証明書CERT-Jは、集中型レジストラによって、集中型レジストラの秘密鍵PrK-CRを使用して前もって署名されており、知られている、または検証ノード504によって取得されることが可能な集中型レジストラの公開鍵Puk-CRを使用して検証され得る。証明書CERT-Jは、少なくとも

10

20

30

40

50

も、参加ノード識別子Node ID-Jを生成するのに使用された、集中型レジストラからのランダムな入力Rand-CRを含む。図4における生成されたノードIDを検証する事例において、CERT-Jは、hash(RAND-J)も含む。

【0047】

次に、検証ノード504が、例えば、証明書CERT-Jおよび署名SIG-Jを検証することによって、応答514を検証する。さらに、検証ノード504は、参加ノードID(Node ID-J)を検証することが可能である。そうするのに、検証ノード504は、Rand-JおよびRand-CR(例えば、CERT-Jから取得された)を使用して、参加ノードIDを計算し、この参加ノードIDを、受信されたNode D-Jと比較する。参加するノードの証明書CERT-Jは、検証ノード504が、検証のために集中型レジストラの公開鍵Puk-CRを使用することによってCERT-Jの中のデータ(例えば、Rand-CR)を抽出することができるように、信頼される証明機関(例えば、集中型レジストラ404)によって署名されることに留意されたい。検証が成功した場合(例えば、ローカルで生成されたNode ID-J=受信されたNode ID-J)、検証ノード504は、2つのナンス(Nonce-JおよびNonce-V)、検証者のノードID(Node-ID-V)、対応する検証者証明書(CERT-V)、およびメッセージ516全体の署名(SIG-V)を送信する。この検証プロセスは、相互に行われ得ることに留意されたい。鍵管理交換が検証交換の上に乘せられることが可能である。DH(デフィーヘルマン)暗号交換が、1つのオプションである。別の可能性は、2つのパーティが、他方の公開鍵を使用して保護されて、互いに秘密を送信することであり得る。これらの鍵が混合されて、セッション鍵が生成されてもよい。

【0048】

ノード参加プロセスは、オーバーレイ・ネットワークにおいていくつかのメッセージだけしか生成しない可能性がある(オーバーレイルーティング状態を扱う通常のメッセージング以外に)。ノード参加に関するメッセージングは、ノードID検証だけを目的とし、そのメッセージングは、ルーティング状態を作成するために使用されるメッセージングと潜在的に組み合わされることが可能である。最良ケースシナリオにおいて、この検証方法は、新規ノード参加の時点でオーバーレイルーティングメッセージに加わるメッセージを全く追加しない。この事例において、唯一のオーバーヘッドには、集中型レジストラの署名の検証が含まれる。さらに、集中型レジストラは、ノードIDを配るプロセスをレート制限することによって参加をレート制限することができる可能性がある。集中型レジストラは、任意の1つのエンティティ(ノード)に与えられるノードIDの数を制限するポリシーをさらに使用することが可能である。このことは、ノードID配信を、ネットワークアクセス契約またはクレジットカード番号などの何らかの貴重な情報に結び付けることによって行われることが可能である。

【0049】

しかし、集中型レジストラモデルの1つの弱点は、集中型のエンティティにこのモデルが頼ることである。集中型レジストラは、潜在的な単一の障害ポイントである。さらに、何らかのエンティティが、集中型レジストラを所有して、運用する必要があり、潜在的なノードに広く受入れ可能な単一の管理エンティティを特定することが困難である可能性がある。クレジットカード番号または契約を使用してノードID割当てを制限することは、一般の消費者向けオーバーレイ・ネットワークには余りにも制限的である可能性があり、多くの場合、实际的でない。そうすることは、オーバーレイIDが、他の所属関係(例えば、雇用、組織におけるメンバシップ)に潜在的に結び付けられることが可能であり、さらにすべてのオーバーレイ参加者が同一の、または関連するエンティティに所属しているものと予期される、制限された用途のためなどの、より専用のオーバーレイ・ネットワークを作成するために有用であり得る。集中型レジストラを使用することの別の難点は、集中型レジストラの証明書(CERT-CR)が、オーバーレイ・ネットワークのすべてのノードに事前ロードされなければならないことである。

【0050】

(分散型レジストラ)

第2のオーバーレイ・ネットワークモデルでは、ピアツーピア・オーバーレイ・ネット

10

20

30

40

50

ワークにおけるノードのいくつかは、オーバーレイ・ネットワークに関する分散型レジストラの役割をして、ノードがオーバーレイ・ネットワークにおいてほぼ一様に分布するように、ノードIDを割り当てる責任を負うことが可能である。ノードID割当てプロセスは、集中型レジストラの場合(図3に示される)のプロセスと同様である。

【0051】

図6は、分散型レジストラによるオーバーレイ・ネットワークのDHTに関するノードIDの生成および割当てを示す図である。この例において、参加するノードA602、B604、C606、E610、F612、およびG618が、分散型レジストラ(例えば、ノードD608およびノードH618)の1つと接触して、ノードIDを取得することが可能である。分散型レジストラ(例えば、ノードD608およびノードH618)は、図5に示される検証プロセスを使用することが可能である。

10

【0052】

分散型レジストラモデルを実施する際に少なくとも2つの課題が存在する。第1に、ノードIDを割り当てる分散型レジストラ(例えば、ノードD608およびノードH618)が、オーバーレイ・ネットワークA600におけるノードの様な分布を確実にするように分散型レジストラの間で調整を必要とする。1つのアプローチによれば、各レジストラが、そのレジストラが所有するID空間からノードIDを割り当てる責任を負うように、ID空間がレジストラの間で分割されることが可能である。このアプローチの1つの欠点は、ノードIDが各レジストラのID空間内で一様に分布することが可能である一方で、このアプローチが、ピアツーピア・オーバーレイ・ネットワーク600全体にわたる様な分布を保証しないことである。オーバーレイ・ネットワーク600全体にわたる様な分布は、レジストラ自体の間でノードID割当ての何らかの調整を要求する。

20

【0053】

オーバーレイ・ネットワーク600に少数のこれらの分散型レジストラ608、618が存在する場合、これらのレジストラ608、618が何らかの信頼される機関によって提供されることが可能である。このため、これらのノード(レジストラ608、618)に関して、その他のオーバーレイ・ネットワークノードと比べて、より高い信頼レベルを想定することが実現可能である。

【0054】

集中型レジストラモデルの場合と同様に、ノード参加プロセスは、効率的であり、ノードID生成のための4方向交換、およびノードID検証のための別の4方向交換だけしか関与しない。分散型レジストラとの信頼関係は、検証プロセスがより容易であるように、適度に単純明快にされることが可能である。集中型レジストラモデルとは異なり、分散型レジストラは、オーバーレイ・ネットワークの一部であり、このため、ノード参加プロセスは、参加者/検証者がインターネットに接続されているか否かにかかわらず、行われ得る。

30

【0055】

この分散型レジストラモデルの1つの難点は、このモデルがやはり、他の通常のノードと比べて、より信頼される専用エンティティの存在に依存することである。オーバーレイ・ネットワークのタイプおよび目的に依存して、このことは、実現可能であることも、そうでないこともあり得る。レジストラは、オーバーレイ・ネットワークごとに定義され、展開される必要があり、このことは、かなり高価な運用モードである可能性がある。レジストラ能力を引き受けるように単に任意のノードを使用することは、信頼管理の問題にかかわることであり、完全には実行可能ではない。オーバーレイ・ネットワークに関するDHT内のノードIDの様な分布のため、オーバーレイにおけるノードID割当ての調整が必要とされる。そのような調整は、特に、各分散型レジストラによって扱われる参加要求負荷が非対称的で、オーバーレイ・ネットワークにノードが参加するレートが高い場合、行うのが困難である可能性がある。この分散型レジストラモデルを使用して、各エンティティまたは各ノードに配られるノードIDの数を制限することは、そうすることがレジストラ間の調整を要求するので、かなり困難である。

40

【0056】

代替のアプローチにおいて、すべてのノードは、対称的な信頼関係から始められる。ノ

50

ードがノードの評判を築くにつれ、一部のノードが、オーバーレイ・ネットワークにおいて設定された評判閾値に基づいて、レジストラとして機能するようになる。そのようなレジストラステータスは、そのノードの評判が或る閾値を下回って低下した場合、取り消される可能性がある。さらに、レジストラの評判は、それらのレジストラによって割り当てられたノードIDの検証のために、オーバーレイ・ネットワーク全体にわたって広められる。

【0057】

(分散型ノードID生成)

1つの態様によれば、ノードIDは、よく知られたアルゴリズムを使用して、参加するノード自体によって、ただし、オーバーレイ・ネットワークにおける他の複数のノードからの入力寄与を用いて生成されることが可能である。このモデルでは、すべてのノードが、ID空間管理の点から見て平等であると考えられる。

【0058】

(マルチパーティコミットメントを使用するノードID生成)

図7は、ノードIDが、オーバーレイ・ネットワークにおける他の1または複数のノードからの寄与または支援を用いて、参加するノードによって生成されることが可能な、マルチパーティコミットメントモデルを示す図である。この例において、ノードA702、E710、およびG716が、結託する、または敵対的なノードであることが可能である。ノードC706およびF712は、オーバーレイ・ネットワーク700に参加しようと試みるノードであることが可能である。参加するノードC706が、ランダムな入力 x_C にコミットすることが可能であり、ピアツーピア・オーバーレイ・ネットワーク700におけるノードのランダムなセット(S_C)(例えば、ノードA702、B704、D708、およびE710)からランダムな入力 x_A 、 x_B 、 x_D 、および x_E を求める。このセットSは、オーバーレイ・ネットワーク700における敵対的ノードの最大数より少なくとも1つ多いノードを含む。セット S_C の場合、他のノード(例えば、ノードA、B、D、およびE)からの4つの入力を選択されるのに対して、敵対的ノードの最大数は、3つ(ノードA、E、およびG)である。このことは、残りすべてのノードが、参加するノードCと結託していたとする場合でさえ、ノードID C を計算する際に少なくとも1つの真にランダムな入力が存在することを確実にする。参加するノードCの公開鍵 PK_C が、すべての乱数(x_A 、 x_B 、 x_D 、および x_E)とともに、オーバーレイ・ネットワーク700におけるノードID C を生成するハッシュ関数 h への入力である。同様に、第2のノードF712が、乱数(y_D 、 y_E 、 y_G 、および y_H)を取得するようにノードのランダムなセット(S_F)(例えば、ノードD708、E710、G716、およびH718)を選択すること、および第2のノードF712の公開鍵 PK_F を使用することによって、第2のノードF712のノードID F を計算することが可能である。その後、検証者は、Sの中のノードからのすべての寄与するナンスを検証することによってノードのIDを検証することができる。

【0059】

寄与するノードのセットSは、以下のとおり特徴付けられることが可能である。オーバーレイ・ネットワーク700に割合 f_1 の敵対的ノード(例えば、ノードA、E、およびG)が存在する可能性があるものと想定される。したがって、敵対的ノードの数は、 $f_1 * N$ であり、ただし、Nは、オーバーレイ・ネットワーク700におけるノードの総数である。このことは、寄与するノードのセットSが少なくとも $f_1 * N + 1$ 個のノードを有することを暗示する。

【0060】

ノードIDの検証は、検証するノードが、ノードIDを最初に計算するのに使用されたと考えられる入力を取得し、ノードIDが正しいことを検証するオンラインプロセスであることが可能である。しかし、任意の所与の時点で、ノードのいくつかは、オフラインである可能性がある(例えば、ノードがオフであることが可能である)。平均して、ノードの或る割合 f_2 (すなわち、ノードIDを計算するために入力の数値を供給したノードのうちの)がいつでも同時にオフラインであるものと想定すると、このことは、セットSのサイズが、オンライン検証可能性を確実にするのに少なくとも $(f_1 * N + 1) / f_2$ 個のノードであることを暗示する。

10

20

30

40

50

【0061】

図8は、ノードIDが、オーバーレイ・ネットワークにおける他の1または複数のノードからの寄与または支援を用いて、参加するノードによって生成されることが可能な、マルチパーティコミットメント方法を示す図である。ノード n_J 802が、公開鍵/秘密鍵ペア(Pu_J , Pr_J)806を有する。ノードIDを要求する前に、ノード n_J 802は、ノードIDの中に存在するビット数と少なくとも同数のビットを有する乱数($RAND_J$)を生成する。次に、参加するノード n_J 802は、その乱数のハッシュ $h(RAND_J)$ 、参加ノード証明書 $CERT_J$ を含むとともに、さらに参加要求メッセージがセキュリティのために署名されることが可能であり、署名(SIG_J)も含む参加要求810を紹介ノード n_I 804に送信する(例えば、 $n_J \rightarrow n_I$:参加要求($h(RAND_J)$, $CERT_J$, SIG_J))、ただし、 $CERT_J$ は、参加するノードのデジタル証明書であり、参加するノードの認定された公開鍵 Pu_J を含む)。一例において、参加要求メッセージ810は、参加するノードの秘密鍵 Pr_J を使用して署名されることが可能であり、この署名されたメッセージが受信するノードによって検証され得るように、参加するノードの公開鍵 Pu_J が配信される。そのようなメッセージ署名 SIG_J は、ハッシュされた乱数 $h(RAND_J)$ を覆うことが可能である。

10

【0062】

次に、紹介ノード n_I 804が、オーバーレイ・ネットワークにおける $|S|-1$ 個のランダムに選択されたノード n_{CX} 814にランダム性寄与要求812を送信し、ただし、寄与要求812は、自らの署名 SIG_I を追加した後の参加するノードの要求メッセージ810($h(RAND_J)$, $CERT_J$, SIG_J)を含む(例えば、 $x=\{0, 1, \dots, |S|-1\}$ に関して $n_I \rightarrow n_{CX}$:寄与要求($h(RAND_J)$, $CERT_J$, SIG_J , $\{ID_{CX}\}$, $[CERT_I]$, SIG_I))。紹介ノードの署名 SIG_I は、少なくとも、ハッシュされた乱数 $h(RAND_J)$ および $\{ID_{CX}\}$ を覆わなければならない、ただし、 $\{ID_{CX}\}$ は、 $x=\{0, 1, \dots, |S|-1\}$ に関する寄与するノード n_{CX} に関連するノードIDのリストである。 $CERT_I$ は、紹介ノード n_I の証明書である。オプションとして、この証明書 $CERT_I$ は、寄与するノード n_{CX} が、さもないければ、この証明書 $CERT_I$ を知らない場合、送信されることが可能である(「 $[CERT_I]$ 」によって表される)。この証明書 $CERT_I$ を取得する代替の方法、例えば、X.500ディレクトリサービスのオンライン検索などが、存在することも可能である。

20

【0063】

寄与するノード n_{CX} 814は、元のランダム性寄与(参加するノードからのハッシュされた乱数 $h(RAND_J)$)に加えて、自らのハッシュされたランダム性寄与(すなわち、ハッシュされた乱数 $h(RAND_{CX})$)を有する寄与応答メッセージ816を送信し、個々のメッセージ SIG_{CX} (例えば、 $Y=\{0, 1, \dots, |S|-1\}-\{X\}$ に関して $n_{CX} \rightarrow n_I$ および n_{CY} :寄与応答(ID_{CX} , $h(RAND_J)$, $h(RAND_{CX})$, $[CERT_{CX}]$, SIG_{CX}))、ただし、 ID_{CX} は、寄与するノード n_{CX} のノード識別子であり、 $CERT_{CX}$ は、寄与するノード n_{CX} の証明書である)に署名することが可能である。オプションとして、このノード証明書 $CERT_{CX}$ は、寄与するノード n_{CX} に関する公開鍵を含むことが可能であり、紹介ノード n_I または他の寄与するノード n_{CY} が、さもないければ、このノード証明書 $CERT_{CX}$ を知らない場合、送信されることが可能である($[CERT_{CX}]$ によって表される)。署名 SIG_{CX} は、寄与ノード証明書 $CERT_{CX}$ を除いて、寄与応答全体を覆う。

30

【0064】

すべての寄与応答816が、他のすべての寄与ノード n_{CX} 814および/または紹介ノード n_I 804に送信されることが可能である。代替として、寄与応答816はまず、紹介ノード n_I 804に送信されてもよく、その後、紹介ノード n_I 804が、寄与応答816をすべての寄与ノード n_{CX} 814に中継する(818)。このことは、一部の寄与ノードが、他の寄与ノードに向けられた寄与応答を受信すること、およびそれらの寄与応答を使用して、特定の寄与を生成することを防止する。この時点では、寄与ノード814のいずれも、それらのノード814のランダムな入力 $RAND_{CX}$ を開示しておらず、代わりに、これらのランダムな入力のハッシュ(すなわち、 $h(RAND_{CX})$)を送信していることに留意されたい。このことは、ランダムな入力 $RAND_{CX}$ を実際に開示するのに先立って、すべてのパーティが結び付けられて(すなわち、コミットメント結合検証署名 $V-Sig_{CX}$ によって)、その結果、結託し、さらに/または敵対的なノードが、その後、それらのノードのランダムな入力 $RAND_{CX}$ を変更するのを防止するようにす

40

50

るためである。

【0065】

S個のランダム性寄与(例えば、寄与ノード814からの $|S|-1$ 個、および参加するノード802からの1個)を受信した後、寄与ノード n_{CX} 814のそれぞれが、参加するノードの寄与 $h(RAND_J)$ 、および参加するノードの公開鍵 Pu_J のハッシュを含む、すべてのハッシュされた乱数寄与 $h(RAND_{CX})$ に署名することによって、コミットメント結合検証署名 $V-Sig_{CX}$ を計算し(例えば、各ノード n_{CX} が、マルチパーティコミットメント結合検証署名を以下のとおり計算する。すなわち、 $V-Sig_{CX}=SIG_{CX}(h(Pu_I) \parallel h(RAND_J) \parallel h(RAND_{C0}) \parallel \dots \parallel h(RAND_{C(|S|-1)})$)(820)、ただし、 $SIG_{CX}(\cdot)$ は、寄与するノード n_{CX} の秘密鍵 Pr_{CX} を使用した署名関数である。これらのコミットメント結合検証署名 $V-Sig_{CX}$ は、マルチパーティ(またはマルチノード)検証署名と呼ばれることが可能であり、すべての乱数寄与(異なる寄与ノードからの)が互いに結び付けられることを確実にする。これらのコミットメント結合検証署名 $V-Sig_{CX}$ は、 $|S|-1$ 個すべての寄与ノード814によって供給される、または配信される。次に、すべての寄与ノード814が、それらのノード814の入力値 $RAND_{CX}$ およびコミットメント結合検証署名 $V-Sig_{CX}$ を含む寄与確認を紹介ノード n_I 804に送信する(例えば、 $x=\{0, 1, \dots, |S|-1\}$ に関して $n_{CX} \rightarrow n_I$:寄与確認(ID_{CX} , $RAND_{CX}$, $V-Sig_{CX}$, $[CERT_{CX}]$, SIG_{CX}))(822)。これらのメッセージ822は、他のすべてのメッセージの場合と同様に署名 SIG_{CX} (保護のために)されることが可能である。

【0066】

紹介ノード814が、寄与応答を検証することが可能であり(受信されたランダムな入力/寄与 $RAND_{CX}$ を使用して)(816)、さらに結合署名(すなわち、検証署名 $V-Sig_{CX}$)を検証して(824)、メッセージがプロトコルに準拠していることを確実にする。例えば、紹介ノード804が、受信されたランダムな入力/寄与 $RAND_{CX}$ が、コミットメント結合検証署名 $V-Sig_{CX}$ のそれぞれにおいて使用されているランダムな入力/寄与 $RAND_{CX}$ であることを検証することが可能である。検証が成功した後、紹介ノード804は、乱数 $RAND_{CX}$ 、結合検証署名 $V-Sig_{CX}$ を含む参加応答を参加するノード802に送信することが可能である(例えば、 $n_I \rightarrow n_J$:参加応答($\{ID_{CX}, RAND_{CX}, [CERT_{CX}], V-Sig_{CX}\} | x=\{0, 1, \dots, |S|-1\}, [CERT_I], SIG_I$)、ただし、「 $\{f(x)\} | x=\{0, 1, \dots, |S|-1\}$ 」は、順に $x=\{0, 1, \dots, |S|-1\}$ に関する $f(x)$ の連結を表す)(826)。次に、参加するノード802が、ノード802自らの乱数 $RAND_J$ および寄与された乱数 $RAND_{CX}$ に基づいて、ノード802自らのノードIDを計算する(例えば、 n_J が、ノード $ID_J=h(Pu_I \parallel RAND_J \parallel RAND_{C0} \parallel \dots \parallel RAND_{C(|S|-1)})$ を生成する)(828)。

【0067】

図9は、マルチパーティコミットメントモデルにおけるノードIDの検証を示す図である。検証に先立って、参加するノードが、他の複数のノードが、参加するノードによって、参加するノードのノードIDを生成するのに使用される乱数を寄与する手順(図8と同様の)を経ていることが可能である。参加するノード902が、隣接関係要求906を検証ノード904に送信する(例えば、 $n_J \rightarrow n_V$:隣接関係要求($Node ID_J$, $RAND_J$, $[CERT_J]$, $\{ID_{CX}, RAND_{CX}, [CERT_{CX}], V-Sig_{CX}\} | x=\{0, 1, \dots, |S|-1\}, SIG_J$)、ただし、 $RAND_J$ は、参加するノードからのランダムな入力/寄与であり、 ID_{CX} は、他の寄与するノードを識別し、 $RAND_{CX}$ は、それらの寄与するノードからのランダムな入力/寄与であり、さらに $V-Sig_{CX}$ は、寄与するノードからのコミットメント結合検証署名である)。通常、検証は、参加するノードが別のノード(例えば、検証ノード904)を相手にDHT隣接関係を形成しようと試みると、トリガされる。要求906の一環として、参加するノードは、参加するノードのノード識別子 $Node ID_J$ 、第1の入力/寄与 $RAND_J$ (参加するノードによって生成され、ノードID生成のために使用される)、複数の寄与入力/値 $RAND_{CX}$ 、および寄与ノードからの対応する検証署名 $V-Sig_{CX}$ を送信することが可能である。検証ノード904は、ローカルノード識別子 $Node ID_{LOCAL}=h(Pu_I \parallel RAND_J \parallel RAND_{C0} \parallel \dots \parallel RAND_{C(|S|-1)})$ を計算することが可能であり(908)、さらに、受信されたノード識別子 $Node ID_J$ が、ローカルで計算されたノード識別子 $Node ID_{LOCAL}$ と同一であることを検証する(910)。同様に、検証ノード904は、署名 $V-Sig_{CX}=Sig_{CX}(h(Pu_I) \parallel h(RAND_J) \parallel h(RAND_{C0}) \parallel \dots \parallel h(RAND_{C(|S|-1)}))$ も検証する。受信されたノード識別子 $Node ID_J$

de ID_J が、ローカルで計算されたノード識別子Node ID_{LOCAL} と合致し、さらに受信された検証署名 $V-Sig_{CX}$ が検証に合格することが可能である場合、応答メッセージ912は、隣接関係要求が許可されたことを示し、それ以外の場合、隣接関係要求は拒否される。

【0068】

オプションとして、検証ノード904は、一部の、またはすべての寄与ノード n_{CX} と接触して、それらのノード n_{CX} の存在を検証することも可能である。このことは、寄与ノード識別子 ID_{CX} を使用して、いずれのノードがノードID生成に寄与したかを特定することによって行われることが可能である。

【0069】

表1に、参加するノード n_J 、紹介ノード n_I 、寄与ノード n_{CX} 、および検証ノード n_V の間の、図8および図9のトランザクションを要約する。寄与ノード n_{CX} に関して、 $x=0, 1, \dots, |S|-1$, および S は、選択された寄与ノードのセットであることを明記しておく。さらに、各メッセージは、送信するノードの秘密鍵 Pr_K を使用するメッセージのコンテンツにわたるセキュリティで保護された署名 SIG_K を含むことが可能である。受信するノードは、受信するノードによって知られることが可能な、または取得可能である、送信するノードの公開鍵 Pu_K を使用して、メッセージのコンテンツを検証することができる。さらに、表1において、参加するノード n_J は、ノード n_J の公開鍵 Pu_J (または公開鍵 Pu_J の派生物)を、ノード n_J のランダムな入力 $RAND_J$ (図8および図9で使用される)を使用した。

【0070】

【表 1】

ノード	動作
ノード ID の計算	
n_J	公開鍵/秘密鍵ペア(Pu_J , Pr_J)を生成する
$n_J \rightarrow n_I$	参加要求($h(RAND_J)$, $CERT_J$, SIG_J)、ただし、 $CERT_J$ は、 Pu_J を含む
$n_I \rightarrow nc_x$	寄与要求($h(RAND_J)$, $CERT_J$, SIG_J , $\{ID_{cx}\}$, $[CERT_I]$, SIG_I)
$nc_x \rightarrow n_I, nc_y$	寄与応答(ID_{cx} , $h(RAND_J)$, $h(RAND_{cx})$, $[CERT_{cx}]$, SIG_{cx})、ただし、 nc_y は、他の寄与するノードである
nc_x	検証署名 $V\text{-}Sig_{cx} = SIG_{cx}(h(Pu_J) h(RAND_J) h(RAND_{c0}) \dots h(RAND_{c(S -1)}))$ を計算する
$nc_x \rightarrow n_I$	寄与確認(ID_{cx} , $RAND_{cx}$, $V\text{-}Sig_{cx}$, $[CERT_{cx}]$, SIG_{cx})
$n_I \rightarrow n_J$	参加応答 ($\{ID_{cx}, RAND_{cx}, [CERT_{cx}], V\text{-}Sig_{cx} x=\{0, 1, \dots, S -1\}, [CERT_I], SIG_I\}$)
n_J	ノード $ID_J = h(Pu_J RAND_J RAND_{c0} \dots RAND_{c(S -1)})$ を生成する
ノード ID の検証	
$n_J \rightarrow nv$	隣接関係要求 (ID_J , $RAND_J$, $[CERT_J]$, $\{ID_{cx}, RAND_{cx}, [CERT_{cx}], V\text{-}Sig_{cx} x=\{0, 1, \dots, S -1\}, SIG_J\}$)
nv	ローカルノード識別子 $Node\ ID_{Local} = h(Pu_J RAND_J RAND_{c0} \dots RAND_{c(S -1)})$ を計算し、 $Node\ ID_{LOCAL}$ と $Node\ ID_J$ を比較する
nv	十分な数の署名 $V\text{-}Sig_{cx} = SIG_{cx}(h(h(Pu_J) h(RAND_J) h(RAND_{c0}) \dots h(RAND_{c(S -1)})))$ を検証する
$nv \rightarrow nc_x$	寄与ノード nc_x の存在を確実にする
$nv \rightarrow n_J$	検証成功メッセージ n_J をルーティングテーブルに追加する

【0071】

このマルチパーティコミットメント方法は、完全に非集中型であり、いずれの特別な信頼されるエンティティにも頼らない。このマルチパーティコミットメント方法は、一元的に割り当てられるノードIDの場合と同一レベルのID強度を確実にする。すべてのノードは、信頼の見地から同等に扱われ、このため、この方法は、様々な種類の真のピアツーピア・オーバーレイ・ネットワークを扱う。敵対的ノード(例えば、結託するノード)からの脅威は、ネットワークにおける敵対者の割合が、寄与するノードの数を超えない限り、阻止される。

【0072】

(参加するノードの例)

図10は、ピアツーピア・オーバーレイ・ネットワークに関するノード識別子を生成するために参加するノードにおいて機能する方法の例を示す。参加するノードは、敵対的ノードが結託して、それらのノード自らのノード識別子を選択する際に結託する機会を減らす、または最小限に抑えるようにノード識別子を取得することを求めることが可能である。参加するノードが、第1の寄与 (例えば、参加ノードランダム入力値)を生成することが可能である(1002)。さらに、第1のハッシュ値が、第1の寄与に基づいて生成されることが可能であり、ただし、後の参加要求は、検証署名を計算する際に複数の寄与ノードによ

て使用され得る第1のハッシュ値を含むことが可能である(1004)。

【0073】

参加要求が紹介ノードに送信されることが可能であり、ただし、この参加要求は、オーバーレイ・ネットワークの識別子空間内でノード識別子を生成するためのさらなる1または複数の寄与を取得することを求める(1006)。この参加要求は、第1の寄与の第1のハッシュ値を含むことが可能であり、ただし、第1のハッシュ値は、複数の寄与ノードに供給され、検証署名を計算する際に使用される。紹介ノードは、オーバーレイ・ネットワークにおけるランダムに、または擬似ランダムに選択されたノードであることが可能であり、あるいは紹介ノードは、オーバーレイ・ネットワーク内の割り当てられた、または信頼されるノードであることが可能である。参加要求に応答して、参加するノードは、紹介ノードから複数の寄与(例えば、ランダムな入力値)を受信することが可能であり、この複数の寄与は、先行する外部マルチノードコミットメント動作によって互いに結び付けられるとともに、第1の寄与にも結び付けられる(1008)。一部の例において、参加するノードは、紹介ノードから複数の検証署名を受信することも可能であり、各検証署名は、複数の寄与ノードによって生成された複数のコミットメント値の関数であり、これらのコミットメント値は、寄与ノードによって供給される寄与の関数である(1010)。したがって、各検証署名が、各寄与ノードがそのノードの寄与を実際に開示するのに先立って、これらのコミットメント値と一緒に結び付け、その結果、寄与と一緒に結び付ける。これらの寄与は、ランダム値または擬似ランダム値であることが可能であることに留意されたい。次に、ノード識別子が、第1の寄与および受信された寄与の関数として生成されることが可能である(1012)。

【0074】

次に、参加するノードは、オーバーレイ・ネットワークにおける1または複数の隣接ノードにノード識別子を配信することが可能であり、ノード識別子の有効性は、その1または複数の隣接ノードによって独立に検証され得る(1014)。つまり、参加するノードは、オーバーレイ・ネットワーク内のルーティングの目的で隣接ノードによって維持される分散ハッシュテーブル(DHT)に追加されることを求めることが可能である。DHTは、非集中型であり、オーバーレイ・ネットワークにおける各ノードは、オーバーレイ・ネットワークにおけるノード識別子の一部分またはサブセットだけを維持する。各ノードは、DHTの中に一意のノード識別子を有し、同様に、ネットワークに格納された各サービスまたは各オブジェクトも、一意の識別子またはIDを有する。

【0075】

一例において、参加するノードは、参加するノードのメッセージングをセキュリティで保護するのに使用されることが可能な非対称公開鍵-秘密鍵ペアを有することも可能である。

【0076】

図11は、オーバーレイ・ネットワークのノードとして動作することを求めていることが可能なデバイスの例を示すブロック図である。ノードデバイス1102が、ネットワーク通信インタフェース1106に結合された処理回路1104と、(オプションで)ストレージデバイス1108とを含むことが可能である。処理回路1104は、図10に示される方法を実行するように構成されたノード識別子ジェネレータ1110を含むことが可能である。つまり、ノード識別子ジェネレータ1110は、ネットワークにおける他のノードから(ネットワーク通信インタフェース1106を介して)、ジェネレータ1110が、その後、自らのノード識別子を生成するのに使用する、1または複数の寄与または入力値を要求するように構成されることが可能である。次に、このノード識別子が、ネットワーク通信インタフェース1106を介してオーバーレイ・ネットワークにおける隣接ノードに供給されることが可能である。一部の実施形態において、ストレージデバイス1108は、ノードデバイス1102が、デバイス1102の隣接ノードの1つまたは複数に関するルーティング(例えば、ノード識別子)情報を維持することが可能な分散ルーティングテーブル1114を含むことが可能である。そのような隣接ノードは、それらのノードの識別子の値が、ノードデバイス1102のノード識別子に近い(必ずし

も物理的に近いわけではない)という意味で「隣接している」。

【0077】

図12は、参加するノードの機能構成要素の例を示すブロック図である。参加するノード1202が、ノード1202が用いてオーバーレイ・ネットワークに参加することができる、ノード1202自らのノード識別子を生成することを求めていることが可能である。参加するノード1202は、ランダムな、または擬似ランダムな寄与、入力、または値 $RAND_J$ 1205を生成することが可能な寄与ジェネレータ1204を含むことが可能である。ハッシュジェネレータ1206が、ナンスまたは値 $RAND_J$ 1205にハッシュ関数を適用して、ハッシュされた値 $h(RAND_J)$ 1207を生成することが可能である。次に、参加要求モジュール1208が、ハッシュされた値 $h(RAND_J)$ 1207を含む入力要求を、オーバーレイ・ネットワーク1210における紹介ノードに送信することが可能である。寄与収集モジュール1214が、オーバーレイ・ネットワークの複数のノードから、(要求に応答して)1または複数の寄与、入力、または値 $RAND_{CX}$ 1213および/あるいは1または複数の対応するコミットメント結合検証署名 $V-Sig_{CX}$ 1215を受信することが可能である(1212)。次に、その1または複数の寄与 $RAND_{CX}$ 1213および寄与 $RAND_J$ 1205が、参加するデバイス1202が使用することができるノード識別子(ノードID)1217を生成するノード識別子ジェネレータ1216への入力の役割をする。次に、ノード識別子配信モジュール1218が、オーバーレイ・ネットワークにおける1または複数の隣接ノード1220にノード識別子ノードID1217を送信して、参加するノードが、それらのノード1220の分散ルーティングテーブルに追加され得るようにすることが可能である。隣接ノードは、ノード識別子1217に値が近いノード識別子を有するノードであることが可能である。

【0078】

(紹介ノードの例)

図13は、参加するノードがピアツーピア・オーバーレイ・ネットワークにおいてノード識別子を取得するのを円滑にするように紹介ノードにおいて機能する方法の例を示す。オーバーレイ・ネットワークに関する識別子空間内で新たなノード識別子を生成するための1または複数の寄与を取得することを求める参加するノードからの参加要求が受信されることが可能である(1302)。参加要求は、第1の寄与の第1のハッシュ値を含むことが可能であり、ただし、第1のハッシュ値は、その後、寄与ノードに供給されて、検証署名を計算する際に使用されることが可能である。紹介ノードが、オーバーレイ・ネットワークにおける複数の寄与ノードに複数の寄与要求を送信することが可能である(1304)。一例において、寄与要求が送信される寄与ノードの数は、オーバーレイ・ネットワークにおける(予期される、可能な、または予想される)敵対的ノードの数より少なくとも1つ多い。さらに、寄与要求が送信される寄与ノードは、ランダムに、または擬似ランダムに選択されることが可能である。

【0079】

複数のコミットメント値が、寄与ノードから受信されることが可能であり、各コミットメント値は、寄与ノードのそれぞれにおいて生成される対応する寄与の関数である(1306)。次に、複数のコミットメント値が、寄与ノードに配信されることが可能である(1308)。一実施形態において、この複数のコミットメント値は、寄与ノードによって他の寄与ノードに直接に配信されることが可能である。しかし、これらのコミットメント値を紹介ノードに最初に送信することにより、いずれの寄与ノードが別の寄与ノードからコミットメント値を受信するのにも先立って、すべてのコミットメント値が収集されて、これにより、敵対的な寄与ノードが、そのノード自らのコミットメント値を変更して特定のノード識別子を実現することが防止され得る。

【0080】

複数のコミットメント値を配信することに応答して、複数の寄与(例えば、入力値)、および対応する検証署名が受信されることが可能であり、各検証署名は、複数のコミットメント値の関数である(1310)。紹介ノードは、寄与ノードのそれぞれから検証署名を受信することが可能であり、各検証署名は、複数のコミットメント値の関数であることに留意されたい。各検証署名は、各寄与ノードがそのノードの寄与を実際に開示するのに先立って

、コミットメント値と一緒に結び付ける(その結果、寄与と一緒に結び付ける)ことが可能である。さらに、各検証署名は、参加するノードからの第1の値(例えば、参加するノード寄与のハッシュ)の関数であることも可能である。

【0081】

紹介ノードが、参加するノードに寄与を送信するのに先立ってコミットメント値および検証署名を検証することが可能である(1312)。次に、その複数の寄与が参加するノードに送信されることが可能であり、ただし、ノード識別子が、参加するノードによって複数の寄与の関数として生成され(1314)、さらに、場合により、参加するノード自らの寄与および/または公開鍵の関数として生成される。これらの寄与は、ランダム値または擬似ランダム値であることが可能である。さらに、検証署名が参加するノードに送信されることも可能である(1316)。

10

【0082】

図14は、参加するノードによるノード識別子の生成を円滑にするようにオーバーレイ・ネットワークの紹介ノードとして動作することが可能なデバイスの例を示すブロック図である。ノードデバイス1402が、ネットワーク通信インタフェース1406に結合された処理回路1404と、(オプションで)ストレージデバイス1408とを含むことが可能である。処理回路1404は、図13の方法に示されるとおり、別のノードがそのノード自らのノード識別子を生成するのを支援するように構成されたノード参加要求モジュール1410を含むことが可能である。つまり、ノード参加要求モジュール1410が、ノード識別子を生成するのに寄与を取得することの支援のために、或るノードから(ネットワーク通信インタフェース1406を介して)要求を受信することが可能である。ノード参加要求モジュール1410は、オーバーレイ・ネットワークにおける1または複数の寄与ノードにコミットメント要求を送信し(ネットワーク通信インタフェース1406を介して)、それらの寄与ノードから1または複数の寄与を受信するように構成されることが可能である。次に、ノード参加要求モジュール1410は、その1または複数の寄与を、要求するノードに送る。一部の実施形態において、ストレージデバイス1408が、ノードデバイス1402が、ノードデバイス1402の隣接ノードの1つまたは複数に関するルーティング(例えば、ノード識別子)情報を維持することが可能な分散ルーティングテーブル1414を含むことが可能である。

20

【0083】

図15は、紹介ノードの機能構成要素の例を示すブロック図である。紹介ノード1502は、参加するノード1504から参加要求メッセージ1505を受信する参加要求レシーバ1506を含むことが可能である。参加要求メッセージ1505は、参加するノードからのナンス $RAND_J$ のハッシュ値 $h(RAND_J)$ を含むことが可能である。寄与要求モジュール1508が、オーバーレイ・ネットワークにおける1または複数の寄与ノード1510に1または複数の要求1509を送信することが可能である。寄与ノードのそれぞれが、そのノードが行うことを意図する実際の寄与 $RAND_{CX}$ の関数であることが可能なコミットメント値 $h(RAND_{CX})$ 1512を供給することが可能である。コミットメント値1512は、この時点で実際の寄与が開示されないように、実際の寄与のハッシュであることが可能である。コミットメント配信モジュール1516が、受信されたコミットメント値をすべての寄与ノードに配信することが可能である(1518)。この時点で、寄与ノードが、それらのコミットメント値 $h(RAND_{CX})$ を、検証署名(例えば、 $x=0..m$ に関して $V-Sig_k=hash(h(RAND_{CX}))$)によって互いに結び付けることができ、その結果、まだ開示されていない寄与 $RAND_{CX}$ を結び付ける。その他の寄与ノードに関するコミットメント値 $h(RAND_{CX})$ を送信したことに応答して、寄与収集モジュール1520が、寄与 $RAND_{CX}$ 1519を受信することが可能であり、さらに、場合により、各寄与ノードから検証署名 $V-Sig(h(RAND_{CX}))$ 1521を受信することが可能である。寄与を参加するノードに送信するのに先立って、検証モジュール1523が、コミットメント値および検証署名を検証していることが可能である。次に、寄与配信モジュール1524が、受信された寄与 $RAND_{CX}$ とともに検証署名 $V-Sig$ も参加するノード1504に送信することが可能である。

30

40

【0084】

(寄与ノードの例)

50

図16は、参加するノードがピアツーピア・オーバーレイ・ネットワークにおいてノード識別子を取得するのを円滑にするように寄与ノードにおいて機能する方法の例を示す。寄与を求める寄与要求が受信されることが可能であり、ただし、この寄与要求は、参加するノードからのハッシュされた初期寄与を含む(1602)。寄与ノードが、第1の寄与を取得することが可能であり(1604)、第1の寄与からコミットメント値を生成する(1606)。次に、寄与ノードが、そのコミットメント値を少なくとも1つの別の寄与ノードに送信する(1608)。次に、他のコミットメント値が、その少なくとも1つの別の寄与ノードから受信されることが可能である(1610)。検証署名が、受信されたコミットメント値、およびハッシュされた初期寄与に基づいて生成されることが可能である(1612)。その他のコミットメント値を受信した後、第1の寄与、および検証署名が参加するノードに送信されることが可能であり、ただし、ノード識別子は、少なくとも第1の寄与、および初期寄与の関数として、参加するノードによって生成される(1614)。第1の寄与は、ランダム値または擬似ランダム値のいずれかであることが可能である。検証署名は、参加するノードからのコミットメント値およびハッシュされた初期寄与を結び付けて、他のノードが、その後、ノード識別子の真正性を検証することを許す。一実施形態において、コミットメント値はまず、コミットメント値を収集して、後に、コミットメント値を寄与ノードに配信する紹介ノードに送信されることが可能である。

【0085】

図17は、参加するノードによるノード識別子の生成を円滑にするようにオーバーレイ・ネットワークにおける寄与ノードとして動作することが可能なデバイスの例を示すブロック図である。ノードデバイス1702が、ネットワーク通信インタフェース1706に結合された処理回路1704と、(オプションで)ストレージデバイス1708とを含むことが可能である。処理回路1704は、図16の方法で示されるとおり、別のノードがそのノード自らのノード識別子を生成するのを支援するように構成された寄与ジェネレータ1710を含むことが可能である。つまり、寄与ジェネレータ1710が、ノード識別子を生成するのに寄与を取得するために、或るノードから(ネットワーク通信インタフェース1706を介して)寄与要求を受信することが可能である。寄与ジェネレータ1710は、第1の寄与に基づいてコミットメント値を計算するように構成されることが可能である。次に、コミットメント値は、オーバーレイ・ネットワークにおける他の1または複数の寄与ノードに送信される(ネットワーク通信インタフェース1706を介して)。次に、寄与ノード1702が、他の寄与ノードから1または複数のコミットメント値を受信することが可能である。次に、これらのコミットメント値が組み合わせられて、第1の寄与と一緒に、参加するノードに送信される検証署名が生成される。

【0086】

図18は、寄与ノードの機能構成要素の例を示すブロック図である。寄与ノード1802は、寄与を求める寄与要求1805を受信するための寄与要求レシーバ1806を含むことが可能であり、寄与要求は、参加するノード、または紹介ノード1804からのハッシュされた初期寄与 $h(RAND_J)$ を含む。次に、ナンバジェネレータ1808(例えば、寄与ジェネレータ)が、第1の寄与 $RAND_C$ 1809を取得する。次に、コミットメントジェネレータ1814が、第1の寄与 $RAND_C$ からコミットメント値 $h(RAND_C)$ を生成する。コミットメントディストリビュータ1816が、コミットメント値 $h(RAND_C)$ を少なくとも1つの別の寄与ノードに送信することが可能である(1818)。寄与収集モジュール1820が、その少なくとも1つの別の寄与ノードから他のコミットメント値 $h(RAND_{CX})$ を受信することが可能である。検証署名ジェネレータ1822が、受信されたコミットメント値 $h(RAND_{CX})$ 、およびハッシュされたナンス $h(RAND_J)$ に基づいて検証署名 $V-Sig_C$ を生成することが可能である。その他のコミットメント値 $h(RAND_{CX})$ を受信した後、寄与配信モジュール1824が、第1の寄与 $RAND_C$ および検証署名 $V-Sig_C$ を参加するノード1804に送信することが可能であり、ただし、ノード識別子は、参加するノードによって、少なくとも第1の寄与および初期寄与の関数として生成される。その結果、検証署名は、コミットメント値、およびハッシュされたナンスを結び付けて、その他のノードが、その後、そのノード識別子の真正性を検証することを許す。

【0087】

(マルチパーティコミットメントを使用するスケーラブルなノードID生成)

別の態様によれば、図8および図9のマルチパーティコミットメント方法が、ノードID生成のためにランダムな入力を取得するように接触される寄与するノードの数を減らすように試みることによって、さらに最適化されることが可能である。マルチパーティコミットメント方法の相当な複雑さは、寄与ノードの数が、ネットワークにおける敵対的ノードの数より少なくとも1つ多くなければならないという事実由来する。適度に大きい数のオーバーレイノードを有するシステムにおいて、このことは、ノードが参加する時点で許容できない数のメッセージおよび署名をもたらす。ノード参加のレートが高い場合、このことは、多量の望ましくないオーバーレイ・ネットワークトラフィックにつながる可能性がある。また、実際のシステムにおける敵対的ノードの数を正確に推定するのが難しいという事実の他に、このことは、超えると、このスキームが真にランダムなIDを生成すると保証され得ない、そのそれほど正確でない数に制限を課す。

10

【0088】

ノードID生成のためにランダムな入力を供給するためのランダムな寄与ノードが選択され得ることを確実にすることが可能である場合、そのような寄与ノードの数は、いくつかだけに減らされることが可能である。このことは、高い確率で、入力が、システムにおける結託するノードによって供給されたのではないことを確実にする。前述のマルチパーティコミットメント方法を最適化するためのいくつかのアプローチが可能である。1つのアプローチは、寄与ノードの役割をするノードIDのセットを提供するために、参加するノードの公開鍵のハッシュチェーン、あるいはシードとして参加するノードの公開鍵を用いた事前定義された擬似ランダム生成関数を使用することである。別のアプローチは、そのようなハッシュチェーンまたは擬似ランダム関数からもたらされたノードIDを、参加ノードIDアルゴリズムの直接入力として使用して、メッセージの数を少なく保つことである。これらの事例の両方において、参加するノードが、好都合なノードIDに到達する様々な公開鍵を作成することができることから、選択ロケーション攻撃からは、効果的に保護されない。

20

【0089】

別のアプローチは、紹介ノードの公開鍵 Pu_I を使用して、寄与ノードを見出すことを規定する。紹介ノードの公開鍵に基づいて、寄与ノードをランダムに選択するいくつかの方法が存在する。例えば、値 x が、 $x=h(Pu_I \parallel \text{「文字列」})$ として計算されることが可能であり、ただし、 Pu_I は、紹介ノードの公開鍵であり、「文字列」は、固定の値、またはゆっくり変化する値であることが可能である。値 x は、オーバーレイ・ネットワークにおけるそのポイント(すなわち、値 x)の責任を負うノード n_c を見出すのに使用される。次に、ノード n_c を使用して、それぞれ、オーバーレイ・ネットワークにおけるノード n_c の先行ノードおよび後続ノードである、ノード n_{c-pre} およびノード n_{c-next} が特定される。これらのノード n_{c-prev} および n_{c-next} は、参加するノードの新たなノード識別子を生成するためのランダム性寄与者として選択される。マルチパーティランダム性寄与は、これらのノードを寄与者として実行される。

30

【0090】

別の変形形態において、寄与ノードは、 $x=h(Pu_I \parallel \text{「文字列-1」})$ 、 $y=h(Pu_I \parallel \text{「文字列-2」})$ 、および $z=h(Pu_I \parallel \text{「文字列-3」})$ となるように値 x を計算することによって選択され、ただし、「文字列-1」、「文字列-2」、および「文字列-3」は、固定の値、またはゆっくり変化する値であることが可能である。値 x 、 y 、および z は、オーバーレイ・ネットワークにおける対応するポイント(すなわち、値 x 、 y 、および z)の責任を負う複数のノード n_x 、 n_y 、および n_z を見出すのに使用されることが可能である。オーバーレイ・ネットワークにおける計算されたポイント(すなわち、値 x 、 y 、および z)の責任を負うノード n_x 、 n_y 、および n_z が特定されると、これらのノード n_x 、 n_y 、および n_z は、参加するノードへの入力値を供給する寄与ノードとして使用されることが可能である。

40

【0091】

50

隣接ノードとして追加されるように参加するノードから要求(新たなノード識別子を有する)を受信すると、隣接ノードは、その新たなノード識別子を独立に検証することができなければならないことに留意されたい。これを行うのに、隣接ノードは、検証者として、図9に示されるとおり情報を取得する。

【0092】

これらのスケーリングされたマルチノードコミットメントは、完全に非集中型であり、特別なエンティティを全く要求しない。これらのスケーリングされたマルチノードコミットメントは、一元的に割り当てられるIDの場合と同一レベルのID強度を確実にする。すべてのノードは、信頼の見地から同等に扱われ、このため、これらのスケーリングされた方法は、様々な種類の真のピアツーピア・オーバーレイ・ネットワークを扱う。このソリューションにおいて参加を扱うのに必要とされるメッセージの追加の数は、オーバーレイにおけるノードの総数にかかわらず、小さく、一定である。また、必要とされる暗号動作の追加の数も、オーバーレイにおけるノードの総数にかかわらず、小さい数に限定され、一定である。

10

【0093】

概して、本開示で説明される処理の大部分は、同様の仕方で実施されることが可能であることを認識されたい。回路または回路セクションのいずれも、単独で実施されても、集積回路の一部として1または複数のプロセッサと組合せて実施されてもよい。それらの回路の1つまたは複数の、集積回路、アドバンスRISCマシン(ARM)プロセッサ、デジタルシグナルプロセッサ(DSP)、汎用プロセッサなどの上に実装されることが可能である。

20

【0094】

また、実施形態は、フローチャート、流れ図、構造図、またはブロック図として表されるプロセスとして説明されることが可能であることにも留意されたい。フローチャートは、動作を順次プロセスとして説明する可能性があるものの、動作の多くは、並行に、または同時に実行され得る。さらに、動作の順序は、並べ替えられることが可能である。プロセスは、プロセスの動作が完了すると、終了させられる。プロセスは、メソッド、関数、プロシージャ、サブルーチン、およびサブプログラムなどに対応することが可能である。プロセスが関数に対応する場合、プロセスの終了は、呼び出し元の関数またはメイン関数への関数のリターンに対応する。

【0095】

本出願で使用される「構成要素」、「モジュール」、「システム」などの用語は、ハードウェア、ファームウェア、ハードウェアとソフトウェアの組合せ、ソフトウェア、または実行中のソフトウェアである、コンピュータ関連エンティティを指すことを意図している。例えば、構成要素は、プロセッサ上で実行されているプロセス、プロセッサ、オブジェクト、実行可能ファイル、実行のスレッド、プログラム、および/またはコンピュータであることが可能であるが、以上には限定されない。例として、コンピューティングデバイス上で実行されているアプリケーションと、そのコンピューティングデバイスがともに構成要素であり得る。1または複数の構成要素が、プロセス内、および/または実行のスレッド内に存在することが可能であり、構成要素は、1つのコンピュータ上に局所化され、さらに/または2つ以上のコンピュータの間に分散されることが可能である。さらに、これらの構成要素は、様々なデータ構造を格納している様々なコンピュータ可読媒体から実行され得る。これらの構成要素は、1または複数のデータパケットを有する信号に従うなどして、ローカルプロセスおよび/または遠隔プロセスを介して通信することが可能である(例えば、1つの構成要素からのデータが、信号によって、ローカルシステムにおいて、分散システムにおいて、さらに/または他のシステムを相手にするインターネットなどのネットワークを介して、別の構成要素と対話して)。

30

40

【0096】

さらに、記憶媒体は、読み取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、磁気ディスク記憶媒体、光記憶媒体、フラッシュメモリデバイス、および/または情報を格納するための他のマシン可読媒体を含め、データを格納するための1または複数のデバイス

50

を表すことが可能である。「マシン可読媒体」という用語には、命令および/またはデータを格納すること、含むこと、または伝送することができるポータブルまたは固定の記憶媒体、光ストレージデバイス、無線チャネル、および他の様々な媒体が含まれるが、以上には限定されない。

【0097】

さらに、実施形態は、ハードウェア、ソフトウェア、ファームウェア、ミドルウェア、マイクロコード、または以上の任意の組合せによって実施されることが可能である。ソフトウェア、ファームウェア、ミドルウェア、またはマイクロコードで実施される場合、必要なタスクを実行するプログラムコードまたはコードセグメントは、記憶媒体または他のストレージなどのマシン可読媒体の中に格納されることが可能である。プロセッサが、必要なタスクを実行することが可能である。コードセグメントは、プロシージャ、関数、サブプログラム、プログラム、ルーチン、サブルーチン、モジュール、ソフトウェアパッケージ、クラス、あるいは命令、データ構造、またはプログラムステートメントの任意の組合せを表すことが可能である。コードセグメントは、情報、データ、引数、パラメータ、またはメモリ内容を送ること、および/または受け取ることによって、別のコードセグメントまたはハードウェア回路に結合されることが可能である。情報、引数、パラメータ、データなどは、メモリ共有、メッセージ受渡し、トークン受渡し、ネットワーク伝送などを含む任意の適切な手段を介して送られる、転送される、または伝送されることが可能である。

【0098】

図1、図2、図3、図4、図5、図6、図7、図8、図9、図10、図11、図12、図13、図14、図15、図16、図17、および/または図18に示される構成要素、ステップ、および/または機能の1つまたは複数は、並べ替えられる、さらに/または組み合わせられて、単一の構成要素、ステップ、もしくは機能にされる、あるいはいくつかの構成要素、ステップ、または機能において実現されることが可能である。また、さらなる要素、構成要素、ステップ、および/または機能が、本発明を逸脱することなく、追加されることも可能である。図11、図12、図14、図15、図17、および/または図18に示される装置、デバイス、および/または構成要素は、図1、図2、図3、図4、図5、図6、図7、図8、図9、図10、図13、および/または図16で説明される方法、特徴、またはステップの1つまたは複数を実行するように構成されることが可能である。本明細書で説明される新規なアルゴリズムは、ソフトウェアおよび/または組み込まれたハードウェアにおいて効率的に実施されることが可能である。

【0099】

本明細書で開示される実施形態に関連して説明される様々な、例示的な論理ブロック、モジュール、回路、およびアルゴリズムステップは、電子ハードウェアとして、コンピュータソフトウェアとして、または電子ハードウェアとソフトウェアの組合せとして実施されることが可能であることを当業者はさらに認識されよう。ハードウェアとソフトウェアの、この互換性を明確に示すのに、様々な、例示的な構成要素、ブロック、モジュール、回路、およびステップが、概ね、機能の点で上記で説明されてきた。そのような機能がハードウェアとして実施されるか、ソフトウェアとして実施されるかは、全体的なシステムに課せられる特定の応用上の制約、および設計上の制約に依存する。

【0100】

本明細書で説明される本発明の様々な特徴は、本発明を逸脱することなく、様々なシステムにおいて実施され得る。例えば、本発明の一部の実施形態は、移動している通信デバイスまたは静止の通信デバイス(例えば、アクセス端末装置)、および複数の移動基地局または静止の基地局(例えば、アクセスポイント)で実行されることが可能である。

【0101】

以上の実施形態は、単に実施例に過ぎず、本発明を限定するものと解釈されるべきではないことに留意されたい。これらの実施形態の説明は、特許請求の範囲を例示することを意図しており、限定することは意図していない。このため、本教示は、他のタイプの装置に容易に適用されることが可能であり、多くの代替形態、変更形態、および変形形態が、

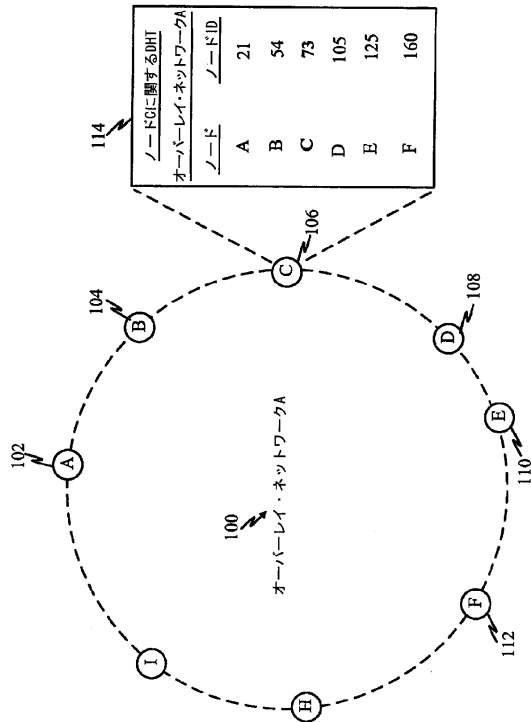
当業者には明白となろう。

【符号の説明】

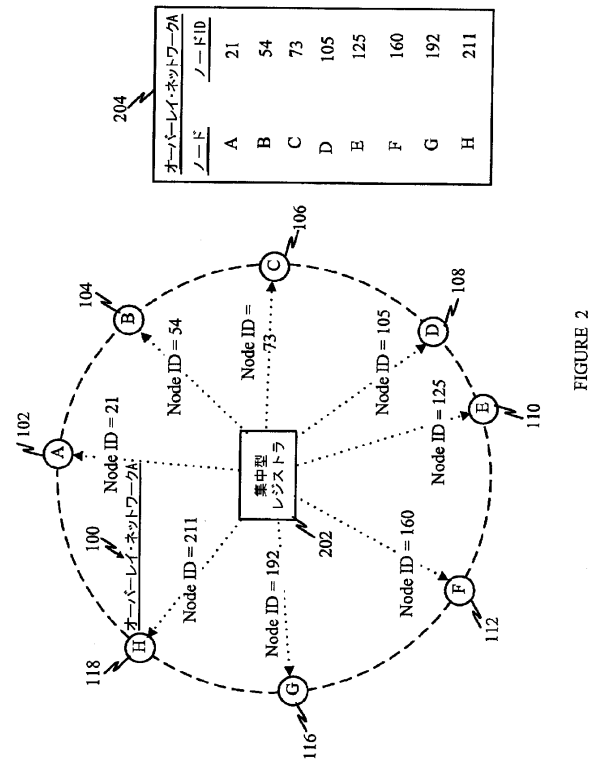
【0102】

100、600、700	オーバーレイ・ネットワーク	
102、104、106、108、110、112、116、118、402、404、502、504、602、604、606、608、610、612、616、618、702、704、706、708、710、712、716、718、802、804、814、902、904、1102、1202、1402、1502、1702、1802	ノード	
114、614、1114、1414、1714	分散ルーティングテーブル	
202	集中型レジストラ	
1104、1404、1704	処理回路	10
1106、1406、1706	ネットワーク通信インタフェース	
1108、1408、1708	ストレージデバイス	
1110、1216	ノード識別子ジェネレータ	
1204	ナンスジェネレータ	
1206	ハッシュジェネレータ	
1208、1410	参加要求モジュール	
1214、1520、1820	寄与収集モジュール	
1217	ノード識別子	
1218	ノード識別子配信モジュール	
1506	参加要求レシーバ	20
1508	寄与要求モジュール	
1514	コミットメントレシーバ	
1516、1816	コミットメントディストリビュータ	
1523	検証モジュール	
1524、1824	寄与配信モジュール	
1710	寄与ジェネレータ	
1806	寄与要求レシーバ	
1808	ナンバジェネレータ	
1814	コミットメントジェネレータ	
1822	検証署名ジェネレータ	30

【図 1】



【図 2】



【図 3】

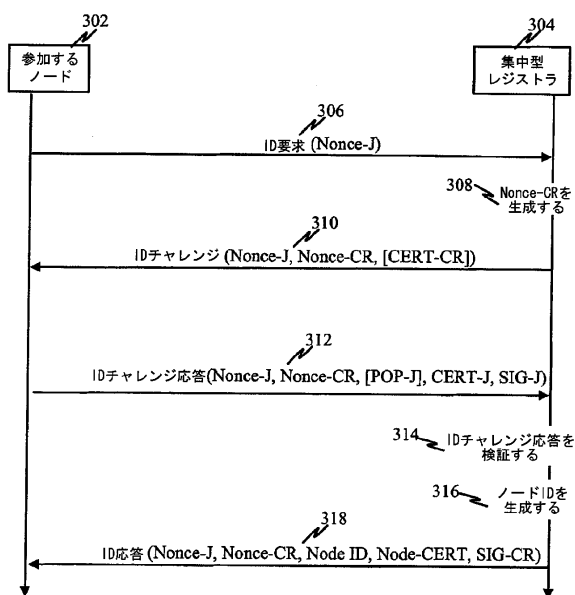


FIGURE 3

【図 4】

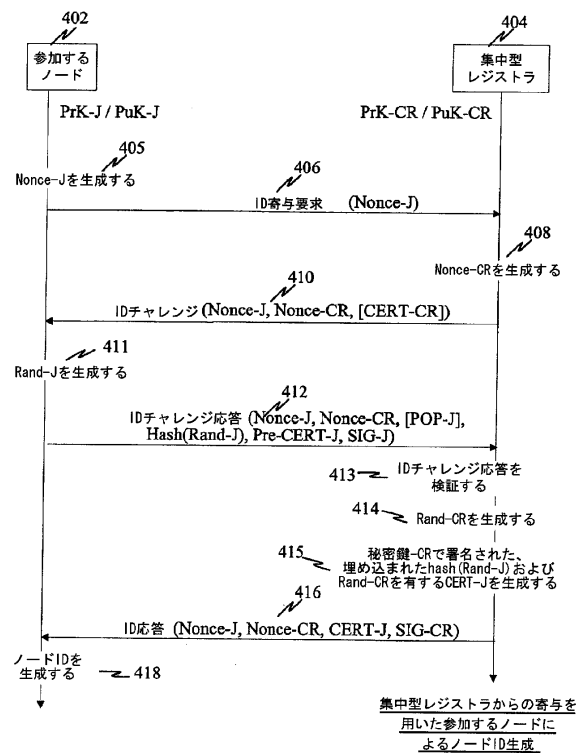


FIGURE 4

【図 5】

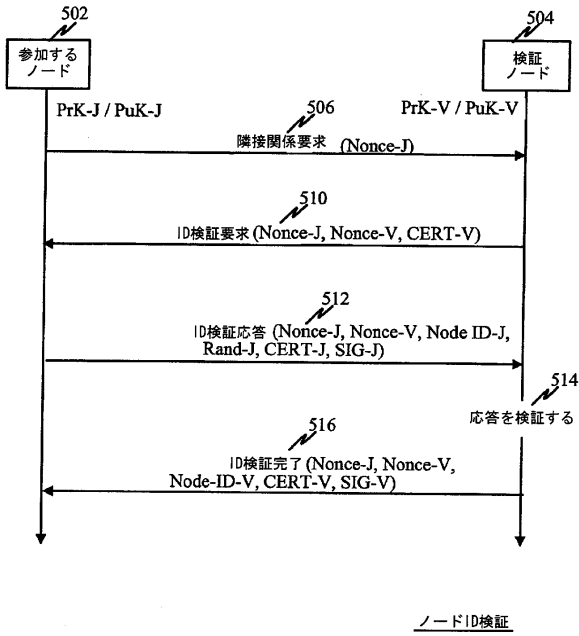


FIGURE 5

【図 6】

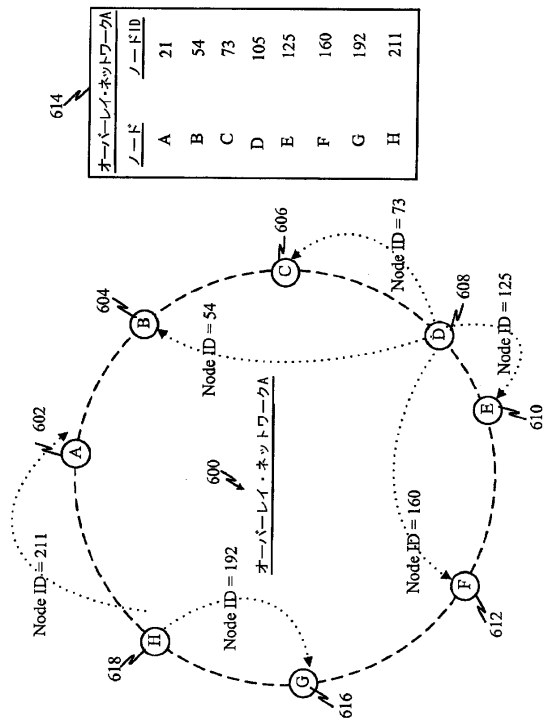


FIGURE 6

【図 7】

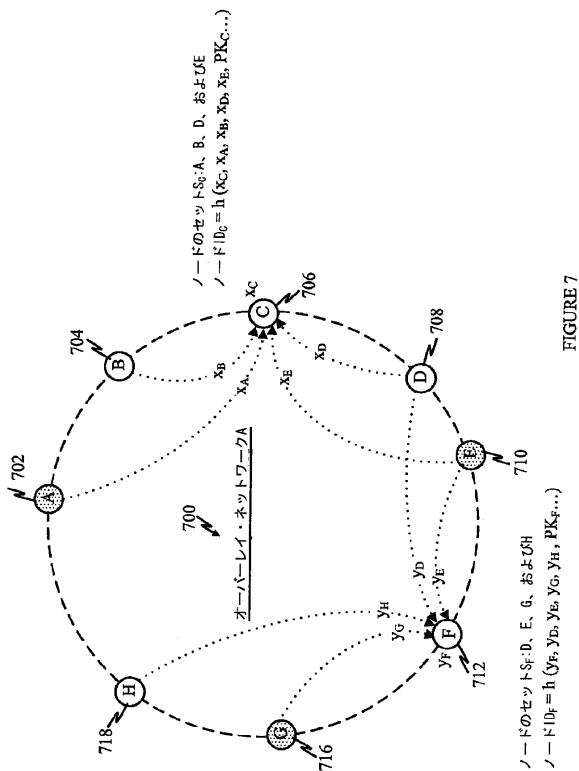


FIGURE 7

【図 8】

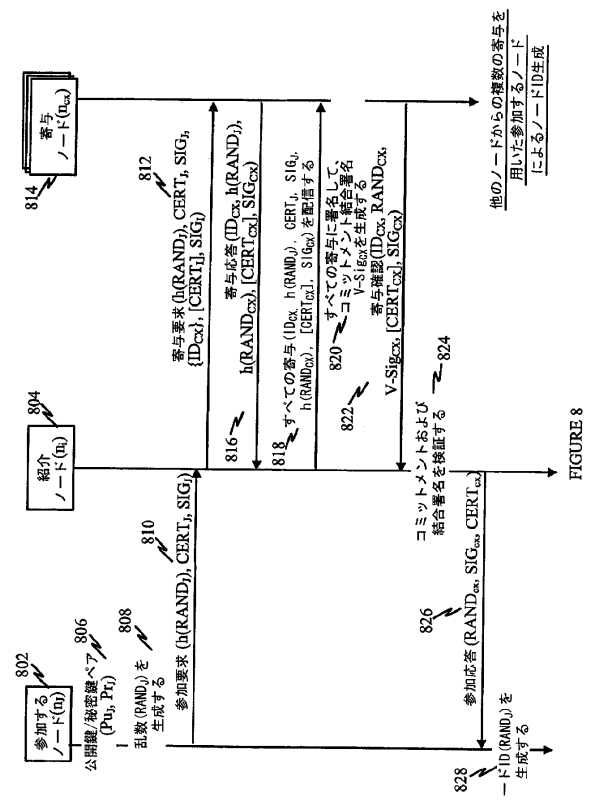


FIGURE 8

【図 9】

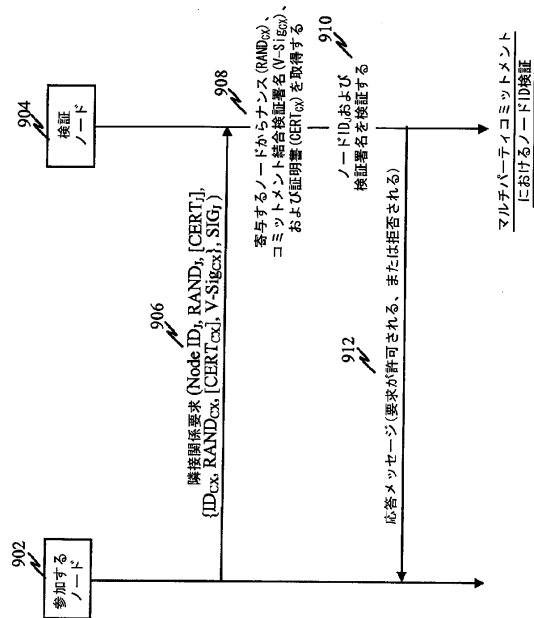


FIGURE 9

【図 10】

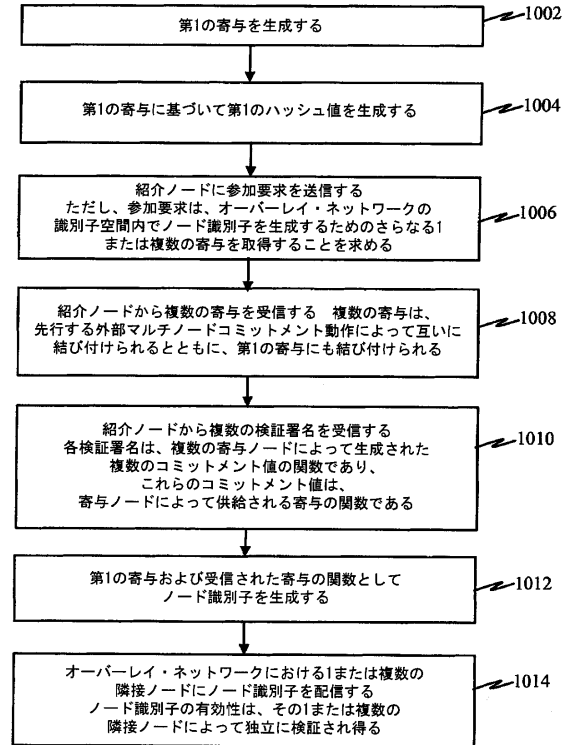


FIGURE 10

【図 11】

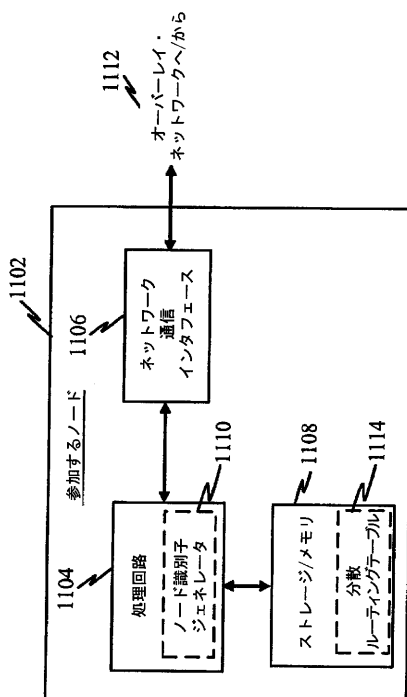


FIGURE 11

【図 12】

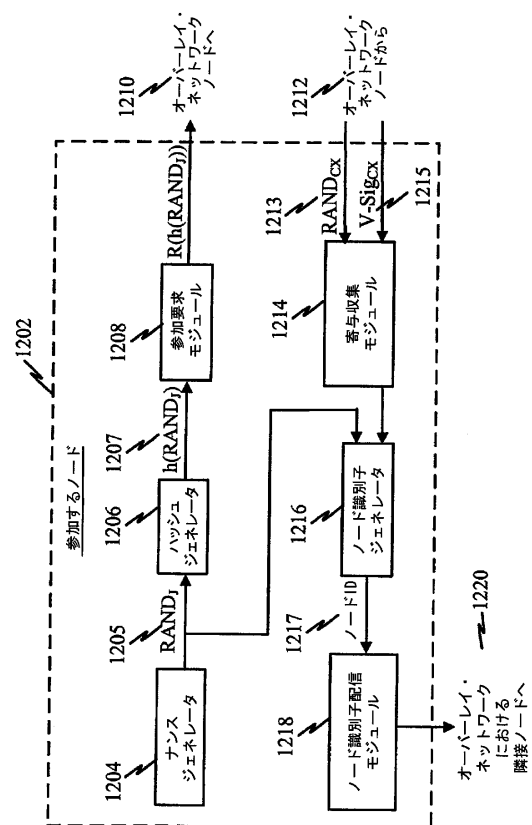


FIGURE 12

【図 13】

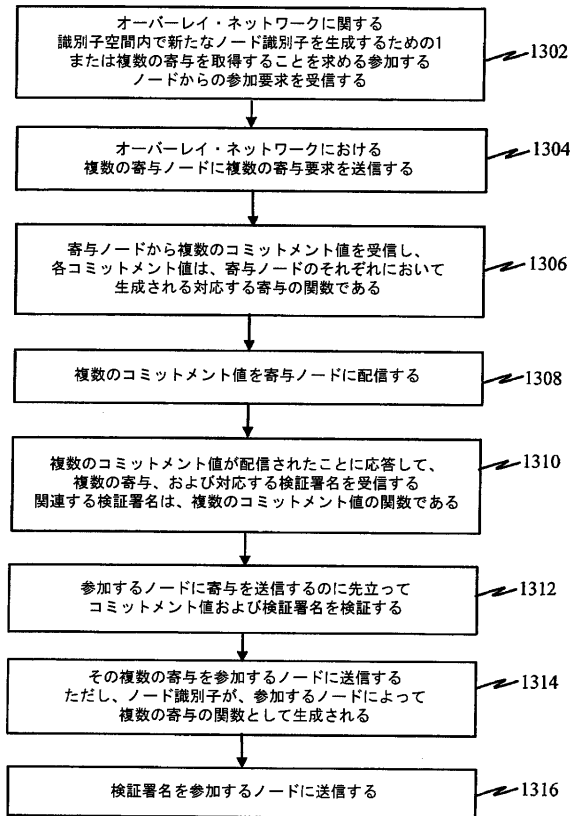


FIGURE 13

【図 14】

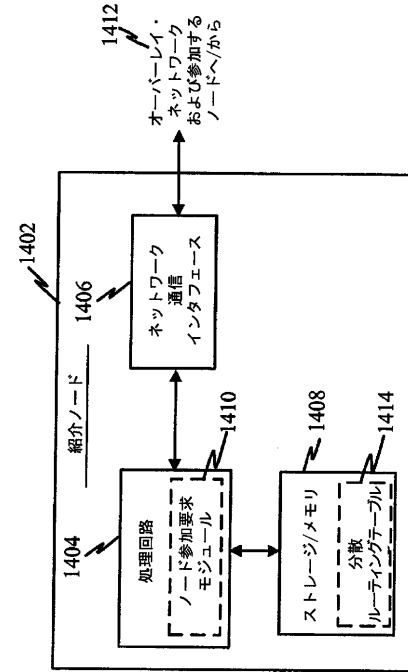


FIGURE 14

【図 15】

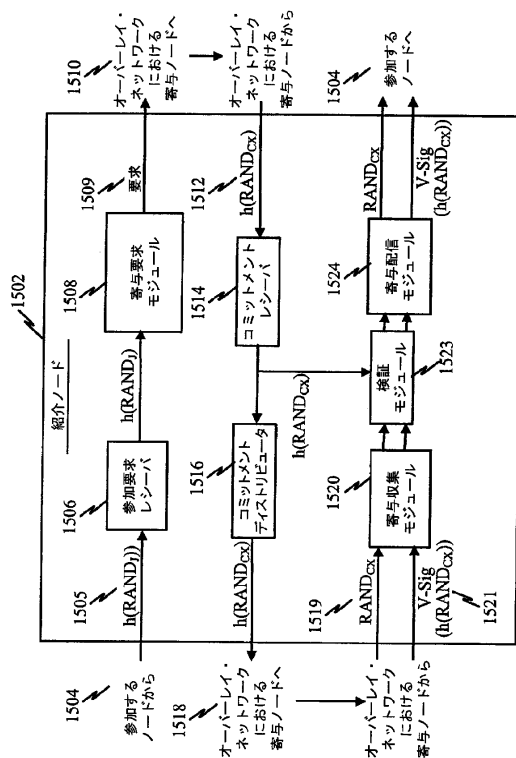


FIGURE 15

【図 16】

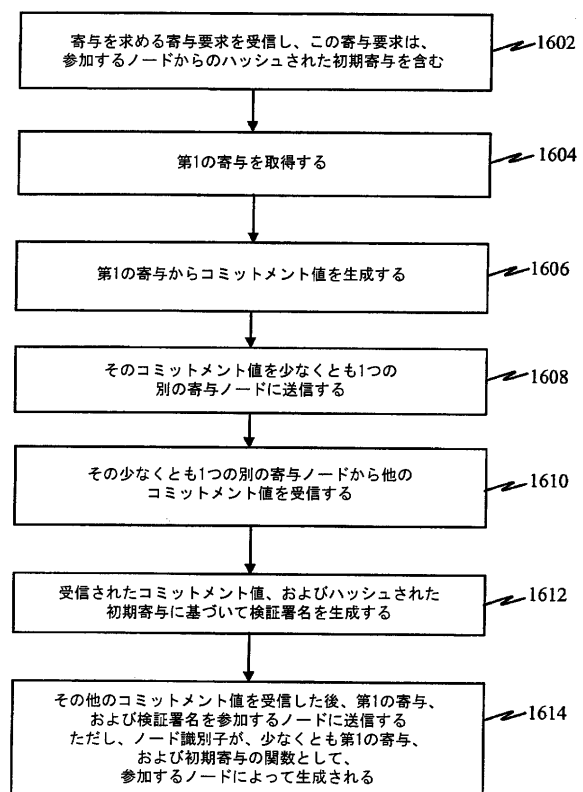


FIGURE 16

フロントページの続き

- (72)発明者 ヴィディヤ・ナラヤナン
アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライブ・5775
- (72)発明者 エドワード・トーマス・リンガム・ハーディー
アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライブ・5775
- (72)発明者 ランジット・エス・ジャヤラム
アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライブ・5775
- (72)発明者 ラマチャンドラン・スブラマニアン
アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライブ・5775
- (72)発明者 ラクシュミナート・レディ・ドンデティ
アメリカ合衆国・カリフォルニア・92121・サン・ディエゴ・モアハウス・ドライブ・5775

審査官 石田 信行

- (56)参考文献 特表2011-503936 (JP, A)
特表2010-519630 (JP, A)
特開2009-17069 (JP, A)
特開2007-235243 (JP, A)
特開2007-193626 (JP, A)
特開2007-53452 (JP, A)
特開2006-191489 (JP, A)
特開2004-266796 (JP, A)

- (58)調査した分野(Int.Cl., DB名)
- | | |
|------|-------|
| H04L | 9/32 |
| H04L | 12/70 |
| G06F | 13/00 |
| G09C | 1/00 |