

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2015-536617

(P2015-536617A)

(43) 公表日 平成27年12月21日(2015.12.21)

(51) Int.Cl.	F I	テーマコード (参考)
HO4L 9/08 (2006.01)	HO4L 9/00 601B	5J104
HO4L 9/32 (2006.01)	HO4L 9/00 675B	
GO6F 21/33 (2013.01)	HO4L 9/00 601F	
	GO6F 21/33	

審査請求 未請求 予備審査請求 未請求 (全 64 頁)

(21) 出願番号 特願2015-541937 (P2015-541937)
 (86) (22) 出願日 平成25年11月8日 (2013.11.8)
 (85) 翻訳文提出日 平成27年7月7日 (2015.7.7)
 (86) 国際出願番号 PCT/US2013/069217
 (87) 国際公開番号 W02014/074865
 (87) 国際公開日 平成26年5月15日 (2014.5.15)
 (31) 優先権主張番号 61/724,763
 (32) 優先日 平成24年11月9日 (2012.11.9)
 (33) 優先権主張国 英国 (GB)

(71) 出願人 515124406
 モスバーガー、 ティモシー
 アメリカ合衆国 カリフォルニア州 91
 942 ラメサ、 レイク マレー ブ
 ルヴァード 5519、107号
 (74) 代理人 100109634
 弁理士 舩谷 威志
 (74) 代理人 100129263
 弁理士 中尾 洋之
 (74) 代理人 100160369
 弁理士 黒田 仁志
 (74) 代理人 100163991
 弁理士 加藤 慎司
 (74) 代理人 100146374
 弁理士 有馬 百子

最終頁に続く

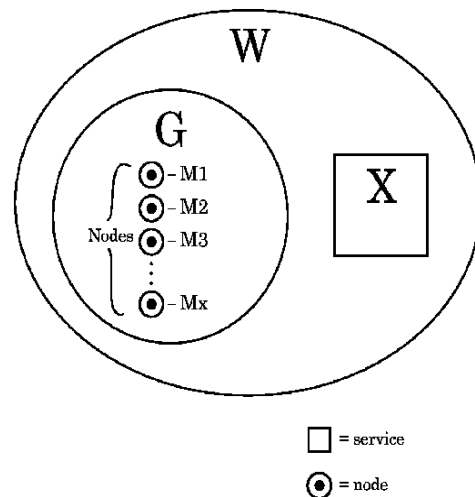
(54) 【発明の名称】 エンティティ・ネットワーク・トランスレーション (ENT)

(57) 【要約】

本発明は、証明書権限および証明書チェーン等の公開非公開キー技術およびPKI概念を用いて、抽象的アイデンティティを識別し認証するためのエンティティネットワークトランスレーション(ENT)スキームを提供する。ENTは、任意の数の要求元に対し、任意の数の真正、不定、かつ抽象的な識別子を与えることができる。これらの抽象的識別子は各々、ペリニムと呼ばれ、これは「実証された名称」を大まかに意味する。それらによって任意の人またはエンティティは、いかなる目的に対しても、物体の真正アイデンティティを電子的に確立しコントロールすることが可能となり、さらにこれらのアイデンティティ間の関係を確立することが可能となる。いくつかの実施形態によれば、ENTは、抽象的識別子をリクエストするユーザに対して抽象的識別子を発行することにより従来のPKI関係確立問題を回避する。それは、これらの抽象的識別子およびその実社会の重要性を定めるエンティティ間に形成された関係を利用することである。

【選択図】

Fig. 1



【特許請求の範囲】**【請求項 1】**

人間、エンティティまたは電子デバイスのための一意識別子を作成するための方法であって、前記方法は、1より大きい数(N)のルートサーバを含むグループ権限構造内で実現され、

第1のルートサーバにおいて、要求元から一意識別子に対するリクエストを受け取るステップと、

前記第1のルートサーバにおいて、一意識別子とポリシーを含む第1の証明書を発行するステップであって、前記ポリシーは、1個またはそれ以上の他の一意識別子を含み、該ポリシー内の他の識別子の数が1より大きい場合には少なくとも1個のブール演算子または数学関数をさらに含むステップと、

前記第1のルートサーバにおいて、該第1のルートサーバに関連付けられた公開/非公開キー対のうちの非公開キーを用いて、前記発行した第1の証明書に署名するステップと、

前記第1のルートサーバから他の前記ルートサーバのそれぞれに対して、前記署名した発行済み第1の証明書を送信するステップと、

他の前記ルートサーバのそれぞれにおいて、前記署名した発行済み第1の証明書の抽象的一意識別子を検証するステップと、

他の前記ルートサーバのそれぞれにおいて、前記一意識別子及び前記ポリシーを含む追加の証明書を発行するステップと、

他の前記ルートサーバのそれぞれにおいて、前記それぞれの他のルートサーバに関連付けられた公開/非公開キー対のうちの非公開キーを用いて、前記発行済み追加証明書に署名するステップと、

データレポジトリにおいて、前記要求元に対する前記署名した発行済み第1の証明書及び前記署名した発行済み追加証明書を記憶するステップと

を備えることを特徴とする方法。

【請求項 2】

前記Nは奇数であり、

前記各ルートサーバは、他の全てのルートサーバから独立して署名し動作することを特徴とする請求項1に記載の方法。

【請求項 3】

2個のルートコンピュータサーバは、2個の異なる要求元に対して同一の一意識別子を発行することはできない

ことを特徴とする請求項1に記載の方法。

【請求項 4】

各ルートサーバは、排他的な範囲の一意識別子を発行することを許可されていることを特徴とする請求項1に記載の方法。

【請求項 5】

前記要求元に対する前記署名した発行済み第1の証明書および前記署名した発行済み追加証明書は、前記要求元のいかなる説明または識別も含まないことを特徴とする請求項1に記載の方法。

【請求項 6】

前記抽象的一意識別子は、多数(X)の前記署名した発行済み第1の証明書および前記署名した発行済み追加証明書が有効である時に有効であるとみなされ、

$X = N / 2 + 1$ である

ことを特徴とする請求項1に記載の方法。

【請求項 7】

前記リクエストは、前記ポリシーをさらに含むことを特徴とする請求項1に記載の方法。

【請求項 8】

10

20

30

40

50

前記ルートサーバにおいて、前記第 1 の発行済み証明書内の前記一意識別子の更新のための更新リクエストを受け取るステップであって、前記更新リクエストは、非公開キーを用いて前記他の一意識別子に関連する人、エンティティまたは電子デバイスのそれぞれによって署名されるステップと、

各ルートサーバにおいて、前記第 1 の発行済み証明書内の前記ポリシーの実行を介して前記更新リクエストを検証するステップと、

各ルートサーバにおいて、前記第 1 の発行済み証明書を交換するための交換証明書を発行するステップと、

各ルートサーバにおいて、該それぞれのルートサーバに関連付けられた公開 / 非公開キー対のうちの非公開キーを用いて前記交換証明書に署名するステップと、

データレポジトリにおいて、前記署名した発行済み交換証明書を記憶するステップとをさらに備えることを特徴とする請求項 1 に記載の方法。

【請求項 9】

前記グループ権限は、前記ポリシーの実施を自動化することを特徴とする請求項 1 に記載の方法。

【請求項 10】

前記第 1 の発行済み証明書は、公開キーまたは前記要求元に関連する公開キーの識別を備える

ことを特徴とする請求項 1 に記載の方法。

【請求項 11】

前記ポリシーは、前記一意識別子を交換または更新するためのポリシーを含むことを特徴とする請求項 1 に記載の方法。

【請求項 12】

前記ポリシーは、前記一意識別子を認証するためのポリシーを含むことを特徴とする請求項 1 に記載の方法。

【請求項 13】

人間、エンティティまたは電子デバイスのための一意識別子を作成するための方法であって、前記方法は、サーバ上で実現され、

前記サーバにおいて、要求元から一意識別子に対するリクエストを受け取るステップと

、
前記サーバにおいて、一意識別子とポリシーと含む第 1 の証明書を発行するステップであって、前記ポリシーは、1 個またはそれ以上の他の一意識別子を含み、該ポリシー内の他の識別子の数が 1 より大きい場合には少なくとも 1 個のブール演算子または数学関数を含むステップと、

前記サーバにおいて、該サーバに関連付けられた公開 / 非公開キー対のうちの非公開キーを用いて、前記発行した第 1 の証明書に署名するステップと、

データレポジトリにおいて、前記署名した発行済み第 1 の証明書を記憶するステップとを備えることを特徴とする方法。

【請求項 14】

前記署名した発行済み第 1 の証明書は、前記要求元のいかなる説明または識別も含まない

ことを特徴とする請求項 13 に記載の方法。

【請求項 15】

前記リクエストは、前記ポリシーをさらに含むことを特徴とする請求項 13 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

30

40

50

関連出願の相互参照

本出願は、2012年11月9日に出願した「エンティティ・ネットワーク・トランスレーション(ENT)のためのシステム及び方法(System and Methods for Entity Network Translation (ENT))」と題する米国特許仮出願番号第61/724,763号に対する優先権を主張するものであり、その開示全体は、本出願の明細書の一部として参照することにより本明細書に組み込まれる。

【0002】

本発明は、応用された暗号法に関し、特に、人間、エンティティおよび電子デバイスの抽象的アイデンティティを識別し、認証するためのデジタル証明書に関する。

【背景技術】

【0003】

センシティブかつ/又はコンフィデンシャルな情報を含むシステムへの安全なアクセスは、周知かつ定着した慣習である。例えば、銀行の顧客は、安全なウェブサイトを通じて、自らの銀行口座についての情報にアクセスすることができる。このような安全なアクセスは、一般に公開キーインフラストラクチャ(PKI)によって提供され、この公開キーインフラストラクチャは、システムに安全にアクセスする際に使用するデジタル証明書を作成、管理、分散、記憶および無効にするために必要とされるハードウェア、ソフトウェア、人々、ポリシー、プロセスの組である。デジタル証明書は、公開キーをアイデンティティと結合するためにデジタル署名を使用する電子的なドキュメントである。公開キー暗号は、ユーザがインターネット等のセキュアでない公衆ネットワーク上で安全に通信し、デジタル署名を介してユーザのアイデンティティを確認することを可能にするPKIと共に用いられる暗号技術である。PKIは、公開キーをエンティティにマッピングするデジタル証明書を作成し、中央リポジトリにこれらの証明書を実際に記憶し、必要に応じてそれらを無効にする。PKIは一般に、デジタル証明書を発行し且つ確認する認証局(CA)と、CAからの情報をリクエストするユーザのアイデンティティを確認する登録局と、インデックスキーを記憶するためのセントラルディレクトリと、証明書管理システムとを含む。

【0004】

従来のPKIシステムでは、発行した証明書が、アイデンティティに直接結合される情報を含む。例えば、証明書が個人に発行される場合、証明書は、電子的な意味における個人のアイデンティティと概念的に交換可能である。

【発明の概要】

【0005】

本発明は、エンティティネットワークトランスレーション(ENT)のための技術を提供する。ENTは、証明書権限および証明書チェーン等の公開非公開キー技術およびPKI概念を用いて抽象的アイデンティティを識別して認証するためのスキームである。ENTは、任意の数の要求元に対し、任意の数の真正、不定、かつ抽象的な識別子を与えることができる。これらの抽象的識別子は各々、ベリニムと呼ばれ、これは「実証された名称」を大まかに意味する。それらによって任意の人またはエンティティは、いかなる目的に対しても、物体の真正アイデンティティを電子的に確立しコントロールすることが可能となり、さらにこれらのアイデンティティ間の関係を確立することが可能となる。いくつかの実施形態によれば、ENTは、抽象的識別子をリクエストするユーザに対して抽象的識別子を発行することにより、従来のPKI関係確立問題を回避する。それは、これらの抽象的識別子およびその実社会の重要性を定めるエンティティ間に形成された関係を利用することである。

【0006】

上記のように、従来のPKIシステムにおいては、発行された証明書は、アイデンティティに直接リンクされた情報を含む。例えば証明書が個人に発行された場合、その証明書は、電子的な意味における個人のアイデンティティと概念的に交換可能である。本発明の実施形態によれば、ENTにおいて、この結合は仮定されない。ベリニムがいかなる特定

10

20

30

40

50

の使用またはコンテキストにもリンクされる、ということは全く仮定されなくてもよい。代わりに、ベリニムは、信頼のおける関係が、任意の目的のために参加者の間に確立され、かつ安定的に維持されることを可能にする。これは、小さなことであるが、現存のPKI解決策との重要な違いである。ENTは、実社会の関係が確立されることを可能にするが、それらが実社会のアイデンティティであることを意味してはいない。関係は、確立のための多くの特定のルールを有することが可能である。銀行は、顧客との関係を確立するために一定の情報を必要とする。ゲームサイトは、他の情報を必要とする可能性がある。ソーシャルネットワークは、さらに異なる基準を有する場合もある。これらの関係の確立のためのプロセスは、問題領域に対して特定のである。しかしながら、本発明の実施形態によれば、ベリニムは抽象的である。

10

【0007】

様々な実施形態において、ベリニムの使用は、要求元によって決定される。利用は、個人に対する例外的なセキュリティを有するオンラインアイデンティティ、コンピュータおよびデバイス、プログラムの識別およびコントロール、会社または個人のグループの識別等を含むことができる。本発明の実施形態によれば、ENTは、領域に固有の技術を必要とすることなく、その能力を通じて、これらの問題領域の全てに亘って使用され得る価値、又はそれ以上の価値を提供することが出来る。ENTは、標準化された包括的なソリューションを用いて、これらの領域に特有のソリューションの多くを減少させ、または除去することができる。さらに、ENTは、問題領域に亘る包括的なENTインタフェースおよび機構を用いて、情報、アクセス、コマンドおよびコントロール等の共有を可能にする

20

【0008】

本発明の一実施形態では、人間、エンティティまたは電子デバイスのための一意識別子を作成するための方法であって、前記方法は、1より大きい数(N)のルートサーバを含むグループ権限構造内で実現され、第1のルートサーバにおいて、要求元から一意識別子に対するリクエストを受け取るステップと、前記第1のルートサーバにおいて、一意識別子とポリシーを含む第1の証明書を発行するステップであって、前記ポリシーは、1個またはそれ以上の他の一意識別子を含み、該ポリシー内の他の識別子の数が1より大きい場合には少なくとも1個のブール演算子または数学関数をさらに含むステップと、前記第1のルートサーバにおいて、該第1のルートサーバに関連付けられた公開/非公開キー対のうち非公開キーを用いて、前記発行した第1の証明書に署名するステップと、前記第1のルートサーバから他の前記ルートサーバのそれぞれに対して、前記署名した発行済み第1の証明書を送信するステップと、他の前記ルートサーバのそれぞれにおいて、前記署名した発行済み第1の証明書の抽象的一意識別子を検証するステップと、他の前記ルートサーバのそれぞれにおいて、前記一意識別子及び前記ポリシーを含む追加の証明書を発行するステップと、他の前記ルートサーバのそれぞれにおいて、前記それぞれの他のルートサーバに関連付けられた公開/非公開キー対のうち非公開キーを用いて、前記発行済み追加証明書に署名するステップと、データレポジトリにおいて、前記要求元に対する前記署名した発行済み第1の証明書及び前記署名した発行済み追加証明書を記憶するステップとを備える。Nは奇数であり、各ルートサーバは、他の全てのルートサーバから独立して署名し動作する。2個のルートコンピュータサーバは、2個の異なる要求元に対して同一の抽象的な一意識別子を発行することはできない。各ルートサーバは、排他的な範囲の一意識別子を発行することを許可されている。前記要求元に対する前記署名した発行済み第1の証明書および前記署名した発行済み追加証明書は、前記要求元のいかなる説明または識別も含まない。前記抽象的一意識別子は、多数(X)の前記署名した発行済み第1の証明書および前記署名した発行済み追加証明書が有効である時に有効であるとみなされ、 $X = N / 2 + 1$ である。前記リクエストは、前記ポリシーをさらに含む。前記方法は、前記ルートサーバにおいて、前記第1の発行済み証明書内の前記一意識別子の更新のための更新

30

40

50

リクエストを受け取るステップであって、前記更新リクエストは、非公開キーを用いて前記他の一意識別子に関連する人、エンティティまたは電子デバイスのそれぞれによって署名されるステップと、各ルートサーバにおいて、前記第1の発行済み証明書内の前記ポリシーの実行を介して前記更新リクエストを検証するステップと、各ルートサーバにおいて、前記第1の発行済み証明書を交換するための交換証明書を発行するステップと、各ルートサーバにおいて、該それぞれのルートサーバに関連付けられた公開/非公開キー対のうちの非公開キーを用いて前記交換証明書に署名するステップと、データレポジトリにおいて、前記署名した発行済み交換証明書を記憶するステップとをさらに備える。前記グループ権限は、前記ポリシーの実施を自動化する。前記第1の発行済み証明書は、公開キーまたは前記要求元に関連する公開キーの識別を備える。前記ポリシーは、前記一意識別子を交換孔または更新するためのポリシーを含む。前記ポリシーは、前記一意識別子を認証するためのポリシーを含む。

10

【0009】

本発明の他の実施形態では、人間、エンティティまたは電子デバイスのための一意識別子を作成するための方法であって、前記方法は、サーバ上で実現され、前記サーバにおいて、要求元から一意識別子に対するリクエストを受け取るステップと、前記サーバにおいて、一意識別子とポリシーとを含む第1の証明書を発行するステップであって、前記ポリシーは、1個またはそれ以上の他の一意識別子を含み、該ポリシー内の他の識別子の数が1より大きい場合には少なくとも1個のブール演算子または数学関数とを含むステップと、前記サーバにおいて、該サーバに関連付けられた公開/非公開キー対のうちの非公開キーを用いて、前記発行した第1の証明書に署名するステップと、データレポジトリにおいて、前記署名した発行済み第1の証明書を記憶するステップとを備える。前記署名した発行済み第1の証明書は、前記要求元のいかなる説明または識別も含まない。前記リクエストは、前記ポリシーをさらに含む。

20

【0010】

本発明の前記の、ならびに他の特徴および利点は、本発明の好ましい実施形態、添付図面および請求の範囲の以下のより特定のな説明から明らかになるであろう。

【図面の簡単な説明】

【0011】

本発明、並びに、その目的および利点をより完璧に理解するために、以下で簡単に述べる添付図面と関連付けて、後続の説明を参照する。

30

【0012】

【図1】本発明の一実施形態によるエンティティおよびエンティティ間の関係を示す。

【0013】

【図2】本発明の一実施形態による自己署名および相互署名証明書を作成するためのプロセスを示す。

【0014】

【図3】本発明の他の実施形態による自己署名および相互署名証明書を作成するためのプロセスを示す。

40

【0015】

【図4】本発明の一実施形態によるエンティティにアクセスする可能性のある初期の認可グループおよび非認可グループを示す。

【0016】

【図5.1】本発明の一実施形態による証明書を置換するためのプロセスを示す。

【0017】

【図5.2】図5.1のプロセスにおいて利用される証明書間の関係を示す。

【0018】

【図6】本発明の実施形態による自己署名および相互署名の証明書を示す。

【0019】

【図7】本発明の実施形態による証明書間の関係を示す。

50

【 0 0 2 0 】

【 図 8 】 本発明の実施形態によるエンティティの関係を示す。

【 0 0 2 1 】

【 図 9 】 本発明の他の実施形態による自己署名および相互署名の証明書を示す。

【 0 0 2 2 】

【 図 1 0 】 本発明の他の実施形態による認可グループの相互署名文書を示す。

【 0 0 2 3 】

【 図 1 1 】 本発明の他の実施形態による置換認可グループの相互署名文書を示す。

【 0 0 2 4 】

【 図 1 2 】 本発明の他の実施形態による、文書を後の文書と置換するために使用する代数学を含む文書を示す。 10

【 0 0 2 5 】

【 図 1 3 】 本発明の一実施形態による、証明書を作成するためのプロセスを示す。

【 0 0 2 6 】

【 図 1 4 】 本発明の一実施形態によるエンティティのグループを示す。

【 0 0 2 7 】

【 図 1 5 】 本発明の一実施形態による J S O N 信任状の例を示す。

【 0 0 2 8 】

【 図 1 6 】 本発明の一実施形態による証明書を作成するためのプロセスを示す。

【 0 0 2 9 】

【 図 1 7 】 本発明の他の実施形態による、ピア署名者を利用する証明書に対する置換リクエストを示す。 20

【 0 0 3 0 】

【 図 1 8 】 本発明の他の実施形態による、記憶装置内の証明書の、より大きい通し番号を有する他の証明書との置換を示す。

【 0 0 3 1 】

【 図 1 9 】 本発明の他の実施形態による、記憶装置内の証明書の、より大きい通し番号を有する他の証明書との置換を示す。

【 0 0 3 2 】

【 図 2 0 】 様々な他のシステムにアクセスするために E N T システムを使用し得るユーザーアクセスタミナルを含む、一実施形態によるエンティティ・ネットワーク・トランслーション (E N T) システムのブロック図である。 30

【 発明を実施するための形態 】

【 0 0 3 3 】

本発明の好ましい実施形態およびその利点は、図 1 ないし図 2 0 を参照して理解することが可能であり、同じ参照符号は同じエレメントを示す。様々な実施形態は、エンティティ・ネットワーク・トランслーション (E N T) のためのシステムおよび方法を提供する。実施形態によれば、E N T は P K I システムである。これは、非公開 / 公開キーと、中央権限と、証明書と、証明書チェイニングとを利用する。E N T はまた、当業者にとってはその実務が直ちに明らかな、現存の技術インフラ、並びに、トランスポート・レイヤー・セキュリティ (T L S) や X . 5 0 9 のような暗号プロトコル及び規格を活用するように設計されている。これによって、(大抵の場合) 現存のシステムを直接変更することなく E N T をそのシステム内で利用することが可能になる。E N T がこれらの現存の技術を利用することは必要条件ではないが、有益であり得る。 40

【 0 0 3 4 】

実施形態による E N T は、典型的な P K I システムではない。それは、全ての基本 P K I アクティビティの強力な自動化により、例外的なスケーラビリティ、耐性および監査を提供できるように設計されている。実質的なりサーチおよび開発が、これらの目的を達成することに費やされた。より形式的には、実施形態による E N T の目的は以下の通りである：

【 0 0 3 5 】

1 . ベリニム (v e r i n y m) の 「 キャノピー 」 を作成する。 E N T は、これらのアイデンティティが、任意の目的のために、第三者間で安全かつ認証された通信のために使用され得ることを確実にすることができる。各々が一つまたはそれ以上のベリニムを所有する全ての第三者の組が、キャノピーを構成する。

【 0 0 3 6 】

2 . いずれかの既存の製品 P K I システムと同等以上の極めて強力な暗号および P K I サービスを提供する。 E N T は、これらのサービスを分配方式で提供可能であり、システム内のベリニムの一意性を損なうことなく、システムの信頼性および安定性に悪影響を及ぼす可能性のある停止、トランクセキュリティの損失、および他の深刻なイベントを許容する。

10

【 0 0 3 7 】

3 . 各ベリニムの直接コントロールを所有者に委託し、いかなる目的のための使用も許可する。一旦ベリニムを作成すると、 E N T システムはもはや、アイデンティティ所有者による暗号の 「 所有権の証明 」 が伴わなければならない所与のベリニムとの関連の定期的更新以外のベリニムの利用に対するコントロールを有しない。

【 0 0 3 8 】

4 . これらのサービスを冗長に、しかも出来るだけ安く提供する。現存するほとんどの P K I システムは、そのコアにおいて 1 個のルート証明書を有する階層的署名機構に依存する。この単一障害点は、欠陥が破局的となることから、莫大な費用の P K I システムを生み出す。人材および実環境プロセスを必要とするシステムを通じて、さらなる費用がかかる。 E N T は、技術革新を通じて、安全性を低減することなく、コストの削減を可能にする。実際、多くのディメンションにおいては、 E N T は、かなりコストを削減した既存の設計よりも実質上安全である。 E N T が既存の P K I システムほど安全でないディメンションは存在しない。

20

【 0 0 3 9 】

5 . ユーザに意識させることなく動作し、ユーザおよび監査人による正常で信頼のおけるチェックを可能にする。これは、システムセキュリティ欠陥、バックドアおよび他の信頼のおけない動きが隠れていられなくなることを確実にする。

【 0 0 4 0 】

6 . ベリニムの利用が、デフォルト設定で、抽象的かつ匿名であることを確実にする。プライベートなシステムは、プライベートでないシステムを構築するために利用され得る。その逆は真実ではない。

30

P K I 定義 :

【 0 0 4 1 】

証明書は、公開 / 非公開キーペア (P P K) に対応する公開キー、および、いくつかの付加的な任意情報を含む、暗号で署名されたメッセージであり、場合によっては、異なる P P K に対応する非公開キーによる署名である。その証明書の 「 ターゲット 」 は、その P P K、または、その証明書内の非公開キーの所有者である。 「 署名者 」 は、その P P K、または、その証明書に署名するために使用する非公開キーの所有者である。

40

【 0 0 4 2 】

P P K の非公開部分とその証明書に署名するために用いられた場合、証明書は 「 署名された 」 とみなされる。さらなる明確性のために、証明書の 「 ターゲット 」 は、 P P K、または、非公開キーが証明書内に存在する P P K の所有者として定義される。その証明書内に見出される公開キーが P P K の公開部分であり、その証明書の署名がその P P K の整合非公開部分である場合、証明書は 「 自己署名された 」 とみなされる。

【 0 0 4 3 】

その証明書に見出される公開キーが P P K の公開部分であり、その証明書の署名がその P P K の整合非公開部分を用いて作成されたものである場合、証明書は 「 自己署名された 」 ものとみなされる。

50

【 0 0 4 4 】

ここで参照するように、P P K が行動を遂行することは、証明書が P P K の公開部分を含むこととして参照される。その理由は、両方が同じ所有者に関するものだからである。例えば、証明書 A が P P K P のための公開キーを含む場合、「A が証明書 B に署名した場合」等の文は、P が B に署名する、として読まれるべきである。その理由は、非公開キーは、P P K 所有者による行動を遂行するために使用される装置であるからである。P の公開キー部分は A 内に存在するので、このチェイニングおよび関連付けは論理的であり、より容易に読み出される。

【 0 0 4 5 】

さらにここで参照されるように、「ターゲット」の動詞形態は、その「ターゲット化」の主題エンティティまたは P P K が、対象となった証明書内にその公開キーを有することを示す。例えば、証明書 A が P P K P を含み、証明書 B が P P K Q を含む場合、A に対応する P P K (この場合 P) が Q の公開キー部分を含むいずれかの証明書に署名したのであれば、「A は B をターゲットとする」ことになる。B を「ターゲットとする」ものは、Q の公開部分を含む証明書に署名した任意の P P K となる。「A が B をターゲットとする」および「B が A によってターゲットにされる」は、同じ意味を表す。

10

【 0 0 4 6 】

非対称暗号法は、その識別および実装が当業者にとって明らかである E C C、R S A 等の技術を含む一方、ゼロ知識証明機構も含む。これらの場合、署名は不可能であるが、秘密の所有権を証明するトランザクションは可能である。したがって我々は、署名、トランザクションを介して、または他の機構により、信頼性を証明することが可能ないずれかの技術として、我々の目的のための非対称暗号法について考えることが出来る。これらの技術の機構は、本開示の範囲を超越しており、当業者によって容易に理解される。

20

グループコマンドおよびコントロール：

【 0 0 4 7 】

従来の P K I システムには、周知のように、証明書を発行し、証明書関連のタスクを実施する認証局 (C A) と呼ばれるセントラルサーバがある。このセントラルサーバは、C A を表す P P K を含む。この P P K 暗号プリミティブは、証明書、失効または更新に署名し、発行するために用いられる。C A または C A の P P K が不正にアクセスされると、P K I システム全体が不正にアクセスされることになる。E N T において C A の同等物を実装の具体的な実施形態を分析する前に、グループコマンドおよびコントロールと呼ばれる新規な技術の概念化について述べる。

30

【 0 0 4 8 】

グループコマンドおよびコントロールは、メンバのグループとして定義され、その各々は、1 個のキーまたは単一障害点に限定されることなくコマンドを発行し、グループのビジネスを処理する 1 個の概念エンティティを形成する P P K をコントロールする。グループは、概念的エンティティの情報漏洩なしに閾値までの損害を被る可能性があり、グループメンバを置換可能にすることにより、堅牢な長期間安定性を可能にする。多数のグループメンバがそれぞれ異なるセキュリティプロトコルおよびプロセスと共に P P K を使用するシステムをサポートすることにより、破局的失敗のリスクはさらに減じられる。グループメンバの例は、1 人の所有者、グループとして行動する多数のユーザ、またはユーザのグループのグループ等のようなより抽象的な概念を有する多数の装置であってもよい。

40

【 0 0 4 9 】

この概念の 1 つの価値は、ノードのグループが 1 個のエンティティとして作用することが可能となる固有のシステム内における、多数の P P K の利用を介して P P K のコントロールが失われることに起因する損害の低減である。たとえある一定のプリミティブが不正アクセスされるか、または失われたとしても、損害を防止することが可能となる。様々な暗号プリミティブから成る不均一システムを利用することにより、リスクをさらに低減することが可能になる。例えば、あるノードは、R S A 暗号化プロセスを利用可能である。他のノードは、D S A を利用可能である。他のノードは、楕円曲線を利用可能である。使

50

用される異種プロセスの上限は、グループ内のノード数の上限である。

【0050】

図1を参照して、より詳細に説明する。仮想エンティティを表すGと呼ばれるN個のメンバノードを有するグループを定義する。この仮想エンティティはそれ自体の識別子を有することが可能であるか、またはその識別子は、例えば、そのメンバノード名の全てを順序付け、その値をハッシュし、そのハッシュを識別子として利用すること、などのような、そのメンバの照合であり得る。エンティティGがコマンドおよびコントロールを実施することを可能にする全ての装置またな外部関係者から成るグループWを定義する。このコントロールは、データへのアクセス、コードの実行またはWのメンバがGを認証したいと思う他のいかなるアクションともなり得る。すなわち、Wのメンバは、グループGに対するアクションを可能にし、他のいずれかのグループに対するアクションを防止することを望む。Gのx番目のメンバとして、ノードM_xを定義する。M_xノードは、グループGの目的を達成するために、PK_xを行使する。XをW内のユーザとして定義する。1つの実装では、Nは常に奇数である。これは、もしアタッカがちょうどN/2個のノードを占拠し、かつ、Nが偶数であった場合におき得るシステムのデッドロックを、アタッカが引き起こすことを妨げる。

10

【0051】

一実施形態では、1個のM_xノードに「タイ・ブレーカ」権限が与えられる。この場合、偶数個のノードが許可される。もし、Nが偶数であってアタッカがN/2個のノードを占拠した場合、タイ・ブレーカノードはデッドロックを防止する。タイ・ブレーカM_xノードは、常に同じノードであってもよいか、またはGのメンバに依って変化してもよい。例えば、Gの最も古いメンバは、Gに対するタイ・ブレーカとなり得る。代わりに、Gの最も新しいメンバには、タイ・ブレーカステータスを与えることが可能である。他の実施形態により、実装は変化可能である。

20

【0052】

図1を続けて参照すると、いくつかの実施形態においては、Gは技術として非対称暗号法を使用する。例えばX509のような、PKIが組み立てる1つの実装においては、証明書が使用可能である。いくつかの実施形態は、その実装が当業者にとって明らかなジャバスクリプト・オブジェクト・ノーテーション(JSON)または拡張可能マークアップ言語(XML)フォーマット等のより現代的なデータ交換フォーマットを利用することも可能であるが、それに限定されない。

30

【0053】

1つの実装においては、図2に示すように、以下のステップが行われる。(1)各M_xが、非対象暗号プリミティブを用いて、非公開キーと、(前記キーを有する)自己署名証明書M_xS_xとを作成する。(2)各M_x署名が、互いにM_yの証明書に署名する。これらの証明書の1個をM_xS_yとして定義する。例えば、Nが3である場合、M₁はM₂およびM₃の証明書に署名し(M₁S₂およびM₁S₃を作成し)、M₂はM₂S₃およびM₂S₁を作成し、M₃は証明書M₃S₁およびM₃S₂を作成する。N=3に対しては、G(ステップ2)に対して、3つの自己署名証明書(ステップ1)および6つの相互署名証明書が存在することになる。(3)GCを、全てのM_xノードに対して、それぞれの自己署名M_xS_xを含むステップ2からの証明書の全体集合として定義する。したがって、サイズNの任意のGに対して、N個の自己署名証明書、N個のノード、および(N-1)*(N-1)個の相互署名が存在し、GCの組において合計N*N個の証明書をもたらすことになる。

40

【0054】

1つの実装においては、各M_xは「ラウンドロビン」プロセスを使用するN-1個の証明書の代わりに、ちょうどN/2個の(切り捨てられた)証明書に署名する。この場合、M₁が常にM₂等の「前」になるように、確定的なオーダプロセスを用いてM_xにより作成された全ての証明書をリストL内に順序付ける。この組はGCを含む。M_xによって署名されたN/2個の証明書は、次により大きいN/2個の(切り上げられた)証明書であ

50

る。この計算が、リストの最後を通過して拡張された場合、該計算は、Lにこれ以上の証明書が存在しない時、リストの最初で継続するべきである。例えば、 $N = 4$ に対して、 M_1 は、 M_2 および M_3 に対する証明書に署名し、 M_4 は、 M_1 および M_2 に対する証明書に署名する。 M_3 は、 M_4 および M_1 に対する証明書に署名する。この組は、GCを含む。図3は、 $N = 7$ の場合における、そのようなラウンドロビン署名技術の一例を示す。

【0055】

1つの実装において、各 M_x は $N - 1$ 個の証明書に署名する。すなわち、各 M_x は互いに一意の M_y をターゲットとする $N - 1$ 個の証明書を作成する。1つの実装において、G内の各 M_x は、異なる非対称暗号プロセスを使用することが可能である。例えば、 N が3である場合、1個のノードはRSAキーペアを使用可能であり、もう1個はDSAキーペアを使用可能であり、残り1個は楕円曲線暗号法を使用可能である。1つの実装においては、各 M_x は、自己署名済み証明書を作成する代わりに、CAにより署名された証明書を使用する。これらの実装の全てにおいて、GCは、G内の任意の $M_x S_x$ に対して $N / 2$ 個よりも多い署名済み証明書が存在するように、自己署名した、または相互署名した証明書のリストから構成される。すなわち、各ノード M_x に対して、 M_x により使用されるPPKを含む $N / 2 + 1$ 個の署名済み証明書が常に存在する。そうすることでGCは、Gと相互作用する際に、Xによる使用のために分析され得ることになる。

10

【0056】

Xは、GCの最初のローカルコピーを与えられなければならない。Xがいずれかのアクションを実施可能になる前に、ローカルストア内にGCを有することが重要である。Xの記憶済み証明書Tの組を呼び出す。ローカルストアまたはコピーは、データを含むRAMまたはディスクメモリ等のコンピュータメモリの一部である。この場合、その記憶装置はTを含む。

20

【0057】

図4を参照すると、1つの実装において、初めにGがXからのサービスをリクエストしたとき、XはGCを受け取る。この時、Xは、サービス初期化の一部としてGCをリクエスト可能であった。初期化時点では、 $T = GC$ である。すなわち、信頼のおけるストアは、正確にGCを含む。GCの前バージョンが存在せず、XがGの事前知識を有さないので、初期通信上でGCからXへの通過は安全である。Xと通信する他のGグループに対して、Xは別のTを記憶する。Tは、Gに対する一意の識別子と同等であることに注目されたい。これは、アタッカにより初期通信ラウンドにおけるGCが汚染されることを防止する。アタッカがXに対して変更されたTを提示する場合、GとTがミスマッチとなる。Xは、Gの代わりにGに対抗するTを記録するだろう。その後GがTを提示した場合、XはGをGと混同せず、TをTと混同しないだろう。Xは、Gがコンタクトする際にはTを使用し、アタッカがコンタクトする際にはTを使用するだろう。

30

【0058】

過半数を、 $N / 2$ よりも大きいカウント、切り上げとして定義する。または、タイブレーカが存在するケースでは、上記カウントは $N / 2$ 以上であり、タイブレーカはカウントの一部となる。したがって、 $N = 3$ に対して、過半数は2となる。 N が35である場合、過半数は18となる。

40

【0059】

ALGO1と呼ばれる特定の実装について、図5.1および図5.2を参照して以下で述べる。この実装では、Xは以下の方法により、Tが首尾一貫していることを確認できる。

1) Xは、まず、Tにおける全ての有効な証明書から構成される集合TVを計算する。Tにおける有効な証明書とは、

- a) 自己署名済み証明書($M_x S_x$)であり、または
- b) 自己署名済み証明書がT内にあるT内の M_x ($M_x S_y$)によって相互署名済みであり、および、
- c) 例えば満了、以前は妥当でなかった、フォーマット等、他の証明書有効性ルールを

50

満たす

証明書である。

2) 一意の公開キーを含む $T V$ 内の全ての $M \times S \times$ 証明書の集合 $T S S$ を定義する。
 3) $T S S$ 内のいずれかの証明書によって署名された $T V$ 内の全ての証明書を含む集合 $T V$ を定義する。すなわち、 $T V$ は、 $M \times S \times$ が $T S S$ 内にある任意の $M \times S y$ を含む。

4) 空集合 $T V$ を作成する。

5) $T S S$ 内の各証明書 y に対して、以下を実施する。

a) $T V$ 内の全ての $M \times S y$ のカウントが $T S S$ 内で見つかった証明書の過半数である場合、 $T V$ 内の全ての証明書 $M y S x$ を $T V$ に追加する。これは、全ての $M y S y$ (自己署名済み) 証明書を含むべきである。他の $M \times$ ノードのうちの過半数の署名を有さない $M y$ は、このステップのため、 $T V$ に加えられない。

6) $T V$ 内の全ての自己署名済み証明書を含む集合 $T S S$ を作成する。

7) $T S S$ 内のいずれかの証明書によって署名された $T V$ 内で見つかった全ての証明書を含む集合 $G T$ を作成する。 $G T$ に加えられた証明書は、 $T S S$ に存在しない $M \times$ ノードをターゲットとする証明書を含まない。

8) X は T を $G T$ と置換する。

典型的な実装において、最初は $T = G C = G T$ である。

【0060】

実際、 T 内の有効な $M \times S x$ を有するノードによって署名された T 内の任意の有効証明書 $M \times S p$ は、ノード p が T 内に $M p S p$ を有しない場合には、破棄されないであろう。その証明書は、以下で述べる $A L G O 2$ と共に後で使用するために取っておく。要点は、 $M \times S x$ が信頼され、 x によって署名された証明書もまた信頼されているが、使用されていないということである。

【0061】

$A L G O 1$ のステップ 8 は、 X が T から証明書を切り取ることを許可することに注目されたい。すなわち、 G 内の過半数のノードの信頼を有さないノードが署名した証明書は、 T から破棄される。また、 $A L G O 1$ は T を変化させることに注目されたい。これは、 T 入力を必要とし、出力として置換 T を生成する。いずれかの T 上で繰り返し動作する場合、 $A L G O 1$ は、一度反復した後、不変状態に到達する。すなわち、 $A L G O 1$ が T を入力として処理し、 $G T$ を出力として生成する場合、 $G T$ 上における $A L G O 1$ の任意の将来の反復によって、 $G T$ が完全に生成される。 $A L G O 1$ は、冪等元である。

【0062】

ノードのいくつかの集合に対して、 $N / 2$ 個以下の相互ターゲット化証明書が存在する場合、 $A L G O 1$ が空の T を生成可能であることに注目されたい。 X に渡された最初の $G C$ が証明書の適切な集合を含むことが極めて重大である。1つの実装において、これは、 G 内のノードに対するただ1つの自己署名済み証明書を含むように $G C$ を設定し、その後、 $A L G O 2$ および $A L G O 3$ (以下で議論する) を用いて T を「成長させる」ことによって達成可能となる。

【0063】

1つの実装において、 T 内の有効な証明書は、「満了」時間値を含む証明書であり、現在の日時 ($A L G O 1$ が動作する際に計算される) がその時間値を過ぎていないものである。過去の「満了」時間値を有する証明書は、無効であるとみなされる。

【0064】

1つの実装において、 T 内の有効な証明書は、「以降有効」の時間値を含む証明書を含む証明書であり、現在の日時 ($A L G O 1$ が動作する際に計算される) がその時間値を過ぎていないものである。将来の「以降有効」の時間値を有する証明書は、無効であるとみなされる。

【0065】

ノードを G に追加する: 1つの実装においては、 G 内のノードは、以下の機構を用いて

10

20

30

40

50

証明書を作成することにより、Gに対して追加のノードを作成し追加することができる。これらのノードは、その後、ALGO2を使用し得るWのメンバに対し、その信頼のおけるストアを更新するために送られ得る。

【0066】

図6を参照して、ALGO3と呼ばれる他のプロセスについて述べる。この実装においては、ALGO3は以下を含む。

- 1) 新たなノードMpが非公開キーと、非対称暗号プリミティブを用いる自己署名済み証明書(MpSp)とを作成する。
- 2) G内の各ノードMxが、MxSp証明書を作成する。
- 3) Mpが、G内の他のノードのそれぞれに対するMpSx証明書を作成する。
- 4) ステップ1、2および3の結果から構成される集合INを定義する。INは、 $(N * 2 + 1)$ 個の証明書を含む。

10

【0067】

1つの実装において、集合INは、1個またはそれ以上の新たなノードによって作成された証明書を含むことができる。また、アタッカによって送られるIN証明書集合が、適切な証明書に加えて、任意の他の不正確な証明書を含む可能性もあることに注目されたい。

【0068】

1つの実装においては、Mp個のノードは、常に対で作成される。すなわち、 $2Mp$ 個のノードが常にALGO3内で作成される。これは、Nが奇数である必要があるときに極めて重大となる。タイブレーカが存在し、Nが偶数である場合には、必要ではない。

20

【0069】

1つの実装においては、Xは、証明書を追加することによって新たなTを作成することが可能である。これは、より多くのノードを含むか、または、有効性のためにもはやT内に存在しないノードを置き換えるために、XがGを安全に再定義することを可能にすることから、望ましいものであり得る。Xは、証明書INの集合を受け取る。以下のALGO2によって、我々は、INのどの部分がTに追加されるべきかを計算することが可能になる。このプロセスによって、我々は、(アタッカによって送られた)不正確な証明書を、その証明書が我々の信頼のおけるストアTに入る前に排除することが可能になる。

【0070】

30

図7を参照して、ALGO2と呼ばれる他のプロセスについて述べる。本実装においては、ALGO2は以下を含む。

- 1) IN内にMxSx証明書の集合SSを作成する。これは、Tへのエントリを入念に検査されているIn内の全ての自己署名済み証明書の集合である。
- 2) SS内の各MySy証明書に対して(yは入念に検査されるべきノードである)、
 - a) T内のいずれかのMxSxによって署名されたTまたはIN内の全ての証明書の集団Tを作成する。T内のMxSxは信頼性があるため、Tは、TまたはINに存在する信頼のおけるノードによって署名された全ての証明書を含む。
 - b) T内の全てのMxSyの集団VTを作成する。VTは、yをターゲットとする全ての信頼性ある証明書の集団である。
 - c) ALGO1のステップ1で設立されたものと同様の有効性ルールを用いてVT内の各証明書の有効性をチェックし、VTから無効証明書を破棄する。
 - d) VT内で任意のMxSyに署名を行ったT内のMxSxの数の合計がT内の全てのMxSxの過半数でない場合、SS内の次のMySyに対して、ステップ2を繰り返す。この場合、yは、適切に検査されなかった。信頼性のおけるノードの過半数は、yを保証する証明書を作成しなかった。
 - e) IN内の全てのMySx証明書の集合VCを作成する(この場合、xはT内のMxSxによって表されるノードである)。これは、yによって署名され、T内の信頼性あるノードをターゲットとする全ての証明書の集合である。
 - f) VC内の証明書の合計がT内の全てのMxSxの過半数である場合、MySyおよ

40

50

び VC 内の全ての証明書を T に追加する。入念に検査されたノード y が T 内の信頼性あるノードの過半数を保証した場合、信頼性ある集合 T に、T 内の信頼性あるノードをターゲットとする y の全ての証明書とともに、y の自己署名済み証明書を追加する。

g) T を T と置換する。

3) T 上で ALGO 1 を実施する。

【0071】

ALGO 2 は、X に対して既知であるように、G 内のノードの数が増加することを許容する。その理由は、T はそれらの新たなノードに対する証明書を含むからである。

【0072】

ノードを G から除去する：ノードを G に追加可能であることに加えて、G からノードを除去できることもまた有益である。1 つの実装において、G 内のノードは、以下の方法で G からノード M p を除去できる。失効証明書を、失効のターゲット (M p) と、G 内のノード (M x) による署名と、失効値を含む証明書として定義する。1 つの実装においては、失効値は、「無効」と呼ばれる証明書の値フィールドである。失効値は、X 並びに G および W のメンバがその証明書のターゲットが無効であることを意味することを理解する何らかの値であればよい。有効な失効証明書は、M x に対する有効な署名を有するものである。

【0073】

ALGO 4 と呼ばれる他のプロセスについて述べる。この実装においては、ALGO 4 は以下を含む。

1) M p をターゲットとする全ての M x によって作成された失効証明書の集合 G R を作成する。M x およびターゲットである M p によって作成された G R 内の任意の証明書として M x R p を定義する。

2) G R は X (および一般的には W) に分配される。

【0074】

1 つの実装において、X は証明書ストア T R 内に、そのような全ての失効証明書を記憶する。

【0075】

1 つの実装において、T 内の有効証明書は、T R 内に証明書 M x R p が存在しない証明書 M x M p である。すなわち、本実装は、T R 内の有効 M x R p の存在が T 内のいくつかの M x S p を無効にするように、ALGO 1 のステップ 1 を修正する。T R 内に証明書 M x R p が存在し、T 内に証明書 M x S p が存在する場合、T 内の M x S p はもはや有効ではない。好ましい実装では、失効証明書 G R の集合を受け取ったことに応じて、X が G R を T R に追加し、ALGO 1 を実施する。好ましい実装においては、X は、T 内の M x S x によって署名された G R からの証明書を、T R に追加するのみである。

【0076】

1 つの実装においては、失効証明書 M y R y は (ここでは M y ノードがそれ自体を無効にする)、M y によって署名された全ての証明書が、その M y S y 証明書を含むことから、無効であると考えられるべきである。これは、ノードが自身を無効にすることを可能にする。1 つの実装においては、これは、M x ノードが M x R y 失効証明書を作成することを除外するべきでない。

【0077】

1 つの実装においては、各 M x が C A によって署名された証明書を使用し、その C A が M x に対する失効証明書を発行する場合、X は、M x によって署名された各証明書を無効にし、このような証明書を全て T から除去することができる。この場合、X は、T R 内に C A の失効証明書を記憶するべきである。本実装は他の有効性条件を ALGO 1 のステップ 1 に追加することに注目されたい。例えば、図 8 は、M 1 が証明書 M 1 R 2 を介して M 2 を無効にした際の証明書間の関係のマップを示す。

【0078】

G を用いて作業を実施する：G は今や、認証された方法で X からサービスをリクエスト

10

20

30

40

50

することができる。GがXを用いてアクションAを実施することを望んでいると仮定する。Aは、XがGに対して認証することを望むアクションである。すなわち、Aを実施するために、Xは、アクションが達成されることをグループGが望むということについての有効な認証を必要とする。

【0079】

1つの実装においては、AxがアクションAを権威付けるMxによって署名されたメッセージとなるように、Axを定義する。全てのAxメッセージから構成される集合AGを定義する。この場合、各Axは、対応する一意のMxによって署名される。例えば、M1はA1に署名し、M2はA2に署名する、等である。

【0080】

1つの実装においては、MxノードMInitは、G内の他の全てのMxノードから署名を照合することによってXとの通信を開始し、管理する。

【0081】

ALGO4と呼ばれる他のプロセスについて述べる。1つの実装においては、Xは、Gが以下の方法でAを実施するように権限を与えることができる。

- 1) Xが、MInitからAGをリクエストする。
- 2) MInitが、その非公開キーを用いてAxに署名し、AをG内の他の全てのMxに送る。
- 3) 各Mxの1個またはそれ以上がAxを作成し、これらの値をMInitに返す。
- 4) MInitは今やAGを保持している。AGは、N個以下の署名済みメッセージを含み、各々は一意のMxからのものである。
- 5) MInitが、AGをXに送る。
- 6) Xが、各Axを有効にし、コマンドが有効であり、署名が有効であり、且つAxに署名するMxがT内に存在することを確実にする。
- 7) Xが、一意なMxノードからの有効なAxメッセージの数を合計する。
- 8) 合計がT内の全N個の署名済み証明書の過半数である場合、Xはアクションを認可する。

【0082】

1つの実装においては、MInitは存在せず、各MxはAxを直接Xに送る。この場合、Xが控えめなMxから各Axメッセージを受け取った後、ALGO6を実行する。

【0083】

ALGO6と呼ばれる他のプロセスについて、図9を参照して述べる。この実装において、ALGO6は以下を含む。

- 1) 各Mxが、AxをXに送る。
- 2) Axを受け取る都度、Xは、それまでに受け取ったAxのそれぞれを検証し、コマンドが有効であり、署名が有効であり、さらにAxに署名するMxがT内に存在することを確実にする。
- 3) Xが、一意のMxノードからの全てのこのような有効なAxメッセージの数を合計する。
- 4) その合計がT内のMxSx証明書の過半数である場合、Xはアクションを認可する。

【0084】

証明書は、TおよびGCのような組またはグループ内で処理され、これらのグループ内の典型的な実装の証明書は、同じグループ内の他の証明書と比較した場合、同一であるコンテンツ部分を含む。例えば、G内の全てのMxに対して、所与のPに対して作成された全てのMxSp証明書は、Mpに対する同一情報を含む。この静的コンテンツは、認可されているアクション、または証明書署名者によって相互署名された他の証明書であってもよい。さらに、同じ概念が、コンテンツの署名者に対しても当てはまる。G内の所与のPに対して、Pによって作成された全てのMpSxがPによって署名される。この対称性のため、相互署名が生じる前に証明書が互いの知識を有する場合、上記プロセスの複雑性を実質上低減することが可能になる。すなわち、合意形成は同期して行われ、ノード間では

10

20

30

40

50

アトミックである。多くの典型的実装においては、G内にどのメンバを含むかについてノードが互いに同意しなければならないことが真実であるため、これは事実である。次に、簡素化されているが同等の方法の具体的な形態を示し、この署名および相互署名が単にG内の多数のM×間での正式な同意であることを示す。

【0085】

同期符号化を用いる1つの実装においては、各M×は、相互証明されるべきG内のM×を互いに知っている。Gを識別する1組の非公開キーを生成することが可能である。このキーリストは、各々のM×によって別個に署名可能であり、公開キーのリストおよびG内の各M×の署名を含む1個の文書に追加可能である。このことは、G内の各M×によってハッシュされ、署名されたGの許可メンバのリストを含む文書Dを生成し、図10におけるJSONフォーマットに示すように(全てのM×が署名されたと仮定して)同数のアイテムを有する第2のリストを生成する。1つの実装においては、Dは「クロック」整数値を含む。他の実装においては、現在の時間値は、「クロック」値として十分となり得る。

10

【0086】

いくつかの実装においては、満了、以前は有効ではない、等の他の有効性の値が存在し得る。

【0087】

この符号化は、上記非対称証明書モデルと比較して、かなり効率的である。7個のメンバを含むGに対して、非対象アプローチは49個の離散的な非対称証明書を生成し、ALGO1を介する処理の検証を要求する。同期符号化において、同一情報は、全ての自己および相互署名を含んで存在するが、1個の文書に対して7個の署名のみが必要となる。さらに、T、GCおよびGTはDのみを含む。さらに他の証明書は必要ない。

20

【0088】

1つの実装においては、ALGO1は、Dが存在すると仮定して、ALGO101と置換することが可能である。この場合、T=DおよびD=GCである。本実装においては、全てのM×メンバは、Dを置換する新たなDがT内に保持されるクロック値よりも大きい「クロック」値を有することに同意する。すなわち、より古いD文書を置換するべきD文書は、より大きい「クロック」値を有することになる。

【0089】

ALGO101は以下を含む。

30

1. D内の全ての一意キーをカウントし、2で割った総数をCOUNTとして定義する(切り上げ)。

2. 変数nを定義し、ゼロに設定する。

3. 以下の基準を満たす場合、D内の各署名に対してnの値を1だけ増す。

a) 署名が、D内のキーのリストに見出されるキーとマッチングすることを確実にする。

b) 署名が、D内のキーのリストに見出されるキーとマッチングすることを確実にする。

c) 署名が、D内のキーのリストに正しく行われたことを確実にする。

40

4. nがCOUNTよりも大きいか、またはnがCOUNTに等しく、さらにタイブレーカノードがステップ3を正しく完了した場合、継続する。そうでない場合は、Dを拒否する。

5. D内の「クロック」値がTの「クロック」値よりも大きい場合、継続する。そうでない場合は、Dを拒否する。

6. Dは、満了、以前は有効ではない、等を含む全ての有効性およびフォーマット要件を満たす。これらの1個が満たされない場合、Dを拒否する。

7. DをDと置換する。

【0090】

いくつかの実装においては、ALGO101はまた、ALGO2、ALGO3およびALGO4を置換可能である。すなわち、我々の同期符号化の場合、ノードのGへの追加ま

50

たは除去およびTの更新は全て、ALGO101を介して機能する。D内のキーのリストは、新たなM×個のノードに対するキーを含む、より多いか、またはより少ないキーを有することが可能であるので、ALGO101はTを変更し、さらにそれを検証することが可能である。いくつかの実装においては、「カウント」フィールドがG内のトポロジ変化につき常に1だけ増加する場合、Dメッセージのストリームは、任意のX内のTを、任意の有効な前バージョンから更新することが可能である。これは、送信されたDがXによって保持されるDよりも1だけ大きい「カウント」値を有するように、各DをXに送信することにより達成することが可能である。これは、任意の数のXにわたって、非常に簡単で的確なTの同期を許容する。例えば、図11は、全ての署名および有効性の要件が満たされていると仮定して、図10に見られるDを置換する。

10

【0091】

1つの実装においては、同期符号化を使用するようにALGO5を変更することも可能である。これらの場合、メッセージグループAGの代わりに1個の文書ADが送られる。ADは、署名リストおよびアクションAを含む。個々のAxメッセージの代わりにAD内の署名に対して、署名の検証が行われる。

【0092】

いくつかの実装では、たとえ少数であっても、M×ノードの一部の組が認可局として行動できるようにすることが有益である。整数の「定足数」フィールドをDに加えることにより、ALGO101に見られるCOUNT値の代わりに定足数値を使用することが可能である。すなわち、厳密な過半数を設定する代わりに、ノードの必要数を任意の値に設定することが可能である。例えば、Gが7個のノードを含む場合、定足数を2に設定可能である。この場合、ALGO101は、Dの有効な置換となるために、2以上より多い数のDに対する署名を要求する。

20

【0093】

いくつかの実装では、ある一定のM×ノードが他のノードよりも安全でない例を構成するために、サブグループGに分類されグループ化された複数のM×ノードを有することが有益である。この場合、「パケット」と呼ばれる図12に見られるものと類似のグループ化構文を定義できる。

【0094】

1つの実装においては、「パケット」GB×内の各グループは、ミニチュアグループGであると考えられる。すなわち、Gは、「定足数」フィールドおよび真であると評価するG内のGB×パケットの数に基づいてアクションを認可し、このような各々のパケットGB×は、その定足数値およびGB×内のM×の数によって定義される。さらに、これらのパケットの各々は、必要に応じてさらなるパケットグループを内部に含むことができる。これは、Gのフラクタル表現であり、G内の投票特性の任意のコントロールを可能にする。DをDの置換として認可するために、ALGO101が再帰的方法で実行され、その結果、各パケットは、そのパケットに対してALGO101を実行させる。例えば、図12に見られるDは、M4および署名に見られる(M1、M2、M3)の過半数を有するDからの置換を許容する。Dはまた、(M1、M2、M3)の過半数および(M5、M6、M7)の過半数によって満たされる。

30

40

【0095】

この実装は、特定の性能特性、負荷分散、安全性、および分配特性に対して微調整されるGのトポロジを、厳密な多数決システムを用いて可能となる以上に許容するという利点を有する。関連する認可セクションで見られるように、この原理は、包括的認可代数学にさらにまとめることが可能である。

グループコマンドおよびコントロールを用いるグループ権限の作成

【0096】

グループコマンドおよびコントロール機構の様々な実装について述べたが、この機構は、グループ権限(GA)と呼ばれる概念の作成に適用することも可能である。ENTにおいて、GAは、従来のPKIシステムにおける認証局(CA)と同等である。いくつかの

50

実施形態によれば、G Aは以下のように描写され得る。

【0097】

最初に、PKIシステムPKおよびユーザUxを定める。Uxは、システムの任意のユーザである。U1はユーザ1等である。Uxによって作成される非公開/公開キー対Ux PPKを定義する。この場合、Uxは非公開キーを保持する。例えば、U1のPPKはU1 PPKである。Uxは、PK内の証明書を管理することが義務である証明書コントロールオフィサを表す場合もある。Uxがこのようなオフィサである場合、U1 Pは、Uxによっては作成されそうにないであろうが、発信側ユーザによっては作成される。この場合、以下の実装に影響を与えないので、Uxの代わりにオフィサを用いる。暗号プロセスCALを定義する。これは、使い古された非対称暗号技術である。

10

【0098】

N個のノードのグループGを作成する。「グループコマンドおよびコントロール」セクションにおいて上で概説した機構を用いることにより、証明書ストアGCを作成する。それらの規則に従うと、Gは1個のエンティティとしてコマンドを発行することができる。G内の任意のノードMxを定義する。この場合、各々のMxは、GC内の自己署名済み証明書および相互署名済み証明書を有する。例えば、M1はG内のノード1である。

【0099】

各ノードMxは、Ux PPKの公開部分を含む証明書に署名することが可能である。このような各々の署名済み証明書は、MxがUxにつき1個以上の固有署名済み証明書を作成しないように一意の相手Uxを識別するものとして証明書が定義できるよう、証明書フィールドの一意の組み合わせを含まなければならない。この証明書をUxCとして定義する。1つの実装において、このような全ての証明書は、グループコマンドおよびコントロールセクションにおいて、同期方法につき1個の文書UCに組み合わせることができる。

20

【0100】

1つの実装において、当業者は、X.509x標準に存在するフィールド、例えば組織、下位組織および共通名等の共通するものを選択することができる。

【0101】

1つの実装において、証明書内の独自の情報は、特定の一意的数値を含むフィールドになり得る。この場合、全てのUxは、証明書を介して、無名数として定義される。

【0102】

1つの実装においては、Uxは、ALGO1を用いるG内の各Mxから証明をリクエストできる。

30

【0103】

ALGO1と呼ばれる他のプロセスについて、図13を参照して述べる。本実装においては、ALGO1は以下を含む。

1. Uxが、Ux PPKを作成する。
2. Uxが、Ux PPK内の公開キーとUxに対する一意識別子Iを含む証明書リクエストUxRを作成する。
3. G内の各Mxは、UxRを受け取ったときに以下のアクションを実施する。

a) Mxは、UxR内の一意識別子を含む証明書を作成していないことを検証する。その時点で次のMxに対してステップ3を実行していた場合、ユーザに証明書を戻さない。

40

b) Mxは、Ux PPKの一意識別子および公開部分を含む新たな証明書UxCを作成し且つ署名し、この証明書をUxに戻す。

【0104】

Uxは、全てのMxによって作成されたUxCからなる証明書の組UxSを有する。この場合、各々の証明書は、一意のMxによって署名される。UALGO1のステップ4の出口は存在しないと仮定すると、UxSはN個の証明書を含む。

【0105】

グループGは、従来のPKIシステム内において、CAに対する置換として使用可能である。

50

【0106】

U x は、P K のいずれかのユーザに接触し、U x P P K の非公開部分を用いて認証することができる。X を、P K に対して U x の認証を望む U x によって接触された相手方として定義する。この場合、P K は G を G A として使用する。U x A を、U x によって X に送られる署名済み認証として定義する。一般に、X はランダム値を U x に送り、U x は署名済みメッセージ U x M を用いて応答する。この場合、U x M は U x の非公開キーによって署名されたものである。

【0107】

A L G O 2 と呼ばれる他のプロセスについて述べる。本実装においては、A L G O 2 は以下を含む。

1. X が、U x S をリクエストする。
2. X が、U x M が適切な署名および情報を含むことを検証する。そうでない場合は、認証は失敗である。
3. U x S 内の各 U x C に対して、X が、U x C が G のメンバによって署名されたことを確認し、U x C が有効であることを宣言する。オプションで、X は、同様に他の型式の有効性、例えば、有効開始日および有効終了日等を任意にチェックすることとしてもよい。
4. X が、全ての有効な U x C 証明書の数を含む合計 S を生成する。
5. S が、N が G 内のノードの数であるとして $(N / 2) + 1$ よりも大きい場合、X は U x を認証している。

【0108】

G 内の J 個のノードがアタッカによって占拠されていると仮定する。これは、J 個の M x 署名が信頼出来ないことを意味する。しかしながら、J が $N / 2$ よりも小さい限り、システムは安全である。アタッカがなりすまされた証明書を用いて U x S を作成する場合、U x S は J 個の証明書のみを含む。このことは、アタッカが、A L G O 2 のステップ 5 における認証に合格するのに必要な過半数を得ることを妨げる。

【0109】

1 つの実装において、X は追加的に、U x C に署名する各 M x が G 内の他の M x の過半数による有効な相互署名を有することをさらに検証する。もしそうでない場合は、その M x によって署名された U x C は、無効であるとみなされる。さらに、X による A L G O 2 の将来の繰り返しは、この情報を用いて、M x を完全にディスカウントすることもできる。M x がディスカウントされた場合、X によれば、G 内のノードの総数は $N - 1$ となる。

【0110】

信頼レベルという概念を定義する。これは、X を成功裏に検証し認証する G 内の認証ノードの割合を意味する。これは $(S * 2) / N$ の値である。信頼レベルが 100% よりも高い場合、システムは安全であり、このようなトランザクションを完全に信頼できると定義できる。100% よりも高いレベルは、追加の値を有さないことに注目すべきである。したがって、200% の信頼レベルを生成する G 内の全ての M x に対する完全な認証チェックは、100% よりも高い信頼レベル以下の値を本質的に有する。

【0111】

タイブレーカノードを使用し、タイブレーカノードが適切に認証され、 $S = N / 2$ である場合、信頼レベルに明らかな過半数を与えるために S に 1 を加算する。

【0112】

基本的には、 $(N / 2) + 1$ 個よりも多い M x の任意のエンティティ収集コントロールがシステムをコントロールする。そのエンティティは、それによってコントロールされていない M x が過半数の M x によって無効化されたその相互署名を有するように、G を変更することが可能である。この場合、N は、適切な相互署名を有する M x の数となり、これはエンティティによってコントロールされた M x のみになる。従って、100% よりも高い信頼値は完全な信頼を表す。

【0113】

1 つの実装において、X によるある種の動作は、信頼レベルにより制限され得る。例え

10

20

30

40

50

ば、20%の信頼レベルは、低セキュリティデータに対する読み出し専用の情報アクセスを許容し、50%のレベルは、メッセージ送信のようなクリティカルでない相互作用を許容するとともに、Eメールがチェックされることを許容し、100%以上のレベルは、送金、PK方針の変更等のクリティカルなトランザクションに対して使用される。

【0114】

各M×証明書がCAによって署名される1つの実装において、Xは、Mによって署名された各U×CがそのCAによって正しく署名されたことをさらに検証することができる。もしそうでない場合は、U×Cは無効であることが宣言される。

【0115】

1つの実装において、Xは、Tと呼ばれる集合の内部にGの部分集合を維持し、Gの代わりにTに対して検証を行う。Tは、GのN-1個以下のノードを含み得る。

10

【0116】

1つの実装において、Xは、有効なM×に対して1個のU×Cのみを検証する。U×Cが有効である場合、XはU×を認証している。この場合、Xの信頼レベルは2/Nである。例えば、Nが3である場合、信頼レベルは66%となる。この動作モードは、G内のただ1個のM×を占拠したアタッカがXに対して有効となることを許容するため、推薦されない。しかしながら、この動作モードが、金融取引、クリティカルなデータの交換、ならびにコマンドおよびコントロールのためにTLS/SSL接続を認証するための全てのウェブブラウザにおける現存の最新式機構と、セキュリティ上同一であることに注目する価値がある。

20

【0117】

ENTは、グループ権限構造を実現することもできる。このGAは、異なる非対称キープロセスを実行する異なる場所にあるサーバのグループから構成される。このようなサーバの各々は、ルートと呼ばれる。各ルートは、他の全てのルートから独立して署名および動作を行う。各ルートは、連続する文字の集合として定義された一意な名前を有する。同時に、ルートは、グループコマンドおよびコントロール並びにグループ権限セクションに見られるプロセスを用いて、新たなベリニムを発行することができる。好ましい実装は、奇数個のノードを有するGAを作成して、デッドロックを回避することである。ENTにおいては、グループコマンドおよびコントロール内のALGO1において計算された過半数を作成する全てのルートに対する署名済みおよび自己署名済み証明書の集合は、ルートリングと呼ばれる。1つの実装において、これらの証明書は、グループコマンドおよびコントロールセクションにつき1個の文書に同期して組み合わせられ得る。

30

【0118】

ENTルートは、破棄される場合もある。これは、ENTシステムの安全性または信頼性を低減するいずれかの理由のために発生する可能性がある。いくつかの潜在的な原因は、不良コンピュータハードウェア、悪意ある攻撃、監査の失敗、自然災害、計画されたルートの老化等である。ルートは、グループコマンドおよびコントロールセクションにおいて概説された機構を介して破棄される。ルートが破棄されるべき場合、ルートリング内の他の各々のルートサーバは、破棄されるべきルートをターゲットとする破棄証明書を生成することになる。過半数のこのような証明書が一意のルートによって発行されると、問題のルートは、グループコマンドおよびコントロールにおいて概説された機構により、ルートノードの連結グラフから除去される。これらの証明書を受け取るENTシステム内のユーザは、同様に、破棄されたルートノードをその信頼性ストア(T)から除去する。

40

【0119】

1つの実装において、ルートノードはそれ自体を無効化することができる。この場合、システムは、他のルートによる過半数の無効証明書と同等であるとして、その無効化を処理する。すなわち、それ自体を無効化するルートは、過半数の他のルートがそのルートに対する無効証明書を生成したかの如く処理されなければならない。様々な実装によれば、両方の機構を使用することが推薦される。

【0120】

50

ENTルートもまた、加算可能である。これは、システムが成長する必要があるとき、または破棄されたルートが置換される必要があるとき、発生する。ENTは、グループコマンドおよびコントロールに見られるノードを加えるための機構を利用して、このタスクを成し遂げる。まず、新たな各ルートに対するPPKが生成される。その後、公開キー部分が、ルートリング内の各ルートに送られる。ルートリングルートの各々は、その後、新たなルートの公開キーを含むこの新たなルートに対する相互署名を生成する。これらの証明書は、その後、グループコマンドおよびコントロールセクションのALGO2に概説された機構を介して、各ルートの信頼性ストア(T)に加えられる。

ENTシステム状態および信頼性ストア

【0121】

10

ENTルート信頼性ストアは、理想的には、全ENTネットワーク内の(ENTソフトウェアを実行する)あらゆるENT可能化システム上に記憶される。これらの装置は、ENTノードと呼ぶことが出来る。ENTシステムの全てのユーザが、別個の信頼性ストア(T)を維持することが有益である。これによって、部分的に、または散発的に接続されたENT装置が、ENTシステムのルートまたは他の部分がたとえ到達不能であっても、動作を有用に維持することが可能になる。さらに、アタッカは、システムの有効性に全体として損傷を与えるためには、システム内の多数のノードを打ち負かすことが必要になる。ルートノード無効化および新たなENTルートノードが処理されるので、これらの証明書は、グループコマンドおよびコントロールのALGO1(またはALGO101)が決定論的かつ冪等元となった後、ENTシステム全体が更新された同一の状態を有するようになるまで、ENTノード間で伝播され得る。

20

【0122】

1つの実装において、ENTノードの信頼性ストアは、通信または情報交換を行う度に同期され得る。1つの実装において、ノードは、信頼性ストアT内の全ての証明書を、処理する前に交換する。これは、かなり大量のデータになる場合があるのであまり好ましくない。

【0123】

1つの実装において、ルートの自己署名済み証明書から暗号ハッシュが生成される。これらの証明書は、最初にオーダされる。決定論的オーダリングは、この要件を満たす。好ましい実装においては、オーダリングは、ルート名の標準アルファベット順オーダリングにより構成される。ルート自己署名済み証明書のオーダリストが一旦作成されると、数値ハッシュを生成するハッシング関数を介して各々のこのような証明書を順に挿入することにより、ハッシュが計算される。各ENTノードは、このハッシュを独立して作成し、結果的に生じる値を記憶する。

30

【0124】

1つの実装において、2個のENTノードが通信するか、または情報交換するとき、このハッシュをまず最初に交換することになる。このハッシュのミスマッチは、2個のENTノードに、その信頼性ストア(T)を交換するように強いる。グループコマンドおよびコントロールのALGO1(ALGO101)に示すように、全てのノードが交換された後に、信頼性ストアが再計算される。ALGO1(ALGO101)の1反復の後、双方のENTノードは、同一の信頼性ストア、従って同一のハッシュ値を有することになる。

40

【0125】

1つの実装においては、ルート証明書のアルファベット順のオーダリングが上記のように実施される。各証明書は、その後、ハッシュされる。このような各証明書のハッシュは、その後、順にデータオブジェクトに加えられる。実際、ハッシュ値は、共に順に連結され、ENTSTATEの値を生成する。さらに、ENTシステムバージョンを決定する値が、ENTSTATEに連結される。ENTシステムバージョンは、システムが使用する定数、許可されたプロセス等の情報を含むことができる。一旦計算されると、ENTSTATEは、(ENTSTATEにおけるその位置ずれにより)個別に認識可能な全てのルートを含むオブジェクトまたはデータ、さらにノード間の互換性チェックのために使用する

50

ることが可能なシステムコンテキスト値を反映する。このENTSTATEは、その後、ノード間で交換可能である。ENTSTATEのハッシュが、最初に交換され得る。これがマッチングしない場合、信頼性ストアの全体またはその一部（ENTSTATEにおけるルートハッシュにより決定される）が、両方の信頼性ストアがマッチングするまで交換可能となる。

【0126】

1つの実装において、ENTノードは、信頼性ストアの全体を受信するためにいずれかのルートに問い合わせることができ、加えて、他のENTノードからこの情報を受信することができる。

【0127】

1つの実装において、システムバージョンは、多数の値および設定によって定義可能である。いくつかの設定は、使い古された暗号プロセス、最小キー長等の決め打ちされた値、証明書命名構造やフォーマット等のポリシー設定を含む。好ましい実装においては、これらの値全てが1個のバージョン属性値によって決定され、これがシステムバージョンとして使用され得る。

【0128】

1つの実装において、ENTノードは、その信頼性ストアハッシュまたはシステムバージョン属性値が異なる場合、相互作用しない可能性がある。ENTノードは、取引を行う前に、その信頼性ストアおよびシステムバージョンのコンセンサスを得なければならない。システムバージョン上のミスマッチの場合、ノードは、その接続を終了するべきであり、より低いシステムバージョンを有するノードは、そのソフトウェアを更新するべきである。信頼性ストアミスマッチの場合、両ノードは、任意のトランザクションを継続できるようになる時点でコンセンサスに到達するまで、証明書を交換するべきである。コンセンサスに到達できない場合、トランザクションを終了するべきである。

ルートノードによるベリニム発行

【0129】

ENTにおいて、ベリニムは、その一意番号またはマッピングに基づいて一意に識別可能であり、第1のルートによって発行されてリクエストを受け取る。他のPKIシステムにおいて、名前または記述的部分は、発行した証明書に対して使用される。典型的な値は、名前、組織、組織単位等である。しかしながら、好ましい実装において、ENTは、番号としての抽象的な識別子を直接発行することによって動作する。各番号は、一意であり、グループ権限セクションにおいて述べたようにシステム内で一意であることが保証されている。2個のルートは、2個の要求元に対して同一の識別子を発行できない。他の実装では、英数字を使用するか、または要求元が所与の識別子の値を選択することも可能である。

【0130】

ベリニムは、各々が一意なルートノードによって発行された1組の証明書として定義可能である。このような各証明書は、要求元が提出する公開キーや、システム内の一意番号を含む。従って、完全ベリニムは、Nはがそのベリニムに対して各々が一意識別子を含むENT内のルートノードの数であるとして、 $N/2 + 1$ 個以上の証明書を含む。本文書内の様々な場所は、ベリニム信任状に関連する。この用語は、ベリニム内で見られる証明書をいう。ベリニムおよびベリニム信任状は、コンテキストが暗号プリミティブまたは信任状に関連する場合、相互交換可能である。いくつかの実装において、1個の文書は、グループコマンドおよびコントロールセクションにより、多数の証明書と同じ情報を含むことが可能である。

【0131】

ベリニムの発行は、要求元（ベリニムをリクエストするユーザ）がPPKを作成し、コアルートのうちのいずれかにリクエストを提出する状態で開始する。リクエストは、PPK対の公開キーを含む。1つの実装において、リクエストはまた、以下で述べるピアノードのリストを含む場合もある。

10

20

30

40

50

【 0 1 3 2 】

1つの実装において、各々のルートは、要求元に割り当てる数値の所定のブロックを有する。例えば、ルート1はブロック値1 - 1 0 0 0を有し、ルート2はブロック値1 0 0 1 - 2 0 0 0を有する、等である。1つの典型的な実装において、これらのブロック範囲は、32ビットブロックとなり得る。ブロックに対する許可を得たルートのみが、与えられたブロック内の番号を割り当てることができる。他のルートブロック内の番号の割り当ては、セキュリティ違反として処理するべきである。ENTの中央局は、どのルートがどのブロックを発行可能であるかを指示する。好ましい実装において、そのブロックの全てを発行したルートは、無効化され、新たなブロック上での発行制御を有する新たなルートと置換されるべきである。他の実装においては、そのブロックの全てを発行したルートに対し、中央局によって新たなブロックを割り当て可能である。好ましい実装において、ルートは、以前は発行されていないその有効ブロック内で選択された連番を、要求元に対して発行する。他の実装において、ルートは、その有効ブロックから乱数を発行することができる。

10

【 0 1 3 3 】

ルートが、そのブロックから、まだ割り当てられていない数値NVを一旦選択すると、ルートは、要求元の公開キーおよびNVを含む証明書を作成し、それに署名する。この証明書は、その後、ENTシステム内の他の有効ルートのそれぞれへ送られる。これらの他のルートノードの各々は、NVを含む証明書をまだ発行していないことをまず検証し、その後、同一要求元の公開キーおよびNVを含む証明書を作成し、それに署名する。この組の証明書は、その後、完全ペリニムとして要求元へ戻される。実際、要求元は、新たに作成された証明書をルートが預け入れるであろうデータストアをチェックするであろう。要求元は今や、いかなる目的にも応じた有効ペリニムを有する。

20

【 0 1 3 4 】

一旦発行されると、ペリニムに対する非公開キーが失われる、盗まれる、または無効化される可能性がある。ある時点で、非公開キーは、所与のペリニムに対して置換される必要がある。これらのイベントが発生する際、コントロールする非公開キーが失くなるとコントロールが失われるので、ペリニム所有者がペリニムのコントロールを再構築するための機構を設けることが重要である。従来PKIシステムにおいては、これは、問題のユーザの識別を再構築するためにいくらかの人材の関与を必要とする。ENTにおいては、任意のPKIシステムに適用可能なリレーショナル認可と呼ばれる新規な技術を用いて、プロセスが自動化される。

30

リレーショナル認可：

【 0 1 3 5 】

以下のセクションでは、ユーザがPKIシステム内の自身の信任状または非公開キーのコントロールを失ったときにPKI信任状のコントロールを（置換によって）再構築する方法、または、他の信任状の相関的な使用に基づいて信任状のアクションを認可する方法について述べる。これらは、オーナーシップポリシーおよび1つ以上のコントロールポリシーとして、それぞれ考えられ得る。上位概念、すなわちダビングされたリレーショナル認可は、エンティティが、自身で定義するピアグループに基づき、オーナーシップを再構築し、エンティティに対する新たな署名済み信任状を作成するために証明書権限又はグループ権限とともに使用され得るそれらのピアの署名又はパウチャを要求することを許容しなければならない。さらに、同じ概念が、それ自体の信任状を有する1組のピアエンティティが、そのエンティティに対するアクション（またはコントロール機構）を裏付けることを可能にする。

40

【 0 1 3 6 】

第一に、リレーショナル認可は、エンティティ自身によって要求されるものを越えてピア内に存在するための特定の特性を必要としないことに注目されたい。すなわち、それは、匿名であり且つ非公開である。第二に、CAは、現時の最新式機構に基づいて更新を実施する必要がもはやなく、これは常に集中アプローチとなる。最後に、マンパワー、管

50

理またはプロシージャに関してCAには何も要求が出されず、事実上、CAは、抽象的識別を超えてユーザ/エンティティの情報を管理する必要が全くないことに、全体を通じて注目されたい。

【0137】

このセクションの大半は、信用状の更新のためのリレーショナル認可の利用に特に集中するものであるが、概説した発明は、認可されたアクションが望ましく、多数の信任状がその認可に対する入力として望ましい時に、より広範囲に亘る適用可能性を有する。例えば、リレーショナル認可は、信頼性に対するアイデンティティのコントロールを証明し、データに対するアイデンティティアクセスを可能にし、または何らかの目的のために、信任状内の公開キーを、協力して作業する多数のエンティティから構成されるポリシーと置換することができる。

10

【0138】

N個のエンティティ/ユーザのグループから構成されるPKIシステムPKを定義する。ユーザは、人間、コンピュータ、モバイルデバイス、または信任状を記憶且つ使用可能にする他の電子的可能化システムであってもよい。CAを、PKの証明書権限として定義する。CAは、グループ権限(GA)であってもよい。各エンティティをU1ないしUNとして定義し、さらにCAにより署名されたその信任状をC1ないしCNとして定義する。この場合、U5はユーザ5を表し、C5はユーザ5の信任状を表す。各エンティティの非公開キーをP1ないしPNとして定義し、この場合、P5はエンティティ5の非公開キーであり、さらにPmを任意のエンティティの非公開キーとして定義する。Uxを、Cxを新たな証明書Cxと置換する必要があるエンティティとして定義する。M個のエンティティでグループGを定義し、この場合、各エンティティは、PK内に存在し、Uxとの実在の、または帯域外の関係を有する。それらのエンティティのうちの任意のものをUmとして定義する。Umに対する信任状をCmとして定義する。例えば、3個のエンティティUq、UzおよびUyでGを定義し、各々はCq、CzおよびCyをそれぞれコントロールする。図14を参照されたい。

20

【0139】

1つの実装において、Gを含むデータオブジェクトLおよびポリシーステートメントSを定義する。Sは、ブール値を出力として提供するステートメントに、Gのメンバを組み合わせるための1組の処理ルールから構成されるポリシーステートメントである。例えば、Sは、連続的な文字リスト「(UyおよびUq)または(UzおよびUy)」を含むことが可能である。さらに他の非ブール値、例えば「過半数(Uy、Uq、Uz)」または「(Ua、Uy、Uq、Uz)の2」等が作成され得る。Sは、CAがUxのためのキーの取り換えを可能にする際の基準を定める。有用なルールは、基本ブール演算子(OR、AND、NOT)、グループ化ステートメントに対するオーダリング、および関数を含む。任意の数の異なる関数がサポートされ得るが、関数は、真偽を示す値を戻さなければならない。Sの構文は、信任状フォーマットおよび具体的な実装に大きく依存するが、XML、JSON、文字列または他のバイナリフォーマットから構成することが可能である。これらのステートメントは、もしステートメント内の任意のUmが「真」の値と置換されるのであれば、真偽を評価することができる。デフォルトでは、全てのこのような値は偽であるとみなされる。要するに、Um値は、Umがアクションを認可するパウチャに署名した場合、または、Cmが自身で真であると評価したポリシーSmを含む場合、ALGOYを介して「真」の値と置換される。

30

40

【0140】

他の実装において、S内の演算子のグループ化は、先行値を有することができる。この値は、入力に関する優先度を設定することができる。例えば、「(UxおよびUy、101)」は、ステートメントが優先度101を有することを表すことができる。1つの実装において、真であると評価するより高い優先度ステートメントは、真であると評価するより低い優先度ステートメントより優位に立つ。いくつかのUmエンティティがアタッカに

50

よって危険にさらされていることをもしアタッカが獲得するならば、優先度は有用となり、アタッカは、Sを介して真の値になりすまることができる。この場合、アタッカがコントロールを持つ、有効であるがなりすまされたエンティティよりも、より安全な組のUmを優先させることがCAにとって有益である。これによって、Sの代数は、ポリシー内でさえ、コントロールの階層を含むことが可能となる。この場合、CAは、最後に認可されたアクションの優先度の値を記憶する。より高い優先度のアクションが後に発生した場合、CAは、新たな相互作用を許可し、前の相互作用を破棄することができる。

【0141】

いくつかの実装において、より高い優先度の信任状のリセットは、特定の期間、より低い優先度のリセットを無効にすることが出来る。例えば、2週間である。これは、アタッカがその期間、再度ポリシーをリセットすることを許可せず、異なる認可グループの間でポリシーがばたつくことを防止する。

10

【0142】

他の実装において、データオブジェクトLはGを含む必要がない。なぜなら、この情報もまたS内に存在するからである。

【0143】

典型的な実装において、認証は、一般に非公開キーを用いて直接行われる。しかしながら、リレーショナル認可を用いると、信任状は、代わりにSを介してピアのグループに対して認証されることができる。すなわち、Sは、信任状内の公開キーを取り換えることが可能である。例えば、信任状が組織を表し、その組織が秘密のデータアクセスを必要とし、3人がそのアクセスを認可するために必要とされた場合、特定のPPKを必要とする組織の代わりに、その認証プロセスを満たすためにリレーショナル認可を使用することができる。一例として、ある人の信任状は、銀行口座にアクセスするために、スマートフォンおよびキーホルダーサイズのデバイスの両方を用いることを求められるかもしれない。電話とキーの両方が公開キーを含み、協力して、その個人が自分の信任状においてSを介して銀行口座にアクセスすることを可能とする。この場合、信任状は、公開キーの代わりに、認証に対するポリシーを含む。公開キーは、ポリシーS内の当事者によって保持される。明白にするために、Pmは、当事者Xを含むポリシーステートメントSと置換され、Xが所有するPxは、非公開キーであってもよいし、他のポリシーステートメントSxであってもよい。このような場合、ネスト化されたSxステートメントがループを形成しないことは極めて重大である。例えば、U1が「U2」からなるS1を有し、U2が「U1」からなるS2を有する場合、いずれのステートメントも決して真であると評価出来ないことになる。その理由は、いずれのステートメントも、ステートメントSmの部分で真であると設定できるPPKからの入力を有しないからである。いくつかの実装において、このルーピングは、深さ基準を用いて制限することが可能であり、その結果、プロセスが終了する前にはある一定の整数回の再帰のみが許可され、偽の戻り値が戻される。いくつかの実装において、ポリシーSは、少なくとも1つのパスが真の値を戻す場合にのみ、正当であるとされてもよい。

20

30

【0144】

トークン化プログラムについて述べる他の文献に含まれることであるが、ステートメントSを介してコントロールされたU1を含むステートメントSが、U1がS内のどこに現れるとしても、U1をSと置換することによって拡大できることには、注目する価値がある。例えば、Sが「U1およびU2」であり、SがU1に対して存在し、「U4およびU5」から構成された場合、Sは「U1」をSと置換し、「U4、U5およびU2」を生成することが可能である。

40

【0145】

1個のLには、各々がシステム内の特定のアクションまたは権限に対応する多数のポリシーが存在し得る。例えば、アクセス、認証、更新等である。いくつかの実装においては、オーナーシップポリシーがCAを用いて作成および管理され得るのと同じ方法で、任意の数のこのようなポリシーが、ユーザによって作成、分配、管理され得る。JSONフォー

50

マットにおける信任状の例として図 15 を参照されたい。ある一定の実装においては、これらのポリシーステートメントは、CA によって発行された信任状内に存在する可能性がある。他の実装においては、これらのポリシーステートメントは、それら自身が署名し且つ認証済みであり、CA によっても署名されたメッセージ内に存在することとしてもよい。

【0146】

典型的実装において、ポリシーは、図 15 に示すように、ALGOY を介して置換可能な信任状内に存在する。すなわち、これらの（グループとしての）ポリシーは、1 つの信用状更新プロセスを介して置換される。いくつかの実装は、これらの別個のポリシーの更新を個々に分離することを望む可能性もある。このような場合、ALGOY が存在し、このような各々のポリシー宣言に対して実行される。この機構は、一般に思い浮かぶものよりも、「アイデンティティ」のさらに分散された概念を提供し、管理コストの増加、複雑さの増大等を含む実質上の二次的効果を有する可能性がある。

10

【0147】

1 つの実装において、CA は、公開キーを提示する要求エンティティに対して、一時的な信任状を提供する。これらの信任状は、決して繰り返されない単に増大する数値を含む。発行されたこのような証明書は、この番号および関連付けられた公開キーにのみ基づく、その識別メトリックにおいて互いに異なる。さらに、これらの信任状は、他の目的のためには使用出来ないように、システム内で明確にマークされる。いかなるユーザも、いつでもこのような証明書をリクエストすることができる。例えば、何人かのユーザは、連続の 1000 を含む証明書をリクエストすることができる。このような証明書をリクエストする次のユーザは、証明書 1001 を受け取る、等である。同じエンティティが、任意の数の証明書をリクエストすることが可能である。この一変形例は、このような証明書がエンティティ情報を含むことを可能にすることである。この情報は、Ux が要求元である場合、Ux のアイデンティティとマッチする。この証明書は、いかなる形でも、Ux を識別するために使用されてはならない。その目的は、安全でアドレス可能な方法で、任意の連番と公開キーとの関係を確立することのみである。

20

【0148】

Ux が新たな PPK キー Px を作成し、上記機構を用いて Px（公開部分）を含む一意証明書をリクエストすると仮定する。この一時的証明書を Tx と呼ぶ。Ux は今や、PK 内で認識される証明書を有し、それによって他のユーザがその連番を用いて Tx の一意性を検証できる機構を提示する。さらに、Tx は今や、Ux としてではなく PK 内の一意エンティティとして PK の他のメンバと共に認証するために、一時的容量で Ux によって使用され得る。一意性は、一意の連続番号によって定義される。

30

【0149】

Ux および Um が人々である 1 つの実装において、Ux は実世界における Um と接触し、Ux に対する再入力リクエストを CA に提示するようリクエストする。好ましい実装において、Ux は連番を言葉で Um に伝達する。他の伝達方法は、電話、言葉による対面、または音声付ビデオを含み得る。重要な基準は、Ux が新たなキーの必要性および Tx 内の連番を伝達し、Ux が Um に対しアイデンティティの確実な証拠を提供することである。このコンテキストにおけるアイデンティティの証拠は、Um が、Ux を Cx の正当な所有者として認識し、Ux が人であり、Ux が、Um が正当所有者であると考えた人であることを意味する。最良の解決策は Ux および Um の物理的ミーティングであり、2 番目に最良な解決策はビデオであり、3 番目に最良な解決策は電話である、等である。所有権及びアイデンティティの証拠は、強ければ強いほどよい。代替りの、またはサポートする機構は、DNA サンプル、指紋、またはある型式のバイオメトリクスを含み得る。これらの具体的な使用および手続きは、本文書の範囲にはない。しかしながら、意図は、Um が Ux を認識し、彼らが Ux の実社会におけるアイデンティティになりすますことにより Cx または Cx のコントロールを獲得しようとするアタッカではないことを決定できる、ということである。人々以外のエンティティが、本文書の範囲を超えて、本人確認識別基準

40

50

の異なるセットを使用するであろうが、これらは、共有の秘密、コンピューティングデバイスへの物理的アクセス等から構成され得る。

【0150】

1つの実装において、 U_m は、公開キーを除外した C_x 内の情報と、 T_x 内に見られる一意な連番とを含む U_x のための署名済み更新メッセージ RC_x を作成する。 U_m は、このメッセージを CA に送る。

【0151】

過度に単純化した実装において、 U_x が T_x 内に見られる公開キーをコントロールすることを任意の U_m が裏付ける場合、 CA は C_x を作成する。 CA は、 RC_x を受け取ったことに応じて、 C_x 内の情報が C_x 内の情報にマッチングすることを検証し、 C_x を作成すべきである。より形式的に展開されたこれらのステップは、図16のALGOXを参照すると以下のとおりである。

1. U_x が PK_x （またはポリシー A_x ）を作成する。
2. U_x が CA から証明書 T_x をリクエストする。この場合、 T_x は P_x （または A_x ）の公開部分を含む。
3. U_x が、 U_x のアイデンティティの実社会における検証を実施する U_m に接触する。
4. U_m が、 C_x 内のユーザアイデンティティ情報と T_x の連番とを含む署名済みメッセージ RC_x を作成する。
5. CA が、 RC_x 内の連番を T_x 内の連番にマッチングすることにより T_x 内の公開キーを抽出する。
6. CA が、 U_m の署名を検証し、 RC_x 内のアイデンティティ情報を検証し、その後、 T_x からの公開キーと、 RC_x 内のユーザ情報とを含む C_x を作成する。

【0152】

ステップ6において C_x を作成するための過度に単純化した機構は、多くの例で推薦されない。 PK 内の U_m のコントロールを得たアタッカが、システム内の他のいかなる C_x をも危険にさらすことは明らかである。より堅牢な機構が続く。 T_x が C_x の作成に必要なことにも注目されたい。これは単に有用であるに過ぎない。各 U_m は、代わりに、 U_x の識別情報と、 P_x の公開部分とを含む更新証明書を CA に提出することが可能である。公開キーは、 U_x と U_m との間の検証手続きの最中に U_m に与えられる。 T_x は、 CA に到達するために、公開キー部分のためのより自動的でヒューマンフレンドリな方法を単に提供する。

【0153】

U_x が C_x をコントロールする時に U_x によって署名され、 L と U_x の識別情報とを含むメッセージ RA_x を定義する。 U_x が、 PK 内の C_x を発行した直後、そして、 U_x が P_x のコントロールを失う前に、このメッセージを作成したことは極めて重大である。もし RA_x が作成される前に U_x が P_x のコントロールを失うと、この更新戦略の全体が失敗する。1つの実装において、 U_x は、 C_x を作成する最初の手続きの一部として RA_x を作成した。すなわち、 C_x および RA_x は、相前後して作成された。これは、 RA_x が存在しない期間を排除する。このことは、アタッカが、 C_x を更新して置換する U_x の能力を永久に破壊することを防止する。

【0154】

U_x は、 CA に RA_x を提示する。 CA は、 RA_x が P_x に署名されたこと、および CA によって署名された C_x に P_x が対応していることをチェックすることにより、メッセージを検証する。 CA は、その後、 RA_x を無期限で記憶する。このメッセージは、 PK のメンバがユーザ U_x のために代替の証明書 C_x の作成をリクエストすることを可能とするルールを定める。

【0155】

U_x は今や、各 U_m に接触し、先に概説したアプローチを用いて、 CA に再入力リクエストを提示することを要求する。このような各 U_m は、 U_x に対する情報を識別する署名

済みメッセージ RCx および Px に対応する公開キーを提示する。 Ux は、より少数のユーザが真のブル出力値を生成することを要求されていると S 内のルールが指示する場合、 M 人に満たない Um ユーザに接触する必要があるかもしれない。

【0156】

CA は、 G 内の異なる Um ユーザからいくつかの RCx メッセージを受け取る。 CA は、 CA によって署名された有効な証明書 Cm を使用して、各署名済みメッセージが PK 内の Um から発生したものであることを検証する。 CA はまた、各 RCx が同一の公開キーを含むことを検証する。もしそうでない場合、 CA は、受け取った全ての RCx を計算するべきであり、この公開キーは大抵の RCx においてマッチングする。マッチングしない RCx は、破棄されるべきである。

10

【0157】

多数の有効な RCx メッセージが、 Ux のための有効なキー情報を含む1個のピア x から存在することは可能である。例えば、ピアは2個のパウチャを発行することが可能であり、各々は、 Ux が2個の異なるキーを用いて更新パウチャリクエストをピアに2度送った場合、更新のターゲットとして異なる公開キーを含む。 CA は、これらのうちのどれを使用するべきかを決定する方法を持たない。この場合、 CA は、全てのピアから入力される全てのパウチャを、 Ux のための一意な公開キーによる組に照合する。全ての RCx メッセージの組に2つ以上の公開キーが存在する場合、 CA は各々に対する組を作成する。 CA は、その後、このような各組を処理する。オーナシップポリシーにおける基準を通して第1の組は、 Ux に対する新たな公開キーを決定する。認証に使用されるエンティティ公開キーが存在しないが、代わりに認証ポリシーが存在する場合もある。

20

【0158】

CA は、その後、 Ux に対する L 内の S をロードして実行し、出力値を計算する。出力値は、ステートメント S における各 Um に対する真の値を挿入することにより計算される。例えば、 S が「 $(Uy$ および $Uq)$ 」であり、 CA が Uy から有効 RCy を受け取ったが、 Uq からは何も受け取っていない場合、 S は偽の出力値を有する「(真および偽)」として計算される。計算の結果が偽である場合、 CA は何もしない。 CA が真の値を計算する場合、 CA は、 S に対して書かれた基準が正しく満たされていることを検証する。もしそうであるならば、 CA は Ux に対する Cx を作成し、更新は成功である。形式的に展開されたこれらのステップは、 $ALGOY$ と呼ばれ、以下の通りである。

30

1. Ux が PK 内の信任状 Cx を最初に獲得した時、 Ux がその後、データオブジェクト L を含む署名済みメッセージ Rax を CA に提示する。書名は、 Ux の署名、または Cx がキーの代わりにポリシー S を含む場合、 Rax を認可するのに十分な S 内の認可する所有者の署名でなければならない。

2. CA は、 Rax の署名および有効性を検証した。有効な場合、 CA は Rax を無期限で記憶した。

3. その後、 Ux は Cx (またはポリシー Ax) のコントロールを失う。

4. Ux が、新たな $PKPx$ (またはポリシー Ax) を作成する。

5. Ux が、 CA からの証明書 Tx をリクエストする。この場合、 Tx は Px の公開部分を含む。

40

6. Ux が、 Ux のアイデンティティの実社会における検証を実施する一意な Um に接触する。

7. Um が、 Cx (Ax) 内のユーザアイデンティティ情報と Tx の連番とを含む署名済みメッセージ RCx を生成する。

8. CA が、 RCx 内の連番を Tx 内の連番にマッチングすることにより、 Tx における公開キーを抽出する。

9. CA が、 Um の署名を検証し、且つ RCx 内のアイデンティティ情報を検証する。

10. CA がその後、 Rax 内の S を実行し、 Um が署名するか、または真であると評価されたポリシー Cm (Am) を有する限り、 Um の各場合を「真」と置換する。 Rax に署名した、または真であると評価されたポリシー Cm (Am) を有した G 内のあらゆる一

50

意 U m に対して、このステップを繰り返す。ポリシー評価は恐らく再帰的であることに注目されたい。

1 1 . S が真であると評価する場合、C A がその後、C x と同じ情報を含むが、代わりに T x に見られる更新済み公開キーを有する C x を作成する。

1 2 . C A がその後、U x のための前回の C x (A x) を無効化し、新たな C x (A x) を発行する。これは、C R L を用いて、また好ましくは、新規キーの破棄において概説した一意破棄プロセスを用いて生じる場合もある。その場合、各 R C x メッセージ M U S T が C x (または A x 内のポリシー情報) に見られる公開キーを含むことに注目されたい。そうでない場合は、C A は、所与の R C x メッセージがどの C x を置換することを意味しているのかわかることはない。存在しない場合、これは、アタッカがリプレーアタックを実施することを可能にする。

10

【 0 1 5 9 】

G が変化する場合、U x が R A x を更新可能であることが重要である。あるユーザがもはやユーザ内に存在しないこと、新たなメンバが G に加えられるべきであること等が可能である。このように、R A x は置換可能であるべきである。しかしながら、U x は、R A x を安全には交換できない。P x がアタッカにより不正アクセスされたと思ってみてほしい。U x が R A x を更新可能である場合、アタッカも同様である。アタッカは、R A x をアタッカに有益な R A x と置換可能である。その後、もはや P x のコントロールを有さないことに U x が気付いた場合、R A x はもはや U x のための信頼のグループを含まず、代わりにアタッカが R A x 内に配置したものを何でも含むのであるから、彼らは頼りになるものを持たないことになる。従って、R A x の置換は、他の機構を使用するべきである。

20

【 0 1 6 0 】

代わりに、R A x は C x または A x が作成されるのと同じ方法で置換可能であるべきである。U x は、その選択した新たな L を T x に加える。各 U m はその後、T x に対する連番を含む署名済みメッセージを作成して、これを C A に送る。C A はその後、A L G O Y ステップ 1 0 を用いて各メッセージを検証し、S の同じ結果を計算する。真である場合、C A はステップ 1 1 において、R A x を、L を含む R A x と置換する。C A はその後、R A x の使用を停止し、将来の更新全てに対して R A x ' を使用する。この手続きは、ステップ 1 1 において、C A が L を含む R A x と R A x を置換し、T x が L を含むということを除けば、A L G O Y に見られるものと同一である。

30

【 0 1 6 1 】

1 つの実装において、多数の R C x メッセージを C A に送られた 1 個のメッセージに組み合わせることは可能である。R C x メッセージ内のキー情報は、ユーザ識別情報であり、連番であるので、この情報は、1 個またはそれ以上の U m メンバによって署名された 1 個の文書に加えられる得る。多数の署名を有する文書は、その後、多数の個々の R C x メッセージの代わりに、C A に提示され得る。図 1 7 を参照されたい。

【 0 1 6 2 】

いくつかの実装において、R A x 置換 (修正された A L G O Y のステップ 1 2) を、ある所定期間、エスクロウに配置することが可能である。これは、アタッカが S 内の認可ノードのコントロールを獲得し、信任状オーナーに反応するための時間が与えられる前に信任状をリセットすることを防止する、この場合、C A は、ステップ 1 2 を即座に実施したり、R A x を公開したりしない。その代わりに、C A は、所定の期間、R A x を記憶する。いくつかの実装において、この期間は、期間をデータオブジェクト L に加えることによって信任状オーナーによって設定される。いくつかの実装において、期間は、システム全体にわたって固定される。この時間中に、C A は、S および U x 内の各認可エンティティに接触する場合があります。信任状リセットが保留されていることを、そのエンティティに通知する。代わりに、C A は、信任状がリセット保留中であることを述べる公開署名済みメッセージをポスティングすることが可能であり、U x および他の認可エンティティがその公開場所を定期的にチェックすることを可能にする。この手続きは、アタッカに、リセッ

40

50

トが保留中である様々な認可エンティティを同時に通知しながら、延長期間に多数の信任状を捕らえて保持することを強いる。それらのエンティティが各々、その個々の C x 信任状に対して更新機構としてリレーショナル認可を使用し、各々が他の認可エンティティと共に別個の R A x 更新信任状を有する場合、アタッカが、必要な期間、エンティティ C x の更新なしに、多数の信任状を乗っ取って保持する可能性は極めて低い。

【 0 1 6 3 】

人は今や、そのような設定が現存の P K I 更新手続きよりもどれだけ安全であるかを計算するために、セキュリティ比較を実施することができる。現在の技術水準での実装においては、単一障害点が常に存在する。セキュリティオフィサまたは更新プロセスを担当するグループのいずれかは、C A によって評価されている署名リクエストを作成する。これは、その後、適用される。しかしながら、セキュリティオフィサまたはグループの署名キーがアタッカによって不正アクセスされる場合、アタッカは、キーが破棄可能となるまで、新たなユーザ証明書の将来の作成へのアクセスを得る。

10

【 0 1 6 4 】

1 人の人間が、手続き上のグループまたはオフィサと同一のセキュリティコンテキストまたはセキュリティトレーニングを有する可能性はきわめて低いが、一方で我々は、順列計算を介して、協力して行動するあまり安全でない多数のキーを用いてグループやオフィサのセキュリティを越えることは難しくないということも知ることができる。例えば、セキュリティオフィサの署名に加えて 2 個の一意 U m エンティティからのさらに 2 個の R C x 署名を含むことによって、劇的な効果が生まれる。各 U m の信任状が C m の寿命の間に、飛躍的に高められた 5 0 % のセキュリティ侵害の機会を有したとすれば、セキュリティ全体は、セキュリティオフィサのキーのみの 4 0 0 % まで増大する。実際、このリストに (S 内のプール演算を用いて) 「 a n d e d 」 を追加した各々のさらなるユーザは、セキュリティをさらに 2 倍にまで増大させる。これは、各々のさらなるユーザに対して不正アクセスの確率が半分だけ低減する指数関数である。明らかに、このアプローチは、最新式の再入力プロシージャよりも安全である。さらに、リレーショナル認可原理は、認証、データアクセス、委任等に対する同レベルの増大したセキュリティを生み出すことを望まれる任意の組のコントロール特性に対して適用され得る。

20

E N T 内のリレーショナル認可 :

【 0 1 6 5 】

ベリニムのキーが無効化されるか、または不正アクセスされると、リレーショナル認可が、ベリニムのコントロールを再構築するために使用される。ベリニム (U x) のオーナーが、オーナーが信頼するピア E N T ユーザのリストを作成する。これらのユーザは、その後、そのオーナーのベリニムに対して更新ピアグループとなる。オーナーがベリニムのコントロールを失うと、オーナーは、そのピアグループ (G) に十分に接触し、ベリニムのコントロールを再構築する。どのピアがそして何個のピアがコントロールを再構築できるかを計算するための正確な方法はベリニムのオーナーに任せられ、(ステートメント S を介して) 定義される。これは、当然のことながら、ユーザが早期にオーナーシップポリシーを作成したことを意味する (R A x) 。

30

【 0 1 6 6 】

1 つの実装において、オーナーは、スクラッチから新たな E N T ベリニムをまず作成することにより、ベリニムのコントロールを再構築できる (T x を作成することと同等である) 。この新たなベリニムが一旦作成されると、それは、安全な転送および認証のために、他の E N T ピアと共に使用され得る。その後、このベリニムは、更新ピアに接触するために使用され得る。各ピアは、その後、ユーザが、更新されるベリニムの正しいオーナーであることを (音声またはビデオチャットを介して) 検証し、さらにその後、署名した裏書 (R C x) を作成することにより、その更新に対して保証することができる。この裏書はルートに転送可能であり、そのルートはその後、問題のベリニムに対する 1 組の証明書を再発行する。各ルートは独立して A L G O Y を実施し、さらにグループ権限セクションに見られる組み合わせ論を介して、その C x 証明書が U x に対する新たなベリニム信用状と

40

50

なることに注目されたい。

【0167】

ベリニムが発行された後、ベリニムの所有者は、署名済みオーナシップポリシーメッセージ（上記の R A x）をルートに提示するかもしれない。オーナシップポリシーメッセージは、ベリニムが新たな P P Kを含むように合法的に更新されることを許容するポリシーステートメント（上記ステートメント S と同じ）と、ピア更新メンバのリスト（上記オブジェクト L 内のリスト）とを含むベリニムオーナによって作成された署名済みメッセージである。

【0168】

更新パウチャメッセージ（上記 P C x メッセージ）は、所与のベリニムに対する更新ピアグループの任意のメンバによって署名されたメッセージであり、裏書としてルートサーバに提示される。

【0169】

ポリシーメッセージは、ターゲットベリニウム i d を含む更新パウチャメッセージをルートが受け取る度を実施されるブール式を含む。ブール式が真である場合、そのベリニムに対する新たな証明書をルートは発行する。この場合において、公開キーは、全ての有効な更新パウチャメッセージにおいてマッチングするキーである。

【0170】

リレーショナル認可により、オーナシップポリシー内のブール式は、変数、論理演算子、および真偽を評価する論理関数を含む。組み合わせは、ブールステートメントを形成する。各変数は、ベリニム i d である。受け取られた各々の署名済みの真正更新パウチャメッセージに対して、適切なベリニム i d は真の値と置換される。更新パウチャが存在しない状態でブール式が真であると評価される場合、ポリシーは無効であるとみなされ、維持されない。ブール式が、オーナシップポリシーが適用されるベリニム i d を含む場合、ポリシーは無効であるとみなされ、維持されない。

【0171】

オーナシップポリシーメッセージは、ブールステートメントと、ピアベリニムのリストとを含む。このピアベリニム i d のリストは、ブール式における変数として見出されるベリニム i d から構成されなければならない。ルートに送られた第 1 の有効なポリシーは、このような維持されたオーナシップポリシーのみである。その後のポリシーメッセージは（以下で述べるように、新たなポリシーを確立するために十分なパウチャが伴っていない限り）廃棄される。従って、オーナシップポリシーは、ベリニム証明書が発行された後、可能な限り好都合な方法でサーバに送られることが重要である。アタッカがユーザの非公開キーを簡単にハイジャック可能であり、オーナシップポリシーメッセージがまだ送られていない場合、ハイジャックは永久的であり、逆行できないものとなる。ベリニムの所有者がピア可能化オーナシップポリシーを欲しない場合、そのベリニム所有者は、ブールステートメントが常に偽であると評価するオーナシップポリシーをトランクに提示すべきである。

【0172】

現存のポリシーを変えるためには、以下の基準を満たさなければならない。

1. 現存のオーナシップポリシーのピアリストに見られるピアベリニムが、有効なオーナシップポリシーメッセージを提示できる。ブール式およびベリニム i d リストは、このような全てのメッセージに亘りマッチングする。
2. ブールステートメント内のベリニム i d 変数を真で置換する際に、現在のオーナシップポリシーブールステートメントが満たされなければならない。すなわち、真正のオーナシップポリシーメッセージがマッチしているベリニム i d から受け取られ、その後ブールステートメントが真であると評価されるならば、現存のオーナシップポリシーブールステートメント内の各ベリニム i d は、真の値と置換される。

【0173】

これらの基準を満たすことは、ターゲットベリニムに対して更新するように認可された

10

20

30

40

50

ピアグループもまた、ポリシーの変更を認可する、ということを確認する。それはまた、新たなブルステートメントが全ての関連するピアによって厳密に同意されることも確認する。この時点で、現存のポリシーは、新たなポリシーと置換される。

【0174】

いくつかの実装において、情報漏洩を防止するために改良がなされ得る。情報漏洩は、オーナーシップポリシーを調査する人にはベリニム更新ピアが可視であるという点で起こり得る。ベリニム間の続く関係を追跡することによって、ベリニム間の関係を推察するために使用する連結グラフを生成し、人々または機械に対する実社会マッピングを簡素化することが可能となる。このような情報を有するアタッカは、ベリニムの永久的コントロールを獲得することが可能になる1組のノード上で協調的な攻撃を理論的に計画することが可能である。これを防止する改良を実施することが可能であるが、システムが複雑になる。

10

【0175】

情報漏洩を制限するための1つの実装において、各オーナーシップポリシー（上記Lの内容）は、Lを作成するために、各ルートのPPKの公開キー部分を用いて暗号化される。一旦暗号化されると、ルートサーバのみがLの暗号化された内容を解読可能となる。他のいかなる外部関係者もLの内容を解読できない。ルートが更新パウチャを受け取ると、ルートは、Lの内容を解読してLを生成することが可能となり、その後、上記のようにSに対する値を計算することができる。

【0176】

1つの実装において、外部監査施設Aが更新プロセスを検証し監査できることは有益である。これは、AがLを計算できることを意味している。ベリニムのオーナーOはLを有する。その理由は、元々Oが、暗号化してルートに送る前にLを計算したからである。1つの実装において、Oは比較的非公開であるどこかの場所で単にLを維持している。好ましい実装においては、Oは、ルートからLをリクエストすることが可能である。この場合、Oによって接触されたルートは、LをLに解読し、Oの非公開キーでLを暗号化してLを生成し、このメッセージをOに送信する。Oは今や、その非公開キーを用いて、Lを解読してLを読み出すことができる。一旦何らかの機構を介してLを読み出すと、OはLをAに提示できる。Aは今や、ルートの公開キーを用いてLを暗号化することにより、Lを計算し、検証することができる。これによって、Aが獲得したものと同じLをルートが用いていることが確認となる。

20

30

【0177】

上記実装において、Aもまた、十分な監査を実施するために、ルートに提示された全ての更新パウチャへのアクセスを必要とする。Aは、これらの値をOから読み出すことができる。好ましい実装において、Aは、更新パウチャをルートから直接読み出すことができる。本実装において、ルートは、以前のように、Oに対するポリシーまたはベリニムを更新する。更新が一旦うまくいくと、ルートは、その後、全ての更新パウチャを1つのオブジェクトに並べ、そのオブジェクトをLで暗号化し、RVを生成する。ルートは、その後、RVを公開する。監査人Aは今やRVを読み出すことができ、Lを用いて、ポリシー置換またはベリニムキー置換において使用される全ての更新パウチャを読み出すことができる。Aは今や監査を十分に行って、ベリニムキーを更新するか、または現存のオーナーシップポリシーを置換することについての許可をルートが有したことを確認にすることができる。リクエストに際し、Oに対して監査の失敗が増大する可能性もある。その後、正しい監査情報が存在しない場合、ルートが危険にさらされることが確定する可能性もある。

40

【0178】

1つの実装において、Lは、Lを極めて一意とする乱数または乱数値を含むことができる。例えば、128ビットの乱数値をLに加算する。これは、アタッカがL内のベリニムidに対する潜在価値およびSのフォーマットを推測し、試行錯誤によってLを再構築することを妨げる。

【0179】

好ましい実施形態では、ベリニムに対する信任状の新たな置換組が作成されると、現存

50

の信任状が無効化される。これは、現存のPKIシステムにおいて使用可能であり、以下で述べるキー失効に対する新規なアプローチを介して成就される。要するに、最新の作成タイムスタンプを有する所与のルートによって署名された有効証明書は、有効証明書であるとみなされる。より以前の日付のタイムスタンプを有する同一のペリニムidを含む他の全ての証明書は、無効であるとみなされる。

新規なキー失効

【0180】

新規な改良点を説明するために、適切なコンテキストを提供することが必要である。証明書権限Aと、データ記憶（ディレクトリと呼ばれることが多い）Dと、ユーザUとを有していると想像されたい。Uは、Aから新たな証明書をリクエストすることを認可されている。Uは、非公開/公開キー対（ p_x 、 p_y ）とともに非対称キーKを作成する。Uは、Aが署名し認可した p_y を含む証明書Cを作成することを望む。

10

【0181】

1つの実装において、Uは、Cをリクエストする前に、以下のステップを実施する。

【0182】

他のより典型的な実装において、Uは、Cをリクエストした後に、以下のアクションを実施する。

【0183】

図18を参照して、手続きALGO1について述べる。

1. Uは、非対称キー1ないしNの組KEYSを作成する。この場合、nは証明書更新がAから必要とされる前に仮定された、Cの寿命にマッチングする任意の数である。値nは、任意の時間増分に基づいて決定することも可能である。例えば、証明書更新間の期間が1年であり、増分が1日である場合、 $n = 366$ である。これは、区間ごとに、すなわち日ごとに、1つの証明書を提供する。代わりに、Nはスペース、送信、または他の必要条件に基づいて決定することが可能であり、間隔は番号Nから得られる。値Nは1よりも大きくなる必要はなく、この場合、KEYSは1個のキー対のみを含む。Uは、その後、証明書1ないしNを含む組Sを作成し、キー対KEYS[x]を用いて証明書S[x]を作成し、各証明書は、証明書チェーンC → C が有効な証明書チェーンになるように、Cによって署名される。S内の各証明書もまた、証明書「連続」フィールドに一意的値を含む。一般に、この値は1ないしNとなり、この場合、S内の証明書1はシリアル値を有し、証明書2はシリアル値2を有する、等である。

20

30

2. Uは、 p_x とともにS内の各証明書に署名する。

3. 1つの実装において、Uは、例えば「N TERMINATE」のシリアル終端値を含む最終的な証明書Fを作成する。代替の実装は、異なる証明書値、または、証明書増分の終了を表すいくつかのトークンを含むシリアル値に対する異なるテキスト値を使用する。すなわち、それは、全ての証明書を見る人が、Nよりも大きい値を有するS内の証明書が存在しないことを決定できるように、Sの組を終了させる。他の実装において、Fは作成されない。

4. Cを作成する任意のリクエスト（以前の場合）および全ての上記ステップが達成された後、Uは、 p_x および p_y を含むキーKを破壊する。

40

【0184】

Kは今や、回復不可能である。アタックは、これらのステップが行われた機械にアクセスしない限り、Kにアクセスできないか、またはS内のキーに類似の、または対称の追加のキーを作成できない。Uは今や、数値的に増分されたS内のN個の証明書のリスト、ならびにSの組のサイズを明白に示し、さらに終了を明白に示す情報を含む証明書Fを有する。さらに、Uは、これらの証明書をいかなる他の者ともまだ共有していない。それらは、局所的に作成され、Aは必要ではなかった。

【0185】

N個のオブジェクトから構成されるCERTの組を定義する。ここでは、各オブジェクトは、1とNの間の各xのための対（S[x]、KEYS[x]）を含む。

50

【0186】

1つの実装において、Uは今や、CERT内の各オブジェクトおよび非公開キーPを有する証明書Fを暗号化し、1組のサイズNの暗号化オブジェクトであるPSの組を生成する（各々は、証明書または証明書/KEY[x]対）を含む）。Pは、Uにのみに知られたパスワードである。

【0187】

他の実装において、UはCERT内の各オブジェクトをJ個の部分に分け、J個のキーを用いて暗号化する。これらのキーは、非対称または対称キーであってもよい。非対称キーの場合、1つの実装は、ピアユーザの証明書を用いて各部を暗号化することである。このピアグループをTと呼ぶ、このような場合、PSは、1組の集合から構成され、各々は、これらの暗号化されたオブジェクトを含み、各部分集合はJ個の部分から構成される。

10

【0188】

1つの実装において、Uは、保存のためにディレクトリにPSを転送する。

【0189】

1つの実装において、Uは、保存のために他のピアユーザにPSを転送する。

【0190】

1つの実装において、Uは、ディスクドライブ上、またはペンドライブ等の他の記憶媒体にオフラインでPSを記憶する。

【0191】

1つの実装において、Uは、PS内の各オブジェクトPをJ個の部分に分解し、このような各部分を異なる場所に配置する。場所は、ピア、ローカルの記憶装置、CA記憶装置等、上記の場所を含み得る。

20

【0192】

証明書C（Cプライム）を集合S内のいずれかの証明書として定義する。PKIシステムは、証明書CをCの有効性を有するものとして処理しなければならない。CがAの正しい証明書チェーンを維持することは、検証され得る。AはCに署名し、CはCに署名した。従って、CはPKI証明書チェーンを用いてAへの直接経路を有する。その後、各CがCによって署名されたことが明らかとなる。従って、Cの記録を有する場合、PKIシステムの他のメンバが明らかにCをAまで追跡することが可能であり、UがCを保持する際に、信頼のおける通信、認証、認可等がUに対するシステム内で発生することを可能にする。

30

【0193】

PKIシステムの各ユーザ（ディレクトリ、個人、CA、第3者等）は、より大きいシリアル値を含む証明書Cを有効証明書として、より小さいシリアル値を有するCによって署名された任意の現存の証明書を破棄され無効化されたものとして処理しなければならない。

【0194】

上記を含む場合の1つの実装において、終端値を含む最終的証明書Fを受け取る各ユーザは、Cを用いるいずれのトランザクションもはや許容してはならない。

【0195】

以下の例は、図19を参照してこの概念を実証する。

40

1. シリアル値2を有するCが、ディレクトリDに提示される。
2. Dは、シリアル値1を有するCを含む。
3. Dが、シリアル値1を有するCを廃棄し、シリアル値2を有するCを記憶する。
4. ユーザHが、Uに対する証明書をリクエストし、シリアル値2を有するCを受け取る。
5. Fが、ディレクトリDに提示される。
6. ユーザHが、Uに対する証明書をリクエストし、Fを受け取る。ユーザHは、トランザクションおよびいずれかの将来のトランザクションを許可しない。

【0196】

50

従って、全体としてのPKIシステム内における進行中の最新のC は有効C とみなされ、より小さいシリアル値を有する全ての他のものは、無視され、廃棄されたものとみなされる。システムのいずれかのユーザがC 内のシリアルに対してより大きい値を受け取る場合、そのC が使用され、より小さいシリアルC にかかれた接続部は閉じられ、より小さいシリアルC 証明書に対する全てのサービスが無効とされる。

【0197】

1つの実装において、リクエストまたは署名された指示または他のビジネスが、非公開キー部分、すなわちユーザを有するC を保持するエンティティによって実施され、または開始されると、そのユーザは、多数のディレクトリに対し、より大きいC が存在するか否かをチェックするよう問い合わせる。もしそうである場合には、そのリクエストまたはビジネスはキャンセルされるか、エンティティが接続解除されるか、等である。すなわち、より小さい値C の所有者は、有効なオーナーであるとはみなされない。このような実装においては、PKIを含むディレクトリの世界全体に1つ存在すると仮定すれば、クエリに加えられた各々の追加のディレクトリは、より大きい値C が見出される機会を増大させることが示され得る。

10

【0198】

Uは今や、同一の暗号強度を有して機能するCのための代替品として使用できる証明書のリストを有する。Uは、PKIシステムの残りがそのC を見るのみであり、より大きい値のC は見ない限りは、その集合内のC を使用することが可能である。

20

【0199】

C1、C2、、、CNをCERT内の証明書の値として定める。K1、K2、、、KNをCERT内のキーの公開/非公開対として定め、K1はC1を用いて暗号化されたデータを解読し、K2は、C2個の符号化データを解読する、等である。

20

【0200】

AがCに署名し、UがALGO1を実施した後、Uは、C1およびK1を用いて開始可能となる。C1またはK1が失われる、盗まれる、またはある期間に基づく場合、Uは以下のアクションを実施できる。明確にするために、1つの実装において、Uは、毎日、またはいずれかの期間に基づいて、証明書を循環させることができる。Uは、C2およびK2を含む暗号化されたまたは分割されたオブジェクトを得る。このオブジェクトは、1つの実装においては暗号化され、Uは、それを非公開パスワードで解読する。他の実装において、Uは全てのピースP を集め、様々な場所からC2およびK2を再構築し、その後、内容を（もし解読される場合には）解読する。他の実装においては、UはピアTに接触し、各メンバを解読し、UがC2およびK2を再構築可能になるまでその部分をUに提示する。

30

【0201】

Uは今や、有効なC2およびK2を有する。Uは、1つまたはそれ以上のディレクトリにC2を分配する。このような各ディレクトリは、C1をC2と置換する。このような各ディレクトリに接触する将来のユーザ全ては、C1の代わりにC2を得て、それが有効であり且つC1が無効であることを確認できる。1つの実装において、Uはまた、Uが相互作用するか、または相互作用してきたユーザのリストにC2を分配する。これらのユーザは、C2をキャッシュし、即座にアタッカがC1を使用出来ないようにすることができる。1つの実装において、ユーザがC2をキャッシュするならば、ユーザは、1つまたはそれ以上のディレクトリに接触して最新の証明書をリクエストする必要はなくなる。

40

【0202】

C1を有するアタッカは、C2が一旦システム内に導入されると、Uのためのデータおよびサービスへのアクセスが不可能になる。C1は、たとえ明白な失効プロセスが実施されなくても、実際には無効化される。代わりに、C2の発布が、C1の使用を無効化し且つ不要にする。この型式の積極的なコントロールは、Uに、それ自体の証明書有効性ステータスを管理し、知ることによって最も利益を得るシステムのユーザに対して新たな証明書の知識を公表する権限を与えるものであるため、非常に強力である。

50

【0203】

Uが、証明書、証明書失効リスト(CRL)または他のデータをAからリクエストすることが必要である時がなかったことに注目されたい。全ての失効は、トランザクションが発生する時にのみ、Uおよびシステムの様々な他の部分によって処理されている。さらに、UおよびC2を再構築するために頼りにしたいいずれかのピアグループを除いて、誰も手動の方法に参加しなかった。Xが1からnである将来の各CXに対して、Uは同じ演算を実施することが可能である。1つの実装において、Uが彼らの証明書はもはや使用するべきではないと判断する時、もしくはその場合、または、Cが有効なタイムスタンプをもはや含まないために、Uはディレクトリまたは他の手段を介してPKIシステムに証明書Fをリリースすることが可能である。他の実装において、ALGO1からのステップ2を使用する代わりに、Aは、各キーにpxで署名することに代えて各キーに署名する。Aは、証明書に署名しなければならないが、最初のC1を越えてその証明書を分配したり公布したりしてはならない。この場合、証明書チェーンは、A > C のようになる。そうでない場合は、ステップは同じある。

トラベリングキー：

【0204】

リレーショナル認可を使用してキー更新を実施するには、中央ルートとの通信、ピアの関与およびOの部分に対する時間と努力が必要である。さらに、ユーザが更新プロセスを実施しなければならない時は、その度に、中央ルートが必要となり、これによって、ENT中央システムに余分の負荷を生成され得る。ユーザが使い捨てのキーを有するのであれば、より良い。これによって、ユーザは、様々な目的のために一時的にその非公開キーをホスティングする装置を切り替えることが可能になり、その装置がなくなるか、または盗まれる等の場合を許容する。ユーザが、ルートサーバに接触することなく、より頻りにキーを切り替えることも可能になる。理想的には、ユーザが、ルートサーバに可能な限り接触しないようにするべきである。ENTにおいて、これらの取り換え可能なキーは、トラベリングキーと呼ばれる。トラベリングキーは、非公開非対称キーと、連番を含む公開証明書とから構成される。上記セクションによれば、トラベリングキー証明書におけるより大きい連番は、より小さい連番を有する既存のトラベリングキー証明書を無効化する。

【0205】

トラベリングキーは、キー無効化および除去のための上記機構を使用する。ベリニムの非公開キー部分は、トラベリングキーのグループに署名し、さらにそれを作成するために使用される。そのキーは、その後破壊され、同等レベルのセキュリティおよび必要に応じてキーをロールオーバーする能力を提供する1組のトラベリングキーを残す。

【0206】

1つの実装において、1組のトラベリングキーが製造される。数は実際には変化するが、30以上で十分となるはずである。さらに、終了証明書もまた、上記ルールに従って作成される。終了証明書がENT内のいずれかのピアノードにリリースされる場合、そのピアノードは、もはや現存のベリニム証明書を受け入れず、ベリニムは、ピア更新プロセスを用いて更新する必要がある。

【0207】

1つの実装において、ユーザは、安全な1か所において、そのトラベリングキーのいくつか、または全てを記憶することとしてもよい。しかしながら、好ましい実装において、トラベリングキーはいくつかのグループのピア間で分配される。ピアは、ピア更新プロセス用に使用されるものと同じピアであってもよく、または異なる組であってもよい。

【0208】

1つの実装では、キーは、ラウンドロビン方式でピアに対して、(必要になるまで)記憶するために分配される。例えば、キーが分配されるピアが3個存在する場合、ピア1は、連番1のキーを受け取り、ピア2は、キー2を受け取る、等である。これによってユーザは、任意のピアと接触し、より大きい連番のキーを獲得することが可能になる。ENTシステム内に見られる最も大きい連番が有効なキーとして考えられるので、任意のピアが

10

20

30

40

50

より進化したキーを生成できるべきである。好ましい実装において、終了証明書は、全てのピアと共に記憶される。これによって、任意の既知のピアからユーザにアクセス可能となる。

【0209】

1つの実装において、分配された各キーは、ベリニムオーナーのみに知られているキーで暗号化される。これによって、意のままに、または信任状を含む装置またはメモリ記憶が不正アクセスされるか、または盗まれる場合に、いずれのピアもベリニム信任状にアクセスすることが妨げられる。この機構は、各トラベリングキー証明書および非公開キー対を暗号化するために、多数の非対称キー暗号化プロセスのうちの一つを使用する。非対称キーは、ベリニムオーナーによって選択されたパスワードである。

10

リレーショナル認可およびトラベリングキーを実装することについての操作上の配慮

【0210】

好ましい実装においては、ENTにより、ユーザは、その最近のトラベリングキーをいずれかの、または全てのルートに提示することが可能になる。ルートは、ユーザに対してENTシステム上で観察された最も有効なトラベリングキーを記憶する。実際には、ユーザが新たなトラベリングキーの使用を開始する際、そのキーのコピーをルートサーバに提示し、それによって、最新のキーのためのルートに対する任意のノードによる任意のクエリが、そのキーを返すことになる。

【0211】

1つの実装において、ENTにより、ユーザはその最新のトラベリングキーを他のENTノードに送信することにより、それらのノードを直接更新することが可能になる。これは、キー公布と呼ばれる。これは、ユーザが、直接に接触可能な様々なENT可能化サービスを有する場合に有用である。これらの場合、ユーザ（または、その代わりにソフトウェア）は、ユーザの記録済みサービスの全てに接触し、最新のトラベリングキーを直接送信することができる。ENTノードは、特に、そのノードに関連性がある場合、他のENTノードベリニムのキャッシュを維持するように促される。ENTノードがトラベリングキー証明書を受け取り、その証明書が、あまり有効ではないトラベリングキーのための現存の証明書よりも新しい場合、そのノードは、現存のキー証明書をより新しいバージョンと交換するべきである。この概念は、所与のユーザによって使用される任意のサービスがユーザの最新のENT信任状を有することが確実となることから、非常に有用である。これによって、アタッカが、不正アクセスされたトラベリングキーまたは以前のベリニム信任状を使用するために持ち得る機会を減らすことになる。さらに、サービスセキュリティポリシーによっては、性能を改良する場合もある。

20

30

【0212】

1つの実装において、ENTは、インターネット周辺に配置された多数のデータストアを有する。これらのストアの各々は、システム上のベリニムの部分集合に対するベリニム信任状およびトラベリングキーのリストを含む。ユーザは、これらのセンターのいずれかと共にキー公布を使用してもよい。いずれかのユーザサービスと同様に、より有効なベリニム信任状またはトラベリングキーを受け取るセンターは、その現存するコピーをより有効なものと交換する。実際、データストアは、ベリニム識別子の範囲をサービスするものと思われる。データストアが一旦ベリニムidを1から1000までカバーすると、他のものは、ベリニムidを1001から2000までをカバーすることになる、等である。実際には、多数のデータストアが同じidの範囲をカバーするものと思われる。多数のデータストアが同じベリニムidの範囲をカバーする場合、それらのストアは、ベリニム信任状またはトラベリングキーのいずれかの成功裏の更新を、同じベリニムidをカバーする他のストアに伝達するべきである。

40

【0213】

サービスに対するキー公布は、第1のステップとしての上記ルート提示機構上では、第1のステップとして好ましい。データストアに対するキー公布は、好ましい第2のステップである。全てのステップは、実際に推薦されており、可能な限り迅速に達成されるはず

50

である。好ましい実装においては、不正アクセスの場合の損失の値がより大きいサービスが、より小さい値のノード以前に接触されるべきである。全てのサービスが更新されると、データストアが更新されるべきである。ルート更新は、最終的に成就される。

【0214】

他の実装において、異なる公布技術を使用することが可能である。例えば、ピアツーピアネットワークを使用して、より新しい信任状に対して多数のピアをサーチすることが可能である。このような多くの他のトポロジおよび技術が存在する。

【0215】

所与のノードにより提供されたサービスの臨界性に基づいて、セキュリティレベルを設定することが可能である。

10

【0216】

例えば、ENTを使用する銀行は、損失コストがより高くなるために、チャットサイトよりも、厳密な有効性チェックに対して高い要求を有する。厳密なテストを実施すると、時間（待機時間）、帯域幅およびいくつかの倍数による計算の観点から、トランザクションのコストが増大する。従って、ENTは、多様なセキュリティレベルを提供する。ペリニムの検証は、2つの主たる段階から構成される。第1の段階は、キャノピ検証と呼ばれ、それぞれ異なるルートによって署名された多数の現存の証明書の検証から構成される。もしN個のルートが存在するとすると、全キャノピ検証がセキュリティチェックとなり、その場合、N/2個よりも多いルート署名チェーンが確認される。しかしながら、これは、ある型式のトランザクションに対して有用であるよりもセキュリティが高い可能性がある。従って、1およびN/2 + 1個の署名チェーン間のどこかの場所が、所与のトランザクションに対して検証されなければならない。より高いセキュリティトランザクションに対しては、完全セキュリティチェックを実施するべきである（グループ権限セクションで定義したように、100%以上の信頼性レベル）。自明な、または非常に小さい値のトランザクションに対しては、1個のルートに対して1個の署名チェーンチェックが達成され得る。1個のルート署名チェーンのみのチェックにより、そのルートをコントロールするアタッカがユーザになりすますことが可能になる。これは、より多くのルートチェーンを検証することにより軽減される。なぜなら、アタッカが多数のルートに不正アクセスする可能性は小さいからである。完全キャノピ検証レベルでは、アタッカは、N/2個よりも多いノードのコントロールを獲得していなければならない。事実上、ENTシステム全体のコントロールを握らなければならないことになる。

20

30

【0217】

第2の段階は、ペリニム信任状に対するシステム全体のサーチと、最も有効なトラベリングキー信任状（もし使用するならば）とから構成される。アタッカがサービスにアクセスし、古い信任状を手渡し、サービスがより新しい信任状の存在に対してチェックしない場合、サービスは、アタッカの信任状が有効であったと仮定するだろう。高価値トランザクションに対して、最も安全な機構は、有効ペリニム信任状と有効トラベリングキーとの両方に対して適切なデータストアの1つをチェックすることである。しかしながら、低価値トランザクションに対しては、このステップは、省略されるか、または「怠惰に」実施され得る。怠惰なチェックによって、トランザクションは継続可能となる。しかしながら、チェックは、トランザクションが継続可能である間は、適切なデータストアに対して非対称的に成就される。サーチがより新しい信任状を見つけ出し、トランザクションを開始するために使用した現存の信任状が無効であることを証明する場合、トランザクションは、可能であるならば、終了して逆行させるべきである。

40

【0218】

1つの実装において、第2段階のチェックは、ユーザが決定した適時の期間内にサーチが既に行われている場合、スキップされ得る。例えば、以前のサーチが最後の30分以内に行われていた場合、チェックをスキップすることが可能である。

【0219】

好ましい1つの実装は、3つのセキュリティレベルを使用する。「過度に単純化した」

50

レベルチェックは、1個のルートに対するキャノピ検証を単に実施するのみであり、第2段階は全く行わない。「基本」レベルは、完全セキュリティチェックを実施し、その後、より新しい信任状に対して「怠惰な」サーチを行う、というものである。「完璧」なレベルチェックは完全セキュリティチェックを実施し、トランザクション前の信任状サーチは継続するように許可される。

【0220】

1つの実装において、トランザクションはキャッシュされることを許される。これによって、ペリニム信任状またはトラベリングキーが変化するまで、後のトランザクション上でキャノピ検証がスキップ可能となる。ペリニムを有する第1のトランザクションでは、サービスがキャノピ検証を必要とする。しかしながら、チェックは、ルートの過半数が所与のペリニムに対してパウチャされるのを確実にするように全体的に構成される。ペリニム信任状がその最初のトランザクション以来変化していない場合、後のトランザクションは、他のキャノピ検証を実施する必要なく、このキャッシュされた結果を使用できる。

10

【0221】

両方のセキュリティチェックが一旦行われると、サービスはオーナシップの証拠をリクエストする。これは、サービスを用いてトランザクションを開始するユーザがトラベリングキーの適当な非公開キー部分を有し、またはトラベリングキーを使用しない場合には、ペリニム内に見られる公開部分にマッチングする非公開キー部分を有することを確実にする。これは通常、T L S 標準に見られるようなハンドシェイク機構を伴う。このトピックは他の情報原によってうまくカバーされ、信頼性を決定し、非公開通信チャネルを確立するためのP K Iシステム内に見られる従来機構に従う。E N Tにおいては、もし利用可能であれば、トラベリングキーを使用する場合もある。そうでない場合は、すべて同一の公開キーを有するのであるから、ペリニム信用状内のいかなる証明書が使用されてもよい。

20

【0222】

1つの実装において、トランザクションが開始されると、ユーザは最新のペリニムおよびトラベリングキーの情報をサービスに送信する。これによって、サービスは、「過度に単純化した」または「基本的な」セキュリティチェックを実施中であるならば、他のサービスに接触する必要なく、トランザクションを処理することが可能になる場合もある。

【0223】

図20を参照すると、ブロック図は一実施形態によるシステム100を示しており、このシステム100は、上記のようにE N Tシステムを使用して、様々な他のシステムにアクセスすることが可能なユーザアクセスターミナル105を備える。ユーザアクセスターミナル105は、スマートフォン、携帯電話、V o I Pフォン、パーソナルデジタルアシスタント、タブレットコンピュータ、ラップトップコンピュータ、携帯用デジタル音楽プレーヤ、音声もしくはデータを伝達する他のモバイルデバイス、またはこれらの組み合わせのような多数のデバイスのうちの1つであってもよい。ユーザアクセスターミナル105は、例えばローカルエリアネットワークへの有線または無線接続を含むネットワーク接続コンピュータシステムを備えることも可能である。ユーザアクセスターミナルは、電子アプリケーションに対するユーザアクセスをコントロールするための機能を果たすために動作可能な好適なデバイスを備えてよく、図15に示す特定のコンポーネントは、ここで述べる一般的概念を例示し、述べるためのものであることが容易に理解されるであろう。様々な実施形態においては、ユーザアクセスターミナル105は、上記例による動作が可能である。

30

40

【0224】

図20の実施形態におけるユーザアクセスターミナル105は、直接に、またはネットワークを介してアクセスシステム110に接続する。このようなネットワークは、多数の異なるプロトコルのいずれかでデータを送信することが可能な好適なネットワークを備えることも可能である。このようなネットワークは、周知のものであり、ここでさらに詳細に説明する必要はない。アクセスシステム110は、例えば、他のネットワーク接続コン

50

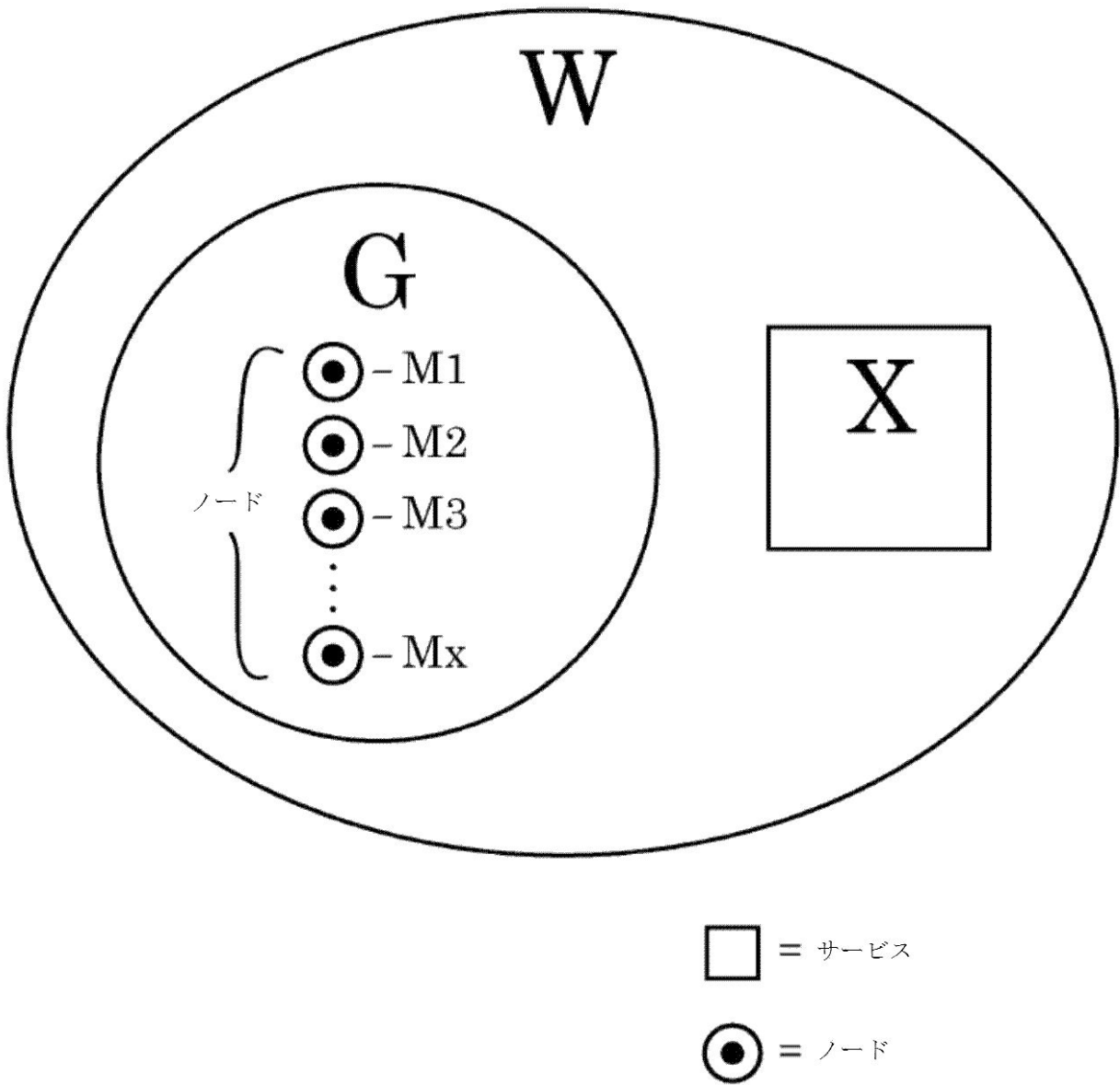
ポーンを有するインターネット等のネットワーク 115 に相互接続される。セントラルサーバコンピュータシステム 120 はネットワーク 115 に接続され、様々な実施形態で、上記のように ENT システムに関連する機能を果たす。セントラルサーバコンピュータシステム 120 は、例えば、1 個以上のサーバコンピュータ、パーソナルコンピュータ、ワークステーション、ウェブサーバ、または他の好適なコンピューティングデバイスから構成することができ、所与のサーバのための個々のコンピューティングデバイスは、ローカルにあるか、または互いに遠く離れていてもよい。ユーザシステム 125 もまた、ネットワーク 115 に直接接続され得る。このようなユーザシステム 125 は、上記のようなシステムを採用できる他のユーザアクセス点であってもよい。

【 0 2 2 5 】

本発明は、例示の目的のためにのみ、特定の実施形態を用いてここで述べられた。しかしながら、本発明の原理が他の方法でも具体化可能であることは、当業者には直ちに明らかであろう。従って、本発明は、ここで開示した特定の実施形態に限定されるものではなく、以下に示す請求の範囲のスコープに十分見合うものとして見なされるべきである。

【図1】

図1



【図2】

図2

	署名者						
	M1	M2	M3	M4	M5	...	MN
署名済み	M1	M1S1	M2S1	M3S1	M4S1	M5S1	MNS1
	M2	M1S2	M2S2	M3S2	M4S2	M5S2	MNS2
	M3	M1S3	M2S3	M3S3	M4S3	M5S3	MNS3
	M4	M1S4	M2S4	M3S4	M4S4	M5S4	MNS4
	M5	M1S5	M2S5	M3S5	M4S5	M5S5	MNS5
	⋮						
MN	M1SN	M2SN	M3SN	M4SN	M5SN		MNSN

【図3】

図3

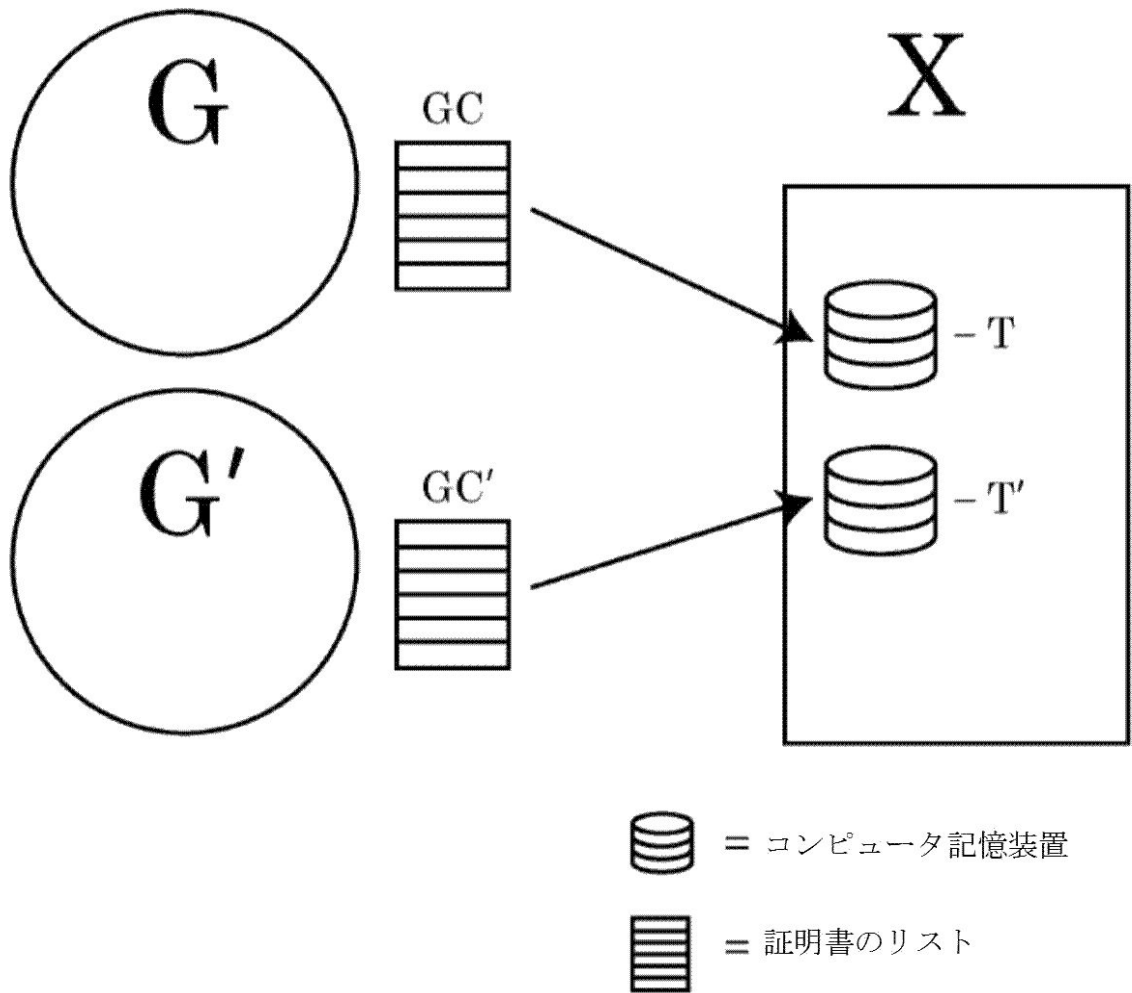
		署名者						
		M1	M2	M3	M4	M5	M6	M7
署名済み	M1	M1S1				M5S1	M6S1	M7S1
	M2	M1S2	M2S2				M6S2	M7S2
	M3	M1S3	M2S3	M3S3				M7S3
	M4	M1S4	M2S4	M3S4	M4S4			
	M5		M2S5	M3S5	M4S5	M5S5		
	M6			M3S6	M4S6	M5S6	M6S6	
	M7				M4S7	M5S7	M6S7	M7S7

N=7であると
仮定する

各 M_x は他の証明書に署名する ($N/2$ 切り下げ)。

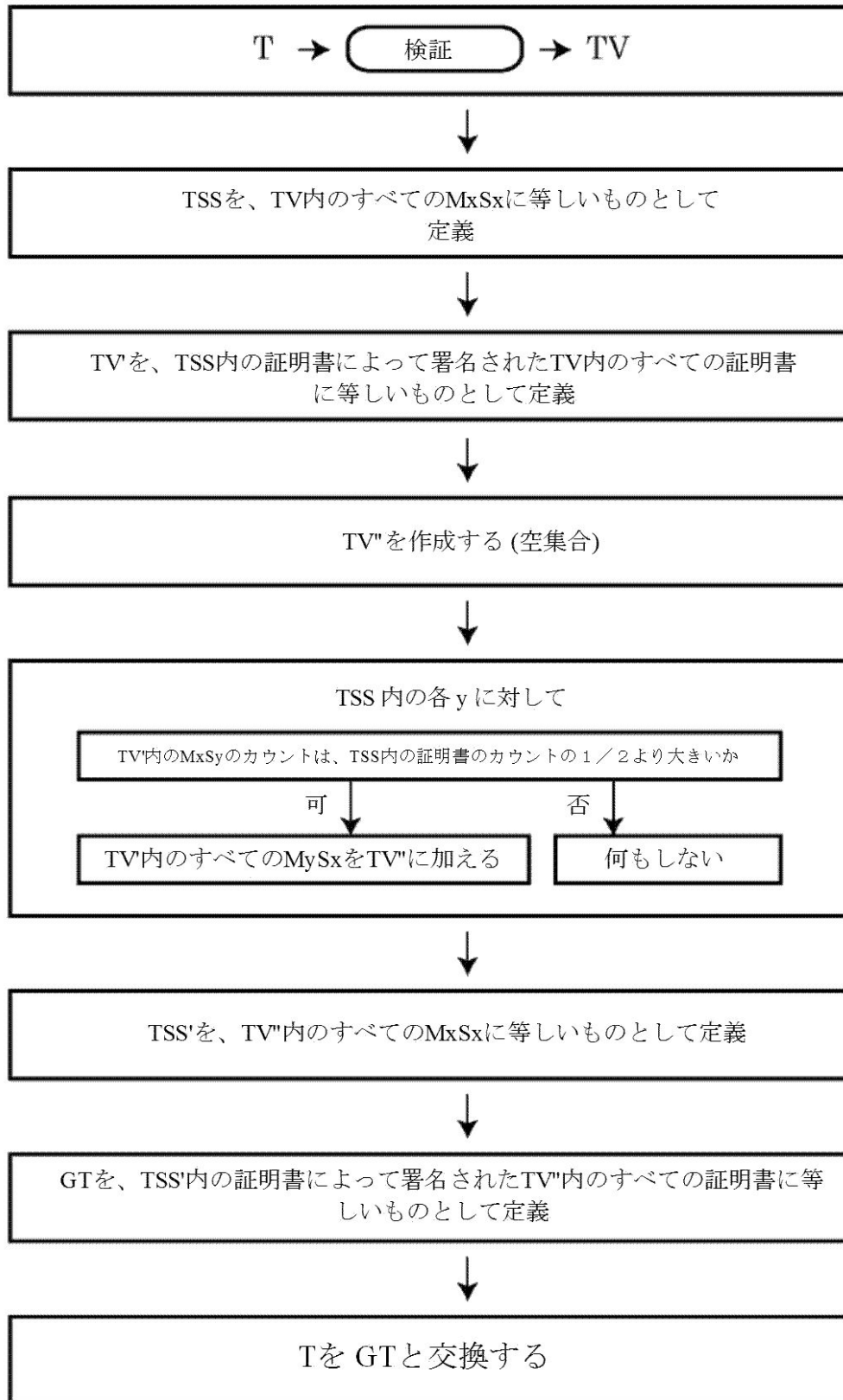
【 図 4 】

図 4



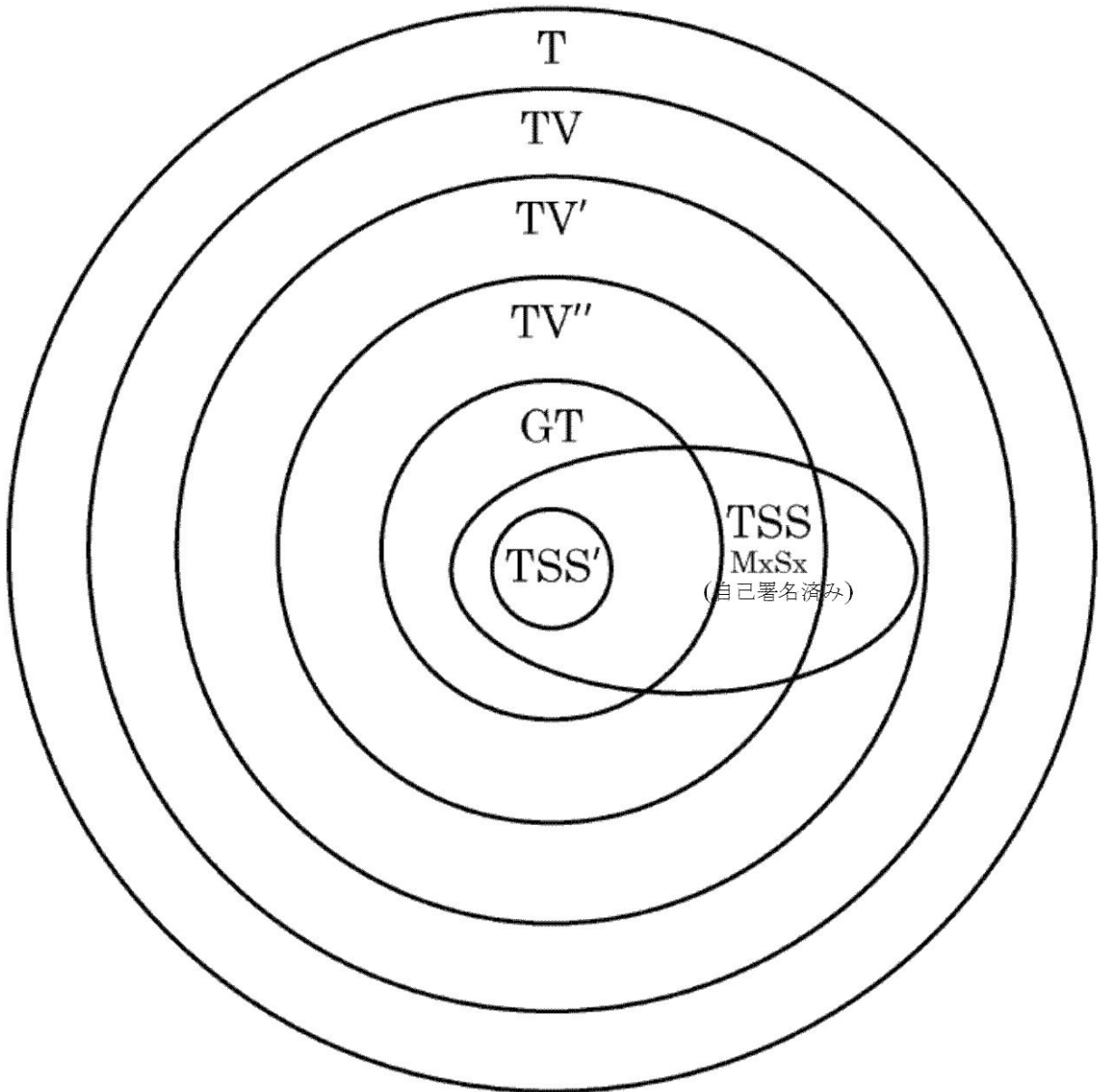
【図5.1】

図5.1



【図5.2】

図5.2



【 図 6 】

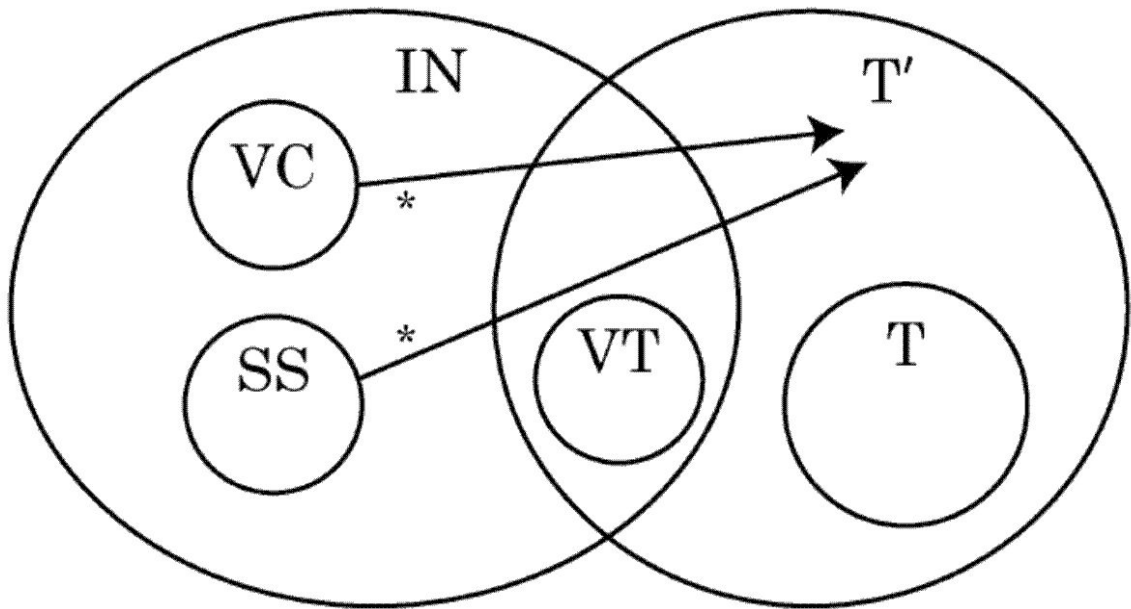
図 6

		署名者							
		M1	M2	M3	M4	M5	...	MN	<i>Mp</i>
署名済み	M1	M1S1	M2S1	M3S1	M4S1	M5S1		MNS1	<i>MpS1</i>
	M2	M1S2	M2S2	M3S2	M4S2	M5S2		MNS2	<i>MpS2</i>
	M3	M1S3	M2S3	M3S3	M4S3	M5S3		MNS3	<i>MpS3</i>
	M4	M1S4	M2S4	M3S4	M4S4	M5S4		MNS4	<i>MpS4</i>
	M5	M1S5	M2S5	M3S5	M4S5	M5S5		MNS5	<i>MpS5</i>
	⋮								
	MN	M1SN	M2SN	M3SN	M4SN	M5SN		MNSN	<i>MpSN</i>
	<i>Mp</i>	<i>M1Sp</i>	<i>M2Sp</i>	<i>M3Sp</i>	<i>M4Sp</i>	<i>M5Sp</i>		<i>MNSp</i>	<i>MpSp</i>

p = N+1である場合

【 図 7 】

図 7



T = X内のGに対する信頼性のあるストア

VC = ステップ 2 d において成功した y に対する IN 内のすべての MySx

T' = T 内の MxSx によって署名された IN 又は T 内のすべての証明書

SS = IN 内の全ての MxSx

IN = 加えられる証明書

VT = B 内の MxSp をターゲットとする T 内のすべての証明書

* ステップ 2 f が成功する場合

【 図 8 】

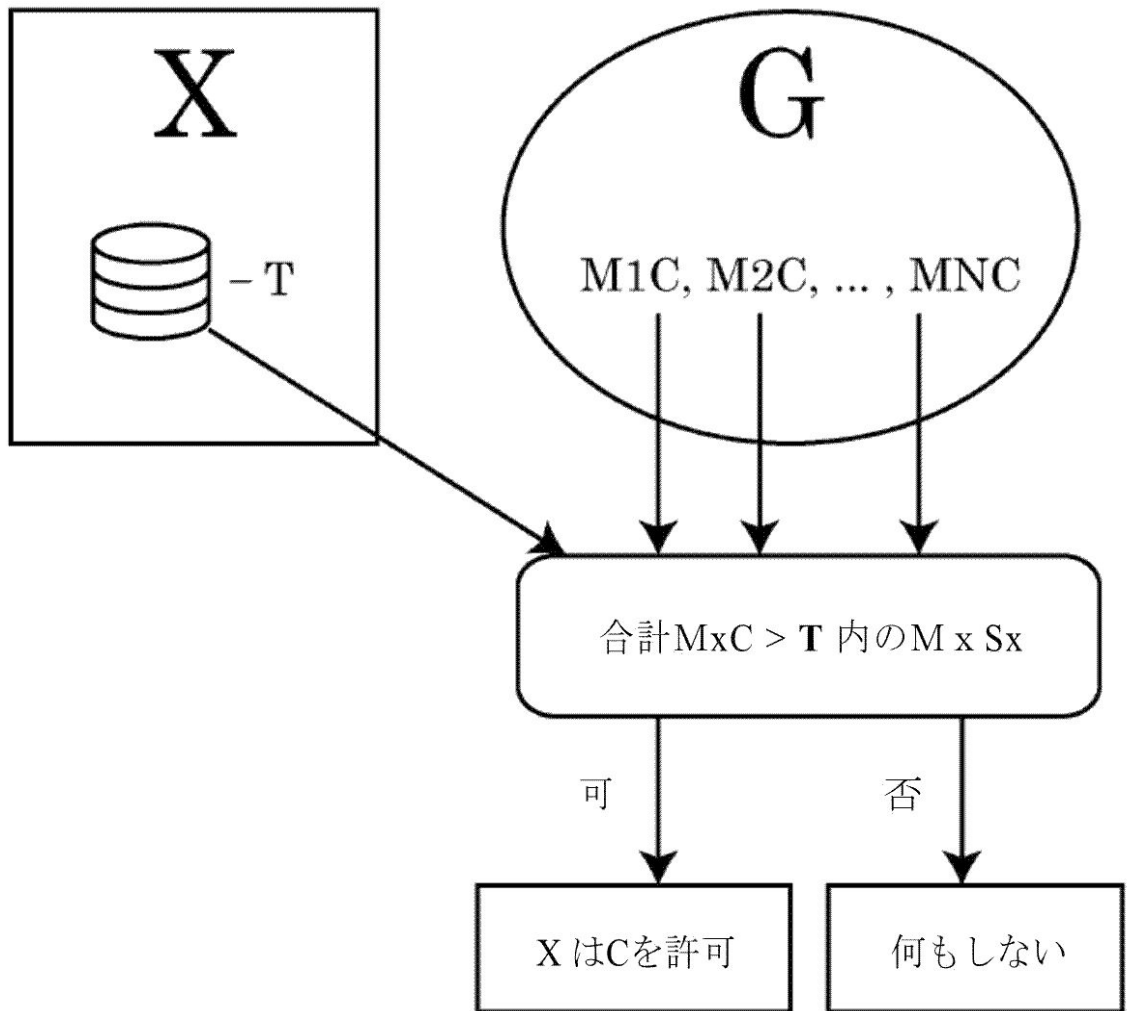
図 8

		署名者						
		M1	M2	M3	M4	M5	...	MN
署名済み	M1	M1S1	M2S1	M3S1	M4S1	M5S1		MNS1
	M2	M1R2	M2S2	M3S2	M4S2	M5S2		MNS2
	M3	M1S3	M2S3	M3S3	M4S3	M5S3		MNS3
	M4	M1S4	M2S4	M3S4	M4S4	M5S4		MNS4
	M5	M1S5	M2S5	M3S5	M4S5	M5S5		MNS5
	⋮							
	MN	M1SN	M2SN	M3SN	M4SN	M5SN		MNSN

MxRy = 破棄

【 図 9 】

図 9



【 図 1 0 】

図 1 0

```

{"keys":[
  {"id":"M1","key":"A0760BCDEF839298","algorithm":"RSA"},
  {"id":"M2","key":"DEF8392076098024","algorithm":"RSA"},
  {"id":"M3","key":"CDEF839207609FB8","algorithm":"RSA"},
  {"id":"M4","key":"9207609802A7CDEF","algorithm":"DSA"},
  {"id":"M5","key":"BC1048A392076098","algorithm":"ECC"},
  {"id":"M6","key":"A0980BCD392076EF","algorithm":"ECC"},
  {"id":"M7","key":"83F0760992CDABE8","algorithm":"DSA"},
],
"hash_type":"SHA256",
"notvalidbefore":0,
"expiry":4000000000,
"clock":1,
"signatures":{
  "M1":"SLDKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLDJ",
  "M2":"LKFJLEHRISLDKFJSELKRJSELKJFDSKHLKJDKLDJ",
  "M3":"SLDSELKJFDLKFJLEHRISKHSLDKFJSELKRJSELKJ",
  "M4":"HRISKHLKFJSELKRJKJDSLDKFJSELKRJSELKJFDL",
  "M5":"KFJSELKRJSEFJLEHRISKHLKFJSELKRJSEKJDKLDJ",
  "M6":"LEHRISKHLKJDKLDJFLSLDKFJSELKRJSELKJFDLK",
  "M7":"DKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLSLDJ",
}
}

```

J S O N シンタックスを用いる本図面においては、G はリスト "keys" で表される。
G 内の各 M x に対する署名は "signatures" 内に見られる。

【 図 1 1 】

図 1 1

```

{"keys":[
  {"id":"M1","key":"A0760BCDEF839298","algorithm":"RSA"},
  {"id":"M2","key":"DEF8392076098024","algorithm":"RSA"},
  {"id":"M3","key":"CDEF839207609FB8","algorithm":"RSA"},
  {"id":"M4","key":"9207609802A7CDEF","algorithm":"DSA"},
  {"id":"M6","key":"A0980BCD392076EF","algorithm":"ECC"},
  {"id":"M8","key":"760DABE992C83F08","algorithm":"DSA"},
],
"hash_type":"SHA256",
"notvalidbefore":0,
"expiry":4000000000,
"clock":2,
"signatures":{
  "M1":"SLDKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLDJ",
  "M2":"LKFJLEHRISLDKFJSELKRJSELKJFDSKHLKJDKLDJ",
  "M3":"SLDSELKJFDLKFJLEHRISKHSLDKFJSELKRJSELKJ",
  "M4":"HRISKHLKFJSELKRJKJDSLDKFJSELKRJSELKJFDL",
  "M6":"LEHRISKHLKJDKLDJFLSLDKFJSELKRJSELKJFDLK",
  "M8":"FJLFELKJFDLKKLSJEHRISKHLKJDDKSELKRJSLDJ",
  }
}

```

J S O N シンタックスを用いる本図面においては、M5はGから除去され、M8は追加されている。本文書が図11に示すものを交換すると仮定する。

【 図 1 2 】

図 1 2

```

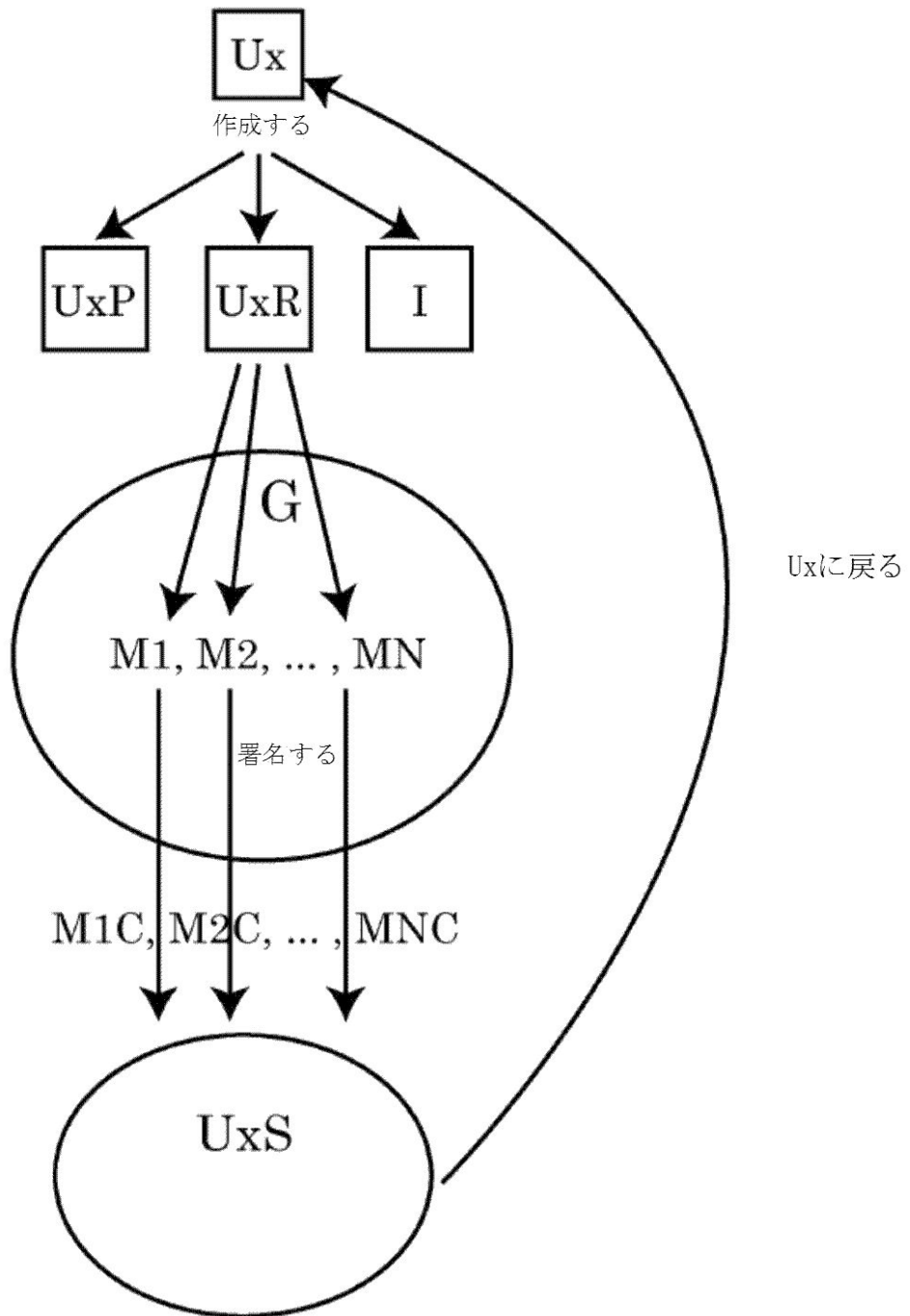
{"keys":[
  {"keys":[
    {"id":"M1","key":"A0760BCDEF839298","algorithm":"RSA"},
    {"id":"M2","key":"DEF8392076098024","algorithm":"RSA"},
    {"id":"M3","key":"CDEF839207609FB8","algorithm":"RSA"},
    ],
    "quorum":2,
  },
  {"keys":[
    {"id":"M5","key":"BC1048A392076098","algorithm":"ECC"},
    {"id":"M6","key":"A0980BCD392076EF","algorithm":"ECC"},
    {"id":"M7","key":"83F0760992CDABE8","algorithm":"DSA"},
    ],
    "quorum":2,
  },
  {"keys":[
    {"id":"M4","key":"9207609802A7CDEF","algorithm":"DSA"},
    ],
    "quorum":1,
  },
],
"hash_type":"SHA256",
"notvalidbefore":0,
"quorum":2,
"expiry":4000000000,
"clock":1,
"signatures":{
  "M1":"SLDKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLDJ",
  "M2":"LKFJLEHRISLDKFJSELKRJSELKJFDSKHLKJDKLDJ",
  "M3":"SLDSELKJFDLKFJLEHRISKHSLDKFJSELKRJSELKJ",
  "M4":"HRISKHLKFJSELKRJKJDSLDFJSELKRJSELKJFDL",
  "M5":"KFJSELKRJSEFJLEHRISKHLKFJSELKRJSEKJDKLDJ",
  "M6":"LEHRISKHLKJDKLDJFLSLDKFJSELKRJSELKJFDLK",
  "M7":"DKFJSELKRJSELKJFDLKFJLEHRISKHLKJDKLSLDJ",
}
}

```

J S O N シンタックスを用いる本図面においては、"keys"フィールドは、交換を計算するための代数学を各々が有するオブジェクトを含む。

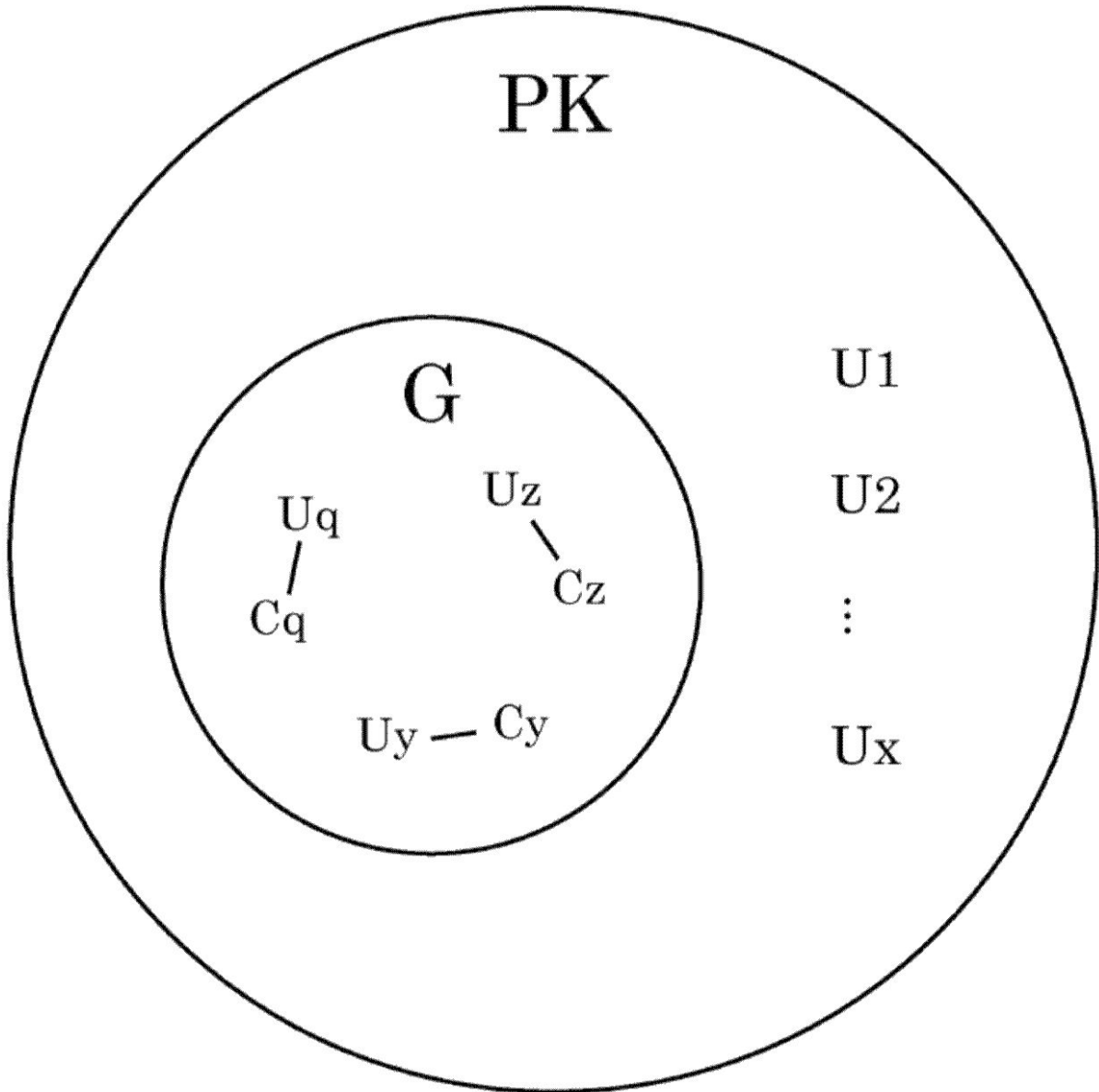
【図13】

図13



【 図 1 4 】

図 1 4



【 図 1 5 】

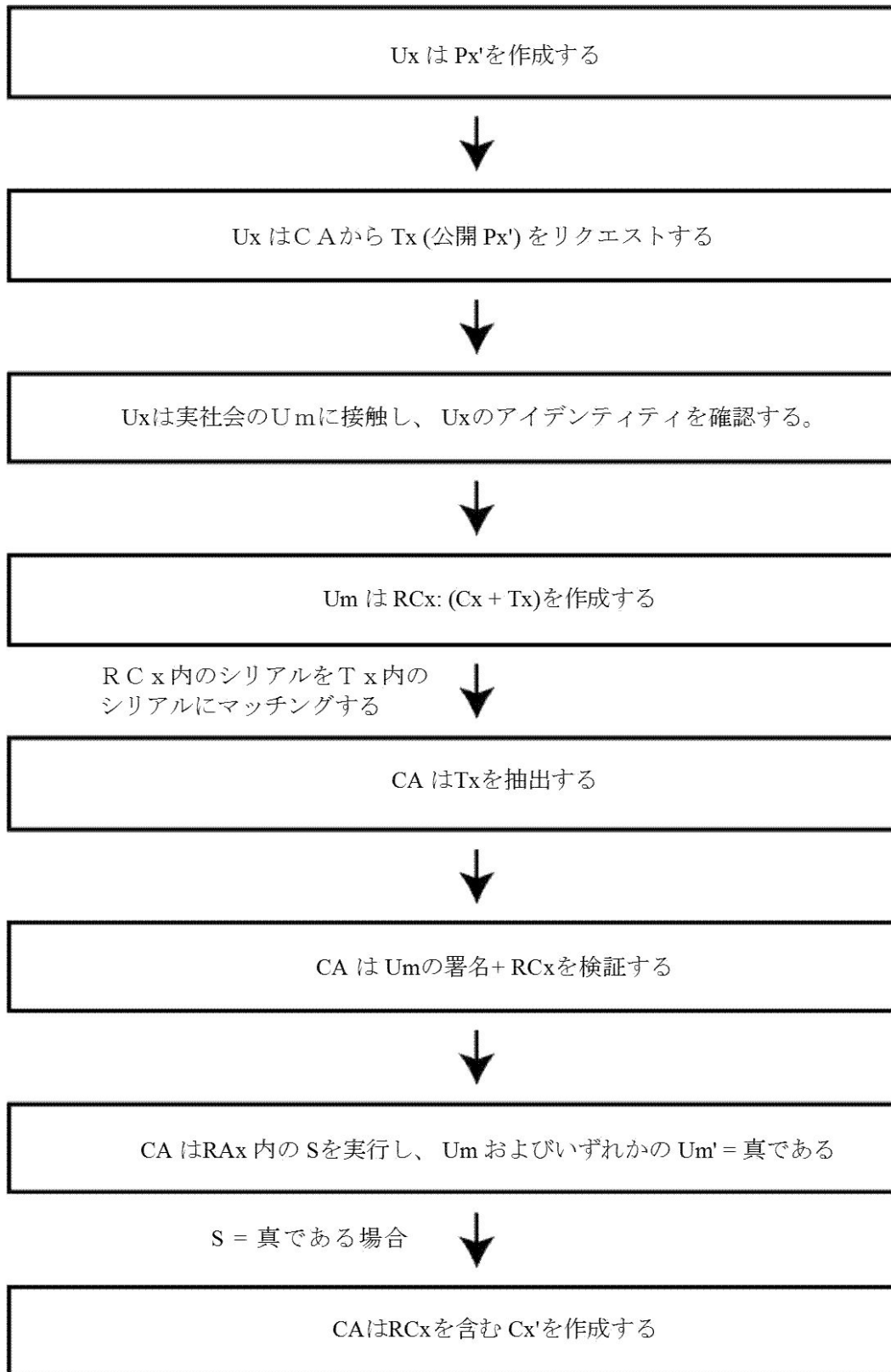
図 1 5

```
{
  "credential": {
    "id": 4,
    "policies": {
      "renewal": "(1 and 2) or (3 and 2) or (1 and 3)",
      "dataAccess": "(1 and 2) or (3 and 2) or (1 and 3)",
      "authentication": "(1 and 2) or (3 and 2) or (1 and 3)"
    }
  },
  "nonce": "SLKJa2r36FkRHS"
},
"hash_type": "SHA256",
"signatureCA": "DLKSJLSEKRHSELKJKDFLFLKJS"
}
```

本図面においては、id 4 を有するU x が3つのポリシーを設定し、1つは更新用、別の1つはデータアクセス用、さらに残りの1つはCAを用いる認証用である。CAは、本文書、特に"credential"サブオブジェクトストリングに署名している。

【図16】

図16



【 図 1 7 】

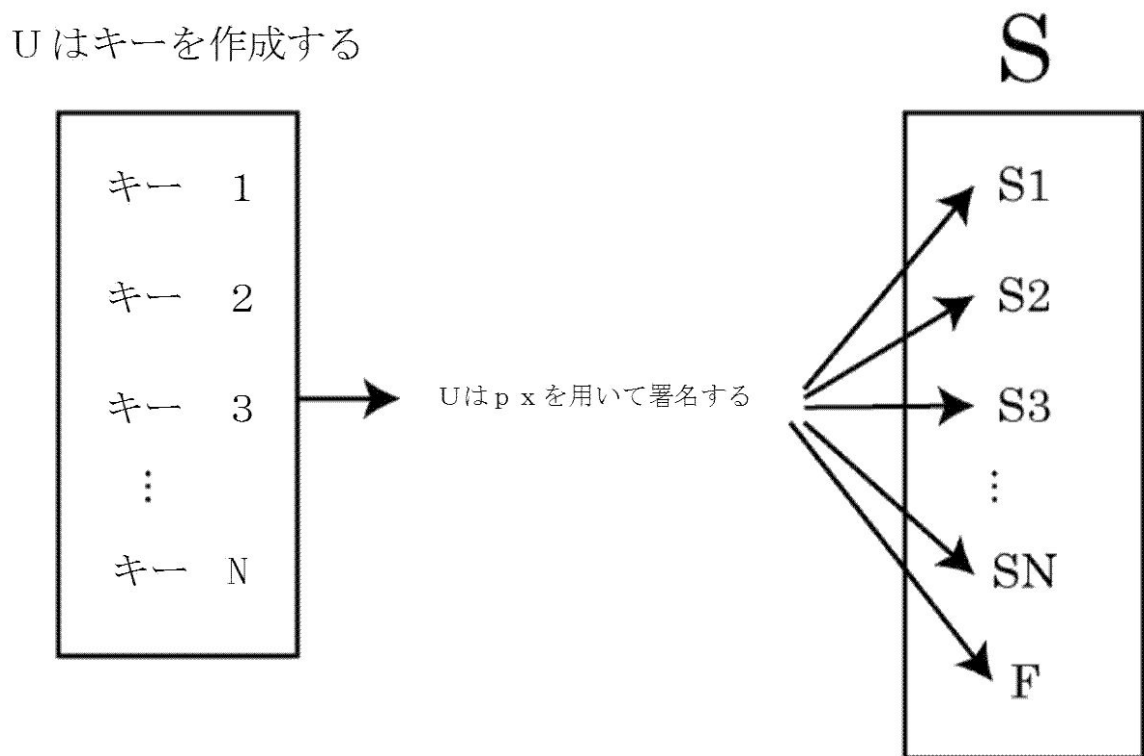
図 1 7

```
"renewal":{
  "id":4,
  "serial":850293,
  "nonce":"SLKJSELRKHS"
},
"hash_type":"SHA256",
"vouchers":[
  {"id":1,"signature":"DLKSJLSEKRHSELKJKDFLFLKJS"},
  {"id":2,"signature":"AWEOTQPNVXIMNERIOWJFLDIK"}
]
}
```

本図面においては、U_xはid 4を有し、U_m1および2は、異なる署名を有する2つの同一の文書を送る代わりに、本文書内に存在する"renewal"サブオブジェクトストリングの署名を有する。

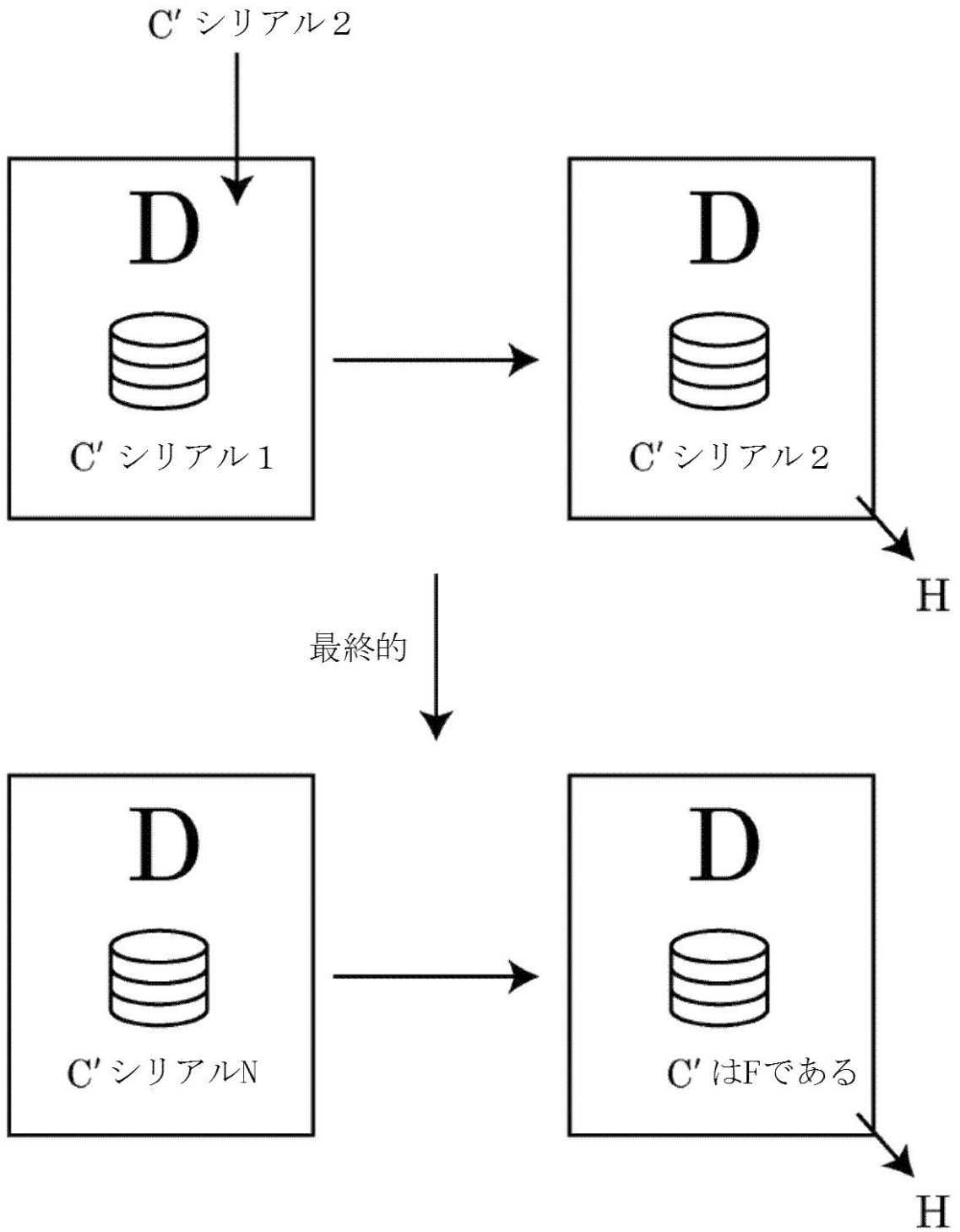
【 図 1 8 】

図 1 8



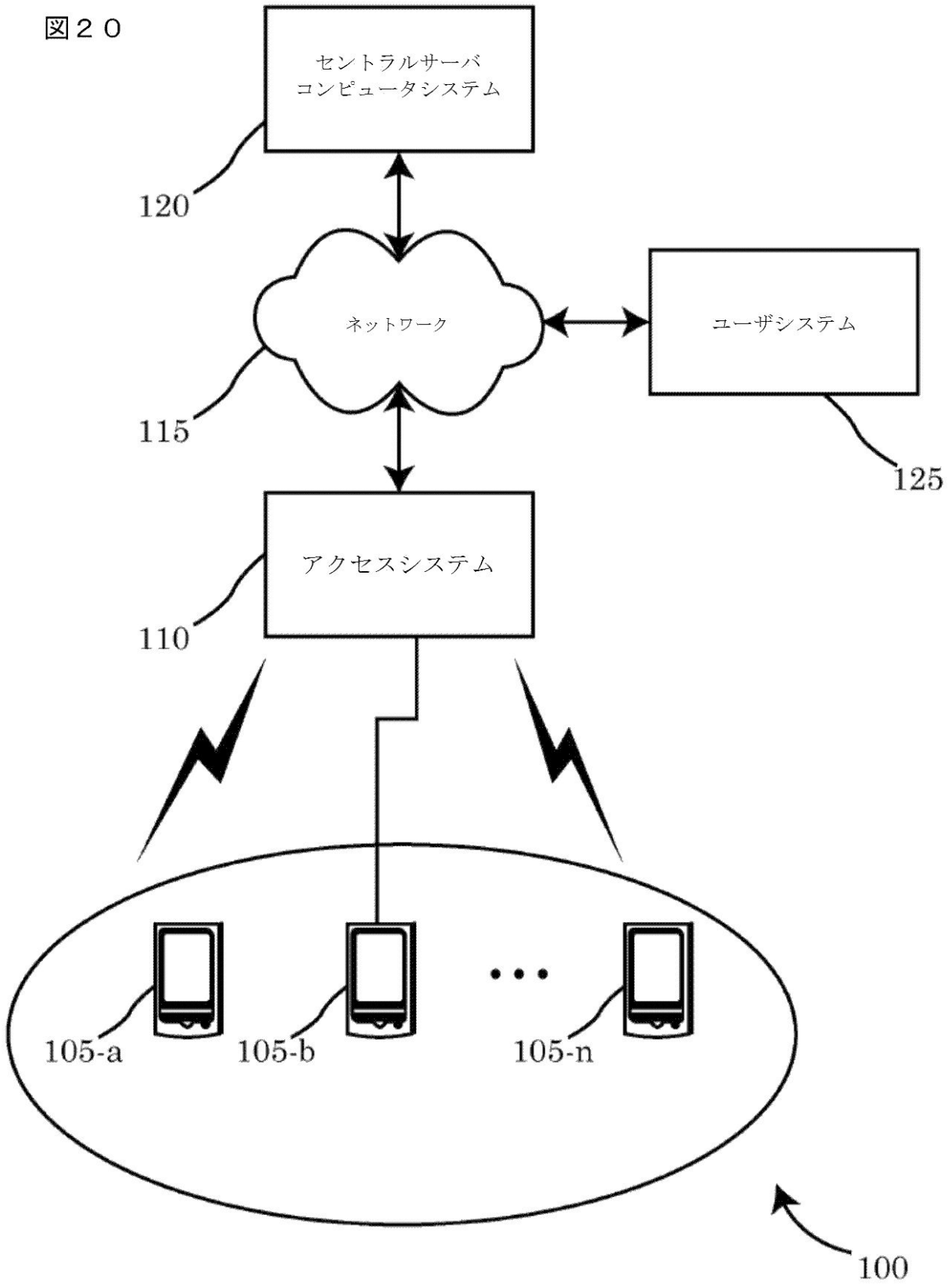
【図19】

図19





【図20】

図20



【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US2013/069217
A. CLASSIFICATION OF SUBJECT MATTER		
H04L 9/32(2006.01)i, H04L 9/30(2006.01)j		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) H04L 9/32; H04L 9/30; H04L 9/06; H04L 9/00; G06F 21/00		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & Keywords: certificate, PKI, group, server		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2008-0028209 A1 (PETER ROY DARE et al.) 31 January 2008 See paragraphs [0057]-[0067]; figures 2A, 2B; and claims 1, 2.	1-15
A	US 2009-0031131 A1 (XIN QIU et al.) 29 January 2009 See paragraphs [0034]-[0046]; and figures 1, 2.	1-15
A	US 2012-0257752 A1 (HERB A. LITTLE) 11 October 2012 See paragraphs [0040]-[0045]; and figure 4.	1-15
A	US 2011-0126022 A1 (WALTER SIEBERER) 26 May 2011 See paragraphs [0022]-[0084]; and figures 1-4.	1-15
A	US 2009-0259841 A1 (KENNETH P. LABERTEAUX et al.) 15 October 2009 See paragraphs [0032]-[0054]; and figures 1-3.	1-15
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 28 April 2014 (28.04.2014)		Date of mailing of the international search report 29 April 2014 (29.04.2014)
Name and mailing address of the ISA/KR  International Application Division Korean Intellectual Property Office 189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. +82-42-472-7140		Authorized officer KIM, Do Weon Telephone No. +82-42-481-5560 

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/069217

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008-0028209 A1	31/01/2008	GB 0204664 D0	10/04/2002
		GB 2385955 A	03/09/2003
		US 2003-0163687 A1	28/08/2003
		US 7308574 B2	11/12/2007
		US 7937584 B2	03/05/2011
US 2009-0031131 A1	29/01/2009	CN 101816140 A	25/08/2010
		MX 2010001059 A	03/03/2010
		US 8392702 B2	05/03/2013
		WO 2009-018032 A1	05/02/2009
US 2012-0257752 A1	11/10/2012	AT 333732 T	15/08/2006
		CA 2312331 A1	23/12/2000
		CA 2312331 C	09/05/2006
		DE 60029391 D1	31/08/2006
		DE 60029391 T2	19/07/2007
		EP 1063813 A2	27/12/2000
		EP 1063813 A3	19/06/2002
		EP 1063813 B1	19/07/2006
		US 2010-174910 A1	08/07/2010
		US 2013-294600 A1	07/11/2013
		US 7707420 B1	27/04/2010
		US 8219819 B2	10/07/2012
		US 8499160 B2	30/07/2013
		US 2011-0126022 A1	26/05/2011
DE 502006008733 D1	24/02/2011		
EP 1946481 A1	23/07/2008		
EP 1946481 B1	12/01/2011		
WO 2007-053864 A1	18/05/2007		
WO 2007-053864 A9	19/07/2007		
US 2009-0259841 A1	15/10/2009	US 8230215 B2	24/07/2012

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US

(72)発明者 モスバーガー、 ティモシー
アメリカ合衆国 カリフォルニア州 91942 ラメサ、 レイク マレー ブルヴァード
5519、107号
Fターム(参考) 5J104 AA08 AA16 EA16 LA03 MA02 NA02 NA12 NA38 PA07