(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3595109号
(P3595109)
(45)発行日　平成16年12月2日(2004.12.2)　(24)登録日　平成16年9月10日(2004.9.10)

(51)Int.Cl.[7]　F I
H04L 9/32　H04L 9/00 675B
H04L 9/00 675D

請求項の数 6　(全 24 頁)

(21)出願番号　特願平9-138724
(22)出願日　平成9年5月28日(1997.5.28)
(65)公開番号　特開平10-336169
(43)公開日　平成10年12月18日(1998.12.18)
審査請求日　平成9年9月25日(1997.9.25)
審判番号　不服2000-19604(P2000-19604/J1)
審判請求日　平成12年12月11日(2000.12.11)

(73)特許権者　591030237
日本ユニシス株式会社
東京都江東区豊洲一丁目１番１号
(74)代理人　100076428
弁理士　大塚　康徳
(72)発明者　八津川　直伸
東京都港区赤坂２－１７－５１　日本ユニシス株式会社内

合議体
審判長　川名　幹夫
審判官　小田　浩
審判官　橋本　正弘

最終頁に続く

(54)【発明の名称】認証装置、端末装置、および、それら装置における認証方法、並びに、記憶媒体

(57)【特許請求の範囲】

【請求項１】

認証に先立ち、端末装置から送られてくる認証情報を検査するための第一の検査情報をサーバ装置のメモリに保存し、前記サーバ装置に接続された端末装置の認証要求に対して、ワンタイムパスワード方式および公開鍵暗号方式による認証を継続して行う認証方法であって、

端末装置から認証要求を受信すると、前記端末装置へ認証情報要求を送信し、

前記認証情報要求に応じて、前記端末装置に関連する秘密鍵を用いた暗号化により種情報から生成された認証情報を前記端末装置から受信し、

前記認証情報を前記端末装置に関連する公開鍵によって復号することにより第二の検査情報を生成し、

前記第二の検査情報と前記メモリに保存された前記端末装置に対応する第一の検査情報とを比較して、前記第一および第二の検査情報が一致する場合に前記認証要求を許可し、

前記認証要求を許可した場合、前記認証情報を、次回の認証用として前記端末装置に対応する第一の検査情報に代えて前記サーバ装置のメモリに保存し、

前記認証要求が許可された場合、前記認証情報は、次回の認証要求のための種情報として前記端末装置のメモリに保存され、

前記認証要求が許可されなかった場合、前記認証情報は、前記端末装置のメモリおよび前記サーバ装置のメモリに保存されることなく、破棄されることを特徴とする認証方法。

【請求項２】

前記認証情報には公開鍵証明書を示す情報が添付されていることを特徴とする請求項1に記載された認証方法。

【請求項３】

認証に先立ち、端末装置から送られてくる認証情報を検査するための第一の検査情報を保存するメモリを有し、端末装置の認証要求に対して、ワンタイムパスワード方式および公開鍵暗号方式による認証を継続して行う認証装置であって、

端末装置との間で認証に関連する情報の送受信を行う通信手段と、

前記メモリおよび前記通信手段を利用して、認証処理を行う認証手段とを備え、

前記認証手段は、端末装置から認証要求を受信すると、前記端末装置へ認証情報要求を送信し、前記認証情報要求に応じて、前記端末装置に関連する秘密鍵を用いた暗号化により種情報から生成された認証情報を前記端末装置から受信し、前記認証情報を前記端末装置に関連する公開鍵によって復号することにより第二の検査情報を生成し、前記第二の検査情報と前記メモリに保存された前記端末装置に対応する第一の検査情報とを比較して、前記第一および第二の検査情報が一致する場合に前記認証要求を許可し、

前記認証要求が許可された場合、前記認証情報は、次回の認証用として前記端末装置に対応する第一の検査情報に代えて前記メモリに保存されるとともに、次回の認証要求のための種情報として前記端末装置のメモリに保存され、

前記認証要求が許可されなかった場合、前記認証情報は、前記端末装置のメモリおよび前記サーバ装置のメモリに保存されることなく、破棄されることを特徴とする認証装置。

【請求項４】

前記認証情報には公開鍵証明書を示す情報が添付されていることを特徴とする請求項3に記載された認証装置。

【請求項５】

前記端末装置のメモリには、前記種情報、前記秘密鍵および前記認証情報を生成するためのプログラムコードが格納されていることを特徴とする請求項3または請求項4に記載された認証装置。

【請求項６】

サーバ装置を制御して、請求項1または請求項2に記載された認証を実行するプログラムコードが格納されたことを特徴とする記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は認証装置、端末装置、および、それら装置における認証方法、並びに、記憶媒体に関し、例えば、ネットワークを介して認証要求を行う端末装置、認証要求を処理する認証装置、および、それら装置の認証方法、並びに、それらを実現するプログラムコードが格納された記憶媒体に関する。

【０００２】

【従来の技術】

情報処理システムが社会活動のあらゆる局面において中心的な役割を演じるようになった現在、ネットワークを介した個人間や個人一企業間或いは企業間の情報通信に対するセキュリティ保護が緊急の課題となっている。

特に昨今のネットワーク・システムのオープン化・汎用化により、機密情報転送や電子商取引（Ｅｌｅｃｔｒｏｎｉｃ　Ｃｏｍｍｅｒｃｅ）のような分野に対し、セキュリティ機能は必要不可欠なものとなっている。例えば、企業間、個人間或いはそれら相互間で法律行為をなす場合、従来（現在でも）、物理的な紙を使用して契約書等を作成し、署名し、印鑑を押印し、更に必要に応じ、印鑑登録証や公証人による公正証書を添付し、次にこれら文書を相手方に送付する際、書留にし、或いは内容証明郵便にする。

【０００３】

このような物理的な書類を中心にした行為を全て、電子的な情報通信によって安全に代替ならしめるものが、ネットワーク・セキュリティ技術である。コンピュータとネットワー

クによる情報通信網が全世界規模に達した現在、このような要求は増大の一途である。
ネットワーク・セキュリティの目的は、ネットワークの安全保護に有り、ネットワーク・システムの機密度に応じた情報をさまざまな脅威から保護することであるとされている。一般的には、▲１▼機密性（Ｃｏｎｆｉｄｅｎｔｉａｌｉｔｙ）、▲２▼完全性（Ｉｎｔｅｇｒｉｔｙ）、▲３▼可用性（Ａｖａｉｌａｂｉｌｉｔｙ）、▲４▼否認拒否（Ｎｏｎ－Ｒｅｐｕｄｉａｔｉｏｎ）を維持することと定義されている。一方、ネットワークに対して想定される代表的脅威としては、盗聴、漏洩、なりすまし、改ざん／偽造、不正侵入／不正アクセス、横取り、事実の否認、破壊などである。

【０００４】

また、ネットワークセキュリティのための要素技術として、秘匿・保全技術、認証技術、鍵配送技術、否認拒否技術、第三者信用機関、アクセス管理、セキュリティ監査、セキュリティ評価基準などがある。
ネットワーク・システムを介した情報通信を行うとき、そのシステムを誰がどのように利用したかということを確認したり、制御、管理することはセキュリティを維持する上において重要であり且つ必須である。システム内で起こる大方のイベントは、情報通信に関わる特定の実体（エンティティ）に起因しているはずであり、従ってそれらの認識はセキュリティ確保の基本であると言える。

【０００５】

認証とは、情報通信に関与した実体（エンティティ：人間、人間の代理として機能するプロセス、ソフトウエア、ハードウエア、通信データ等）が正当なものであるか否かを確認することであると考えられる。一般的には、認証する実体別に第１図のように分類することができる。
エンティティ認証は情報通信に関わる実体、例えばメッセージの送受信等の正当性を確認することであり、一方、メッセージ認証は、それら送受信メッセージの正当性を確認することであると言える。尚、エンティティ認証は利用者認証と呼ばれることもある。

【０００６】

エンティティ認証機構は、エンティティ識別処理とエンティティ認証処理に分けられる。前者は、システムの利用者が誰であるかを識別するもので、後者はその利用者が正当な本人であるか否かを確認する処理である。前者には、一般的に利用者識別名（Ｕｓｅｒ－ｉｄ）等が用いられるが、これは公知の識別子であり、本人だけが持ち合わせる情報（パスワードや暗証番号等）を用いた本来の認証処理は、後者の処理に委ねられる。

【０００７】

以下のエンティティ認証機構は、このエンティティ認証処理について記述している。
エンティティ認証機構には、認証に用いる情報のあり方により、大きく、知識利用、暗号利用、所有物利用、生体特徴利用の４つに分類できる。これらを順に説明する。

【０００８】

〈知識利用〉
知識利用によるエンティティ認証とは、エンティティを認証するために必要な情報を予め登録しておき、認証されるべきエンティティがその情報を知っているか否かでそのエンティティの正当性を確認する方法である。個人認証等で最も良く用いられているのが「パスワード」や「暗証番号」或いは「その個人にしか知り得ない情報（住所、生年月日等）」である。

【０００９】

大抵のシステムでは、「パスワード」により利用者認証を行っている。このような知識利用によるエンティティ認証は導入が比較的簡単で有効であるが、覚え易い文字列を使用したり、人目に付き易いところにメモしがちで容易に他人に見破られたり、通信中に盗聴されたりする危険性が高い。また、パスワード送信時に暗号化しても毎回同じパスワードであればそれをそのまま盗用し、再利用すること（リプレイ攻撃）で「なりすまし」が可能である。さらに、サーバ側のパスワード・ファイル（通常利用者のパスワードをキーとして暗号化された保存されている）が辞書攻撃によって破られる可能性もある。

【００１０】
これらの脅威に対抗するためには、毎回パスワードを変更する等の工夫が必要となる。そのため知識利用のエンティティ認証においては、例えば、ワンタイム・パスワード方式やチャレンジ・レスポンス方式等のような一方向性関数や乱数を利用した、高度な一回限りのパスワード方式が考案されている。
以下に各々の方式について述べる。
（１）ワンタイム・パスワード方式
文字通り一回限りのパスワード認証方式で、Ｂｅｌｌｃｏｒｅ Ｃｏ．Ｕ．Ｓ．Ａによって提唱され、インターネット標準としてもＲＦＣ化されている（ＲＦＣ－１９３８）。以下に、最も有名なＳ／Ｋｅｙ方式の処理概要を説明する。
【００１１】
Ｓ／Ｋｅｙ方式は、Ａをクライアント、Ｂを認証サーバとすると、
１： 一方向性乱数ｆを準備する。
２： Ａは秘密の乱数Ｒと公開の種と呼ばれる任意の数値Ｓを生成する。
３： Ｑ＝Ｒ＋Ｓとし、ｆ（Ｑ）， ｆ（ｆ（Ｑ））， ｆ（ｆ（ｆ（Ｑ））），…を計算し、それらをX_1，X_2，X_3，…，X_{100}，X_{101}とする。
【００１２】
４： ＡはX_1，…，X_{100}およびＲを秘密に保持し、ＢにはX_{101}を何らかの方法（オフライン）で渡し、Ｂはそれらを保持する。
５： ＡがＢに初めてログインする際、パスワードとしてX_{100}をＢに送信する。
６： Ｂはｆ（X_{100}）を計算し、保持していたX_{101}と比べる。もし、一致すればログインを許可し、一致しなければログインを拒否する。ログインが許可された場合、ＢはX_{101}を捨て、X_{100}を保持する。
【００１３】
７： Ａが次にログインするときは、次のパスワードX_{99}を使用する。Ｂ側の以降の処理は同様に行われる。
Ｓ／Ｋｅｙ方式の長所として、
・一回限りのパスワードなので、通信途上で第三者が盗聴しても再利用が不可能である。
【００１４】
・サーバＢのファイル上に保持しているパスワードは、次回ログイン時のパスワードを検査するためのものであり、これが盗まれても支障はない。
・関数ｆは一方向性関数なので、X_nが盗聴されてもX_{n-1}が計算できない。従って、ｆが第三者に知られても支障ない。
しかし、Ｓ／Ｋｅｙ方式の短所として、
・上の場合で、１００個パスワードを使い切ると、サーバの認証プログラムを再初期化する手間が必要である。
【００１５】
・実際のシステムでは上記再初期化をオンラインで可能なように、サーバ側では常に乱数Ｒを保持する必要がある。即ち、再初期化時、クライアントは以前とは異なった種Ｓ’のみをサーバにオンラインで送信し（Ｓ’は盗聴されても問題ない）、サーバは保持していたＲを用いて新たにＱ’＝Ｒ＋Ｓ’を計算し、これから新たなX'_{101}を生成する。このため、第三者がなんらかの方法でサーバに侵入したり、或いはサーバ管理者が悪意でこの乱数を得ると、パスワードが生成できクライアントＡになりすますことが出来る。
（２）チャレンジ・レスポンス方式
これは、パスワード認証における盗聴対策の一種で、代表的なものに、ＣＨＡＰ（Ｃｈａｌｌｅｎｇｅ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｐｒｏｔｏｃｏｌ，ＲＦＣ－１３３４）方式がある。このＣＨＡＰ方式において、認証要求者Ａが認証者Ｂに認証してもらう手順は第２図の通りである。
【００１６】
チャレンジ・レスポンス方式は、チャレンジが毎回変化するので、第三者が図中▲２▼の

メッセージを盗聴していても再利用が不可能であるとの長所がある反面、Ｂ上にＡのパスワードが保持されているので、Ｂの管理者自身がそれを悪用し、クライアントＡになりすまして不正を行うことができるという短所を指摘されている。

【００１７】

〈暗号利用〉

エンティティ認証に暗号を利用するとは、暗号技術を用いて当事者以外には偽造が困難な認証情報を生成し、それを当事者同士が交換・検査することにより当事者（エンティティ）の正当性を確認する技法である。

（１）デジタル署名

デジタル署名は、従来の書面取り引きにおける署名や印鑑による本人確認を電子媒体上で行う機構で、機能的には次の３条件を満たすことが要件であると考えられている。

【００１８】

▲１▼署名文が第三者によって偽造できない。

▲２▼署名文が受信者によって偽造できない。

▲３▼署名文の内容およびそれを送った事実を送信者が後で否定できない。

現状では、▲２▼および▲３▼の要件を満たすために公開鍵暗号方式の利用が必須である。公開暗号方式は、１９７６年にスタンフォード大学のディフィ（Ｄｉｆｆｉｅ）とヘルマン（Ｈｅｌｌｍａｎ）によって発表された概念で、一対の暗号化鍵と復号化鍵とが異なり、復号化鍵のみを秘密に保持し、暗号化鍵は公開して構わない。そのために、鍵の配送が容易であること、秘密に保持する鍵の種類が少なくて済むこと、認証機能（デジタル署名）を有すること、等の特徴を有するといわれている。公開鍵暗号方式の一般モデルを第３図に示す。

【００１９】

この公開鍵と秘密鍵の関係を逆にするとデジタル署名機能となる。即ち、送信者のみが知る秘密鍵で平文を暗号化し受信者に送信する。また、受信者は、送信者の公開鍵で復号化し平文を得る。この場合、暗号化鍵は送信者しか知らないので、暗号文が第三者および受信者によって偽造できないことになる。また、暗号化鍵を持つ本人にしか平文の内容を暗号化し送信することができないため、後になって暗号文の内容およびそれを送った事実を送信者が否定できず、上述のデジタル署名の要件を満たす。

【００２０】

現在、この公開鍵暗号の概念を実現した世界で最も有力なアルゴリズムとして、ＭＩＴのＲｉｖｅｓｔ、ＳｈａｍｉｒおよびＡｄｌｅｍａｎによって開発され、それぞれの頭文字を採って命名されたＲＳＡ暗号がある。尚、国際的に標準化されつつあるデジタル署名方式としては、次の２つがある。

・認証子照合法（ｗｉｔｈ　ａｐｐｅｎｄｉｘ）－－ＩＳＯ／ＩＥＣ　ＣＤ　１４８８８　ＰＡＲＴ１／２／３（Ｓｅｐ　２１，１９９５）

・通信文復元法（ｇｉｖｉｎｇ　ｍｅｓｓａｇｅ　ｒｅｃｏｖｅｒｙ）－－ＩＳＯ／ＩＥＣ　９７９６：１９９１（Ｅ）

実際に広く利用されているのは前者の認証子照合法であり、その概要を第４図に示す。

【００２１】

デジタル署名を用いて受信者が送信者の正当性を検証するためには、送信者の公開鍵が真の差出人のものである保証が必要である。例えば、物理的印鑑が正当なものであることを証明する印鑑登録証に相当するものがデジタル署名に必要となる。この保証のために信頼出来る第三者による公開鍵証明書制度が設けられ、その発行機関はＣＡ（Ｃｅｒｔｉｆｉｃａｔｉｏｎ　Ａｕｔｈｏｒｉｔｙ）と呼ばれる。ＣＡはインターネット標準（ＲＦＣ１４２１－１４２４）として制定され、公開鍵証明書の発行と管理を行う。

【００２２】

証明書のフォーマットは、国際標準（Ｘ．５０９→ＩＳＯ９５９４－８）として制定されており、既に、Ｘ．５０９は第三版が出ており、今後それに対応したＩＳＯ標準も制定される予定である。証明書は、利用者の識別子、利用者の公開鍵、証明書の有効期限、シリ

アル番号、発行機関名、発行機関のディジタル署名等の項目からなり、これらの後に当該ＣＡの電子署名が付される。

【００２３】

第４図の例において、送信者Ａは送信本文およびそれに施したＡのデジタル署名と共に、このＡの公開鍵証明書をＢに送信する。受信者Ｂはまずこの公開鍵証明書のＣＡによるデジタル署名を検査することにより、Ａの公開鍵証明書の正当性を確認する。これが正当であればＢは正当なＡの公開鍵を入手できたことになる。この後、ＢはＡのデジタル署名を検査することにより送信者認証を行う。

【００２４】

認証子照合法の長所として、厳格なＣＡが存在し、且つ送信者が自身の秘密鍵を厳密に保持できれば、第三者による「なりすまし」は一般的に困難であるとの点が指摘されている。しかしながら、ネットワークを介したリモートログインにおいて、署名をリモートログインのためのパスワードとして使用した（即ちデジタル署名を相手認証情報として用いる）場合には、第三者がそれを盗聴しそのまま再利用する（リプレイ攻撃）ことで「なりすまし」が可能であるという短所もある。

（２）デジタル署名付認証トークン方式

これは（１）の方式のリプレイ攻撃に対する強度を改善したものと言える。第５図に、デジタル署名付認証トークン方式の処理の概略を示す。

【００２５】

即ち本方式の前提として、クライアントＡはＣＡの秘密鍵によってデジタル署名されたＡの公開鍵証明書を、またサーバＢはＣＡの公開鍵を保持しているものとする。

この状態においてクライアントＡは認証情報（以下、認証トークンと称する）として、次の▲１▼から▲４▼から組み立てられたものをサーバＢに転送する。この認証トークンには、トークン作成時のタイムスタンプＴが含まれている。

【００２６】

▲１▼：Ａの公開鍵証明書（Ｃａ）

▲２▼：タイムスタンプ（Ｔ）

▲３▼：受信者ｉｄ：ＢのＥ－Ｍａｉｌアドレス等

▲４▼：▲２▼＋▲３▼のデジタル署名（Ｓａ）

この認証トークンを受信したサーバＢは先ず署名の検査を行い、タイムスタンプＴ等が改ざんされていないことを確認の上、このＴと現在時刻とを比較する。もし比較結果がほぼ等しければクライアントＡのログインを許可する。

【００２７】

しかし、Ｔが一定時間以上過去の時刻であれば、この認証トークンがＡおよびＢ以外の第三者によって再利用（リプレイ攻撃）されているものと見做してログインを拒否する。

このトークン方式は、厳格なＣＡが存在し、且つ送信者が自身の秘密鍵を厳密に保持できれば、第三者による「なりすまし」はかなり困難であるとの長所がある反面、一定時間内であれば、盗聴した認証トークンをそのまま再利用すること（リプレイ攻撃）で「なりすまし」が可能であるとの短所も有している。

（３）ＳＳＨ（Ｓｅｃｕｒｅ　ＳＨｅｌｌ）方式

ＳＳＨ方式はＵＮＩＸにおけるリモートログインのためのｒｓｈ／ｒｌｏｇｉｎなどｒ系コマンドプロセスに対するセキュリティパッケージであり、インターネット・ドラフトとして検討されている。認証処理に関する部分を以下に示すが、基本的には共通鍵暗号と公開鍵暗号を併用したチャレンジ・レスポンス認証方式である。

【００２８】

第６図は、クライアントＡがサーバＢにログインする際のシーケンスである。同図において、共通鍵暗号（ＤＥＳ、ＩＤＥＡ等）用のセッション鍵を共有するためのフェーズ（▲２▼、▲３▼）と認証処理を行うフェーズ（▲４▼、▲５▼、▲６▼）に分かれている。

処理シーケンスは次のようである。

▲１▼クライアントＡはサーバＢにログイン要求を送る。

【００２９】
▲２▼このログイン要求に基づき、サーバＢはセッション鍵共有のため自身の公開鍵、乱数等をクライアントＡに送る。
▲３▼クライアントＡはセッション鍵を生成し、それをサーバＢの公開鍵で暗号化してＢに送る。サーバＢがこれを受信した時点で、クライアントＡとこの間にセッション鍵を共有できたことになるので、▲４▼以降、ＡＢ間のメッセージは全てこのセッション鍵で暗号化してやり取りされる。
【００３０】
▲４▼クライアントＡは、自身の公開鍵、ユーザ名をサーバＢに送る。
▲５▼サーバＢは、クライアントＡの公開鍵とユーザ名とが登録されていることを確認の上、認証のためのチャレンジ（乱数）を生成し、それをＡの公開鍵で暗号化してクライアントＡに送る。
▲６▼クライアントＡは上記チャレンジのハッシュ値を計算し、それをチャレンジ・レスポンスとしてサーバＢに送る。
【００３１】
▲７▼サーバＢは、▲６▼で受けたチャレンジ・レスポンスの値と保存してあったクライアントＡ向けチャレンジのハッシュ値とを比較し、それが同値であればＡのログインを許可し、異なっていればログインを拒否する。
ＳＳＨ方式の長所は、チャレンジ・データが毎回変化するので、第三者が▲６▼のメッセージを盗聴しても再利用による「成りすまし」が不可能であるというものであるが、短所として、
・サーバＢの管理者自身が悪意でクライアントＡの公開鍵情報を書き換えることにより、クライアントＡになりすまして不正を行うことが可能である、
ということが指摘されている。
（４）ＰＲＣ認証方式
このＰＲＣ（Ｒｅｍｏｔｅ　Ｐｒｏｃｅｄｕｒｅ　Ｃａｌｌ）認証方式は、ＵＮＩＸの分散環境システムでよく用いられる遠隔手続き呼び出し機能であり、セキュリティ機能としてユーザ認証機能が用意されている。
【００３２】
このＲＰＣ認証は、ＲＰＣ手続きの発行者が誰であるか（エンティティ認証機能）、そしてその発行者の権限はどのくらいであるか等をサーバが確認する機能を備えている。このＰＲＣ認証が持つエンティティ認証機能の概要は第７図のようであり、その手順の概略を述べる。
▲１▼通信に先立ち、まずクライアントとサーバはＤＥＳ暗号に用いる共通鍵（K_{ab}）をＤＨ法（Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎ型公開鍵配送法）により共有する。ＵＮＩＸの世界では、ＤＨ法に用いる公開鍵と秘密鍵とは、ＮＩＳ（Ｎｅｔｗｏｒｋ　Ｉｎｆｏｍａｔｉｏｎ　Ｓｅｒｖｉｃｅ）によって管理され、各ユーザは通信に先立ってこのＮＩＳから予め登録してある通信相手の公開鍵と自身の秘密鍵とを入手し、それから共有鍵（ＤＥＳ鍵）を計算により得る。
【００３３】
▲２▼クライアントでは、次の手順で認証情報を作成しサーバに送信する。
（Ｉ）送信者を表す文字列（ネットネームと呼ばれる）を生成する。ＵＮＩＸの場合、
ｕｎｉｘ．＜ユーザｉｄ＞@＜ホスト・アドレス＞
という形式を有する。
【００３４】
（ＩＩ）セッション鍵（乱数：Ｋ）を生成する。
（ＩＩＩ）タイムスタンプ（現在時刻：Ｔ）をセッション鍵（Ｋ）でＤＥＳ暗号化する（T_e）。
（ＩＶ）セッション鍵（Ｋ）を共有鍵（K_{ab}）でＤＥＳ暗号化する（K_e）。
認証情報として（Ｉ）のネットネーム、（ＩＩＩ）の暗号化されたタイムスタンプ（T_e

）、（ＩＶ）のセッション鍵（K_e）等をサーバに送信する。
【００３５】
▲３▼サーバは、受信した認証情報の中の暗号化されたタイムスタンプ（T_e）を復号化し（Ｔ）、それを現在時刻と比較することによりネットネームの正当性を検証する。即ち、Ｔと現在時刻との差が許容範囲内であればそのネットネームのアクセス要求を許可するが、許容範囲外であれば拒否する。
ＲＰＣ認証方式の長所として、クライアントおよびサーバの各々が自身の秘密鍵を厳密に保持し、且つ正当な相手の公開鍵を確実に得ることができれば、第３者による「なりすまし」は一般的に困難といわれているが、一定時間内であれば盗聴した認証情報をそのまま再利用すること（リプレイ攻撃）で「なりすまし」が可能であるとの短所も有する。
（５）Ｋｅｒｂｅｒｏｓ（ＲＦＣ１５１０）方式
Ｋｅｒｂｅｒｏｓは、ＭＩＴのＡｔｈｅｎａプロジェクトで開発された利用者認証システムであり、１９７８年にＲ．ＮｅｅｄｈａｍとＭ．Ｓｃｈｒｏｅｄｅｒによって提案された「信頼された第三者期間による認証方式」に基づいている。ＯＳＦ（オープン・ソフトウェア財団）が定めた分散処理環境構築のためのソフトウェア・パッケージであるＤＣＥ（Ｄｉｓｔｒｉｂｕｔｅｄ　Ｃｏｍｐｕｔｉｎｇ　Ｅｎｖｉｒｏｎｍｅｎｔ）における認証サービスとして、このＫｅｒｂｅｒｏｓが採用された。
【００３６】
この方式では、通信の秘匿やユーザ認証など全て共通鍵暗号方式（ＤＥＳ）のみで実現している。
各ユーザの鍵を知っているのは各ユーザ自身と認証サーバだけであることを前提に、お互いの正当性を認証サーバで保証してもらうという方式を採用している。
【００３７】
認証サーバにあたる部分をＫｅｒｂｅｒｏｓサーバとＴＧＳ（Ｔｉｃｋｅｔ　Ｇｒａｎｔｉｎｇ　Ｓｅｒｖｅｒ：チケット発行サーバ）に分けて利用者のパスワードや鍵が利用者側のシステム（セキュリティレベルが低い）上に長時間保持されないように工夫している。また、チケット（Ｔｉｃｋｅｔ）とオーセンディケータ（Ａｕｔｈｅｎｔｉｃａｔｏｒ）という考えを導入して、さらに安全性を高めている。Ｋｅｒｂｅｒｏｓの認証方式を第８図に示す。
【００３８】
Ｋｅｒｂｅｒｏｓの認証方式は、各サーバ、利用者ＷＳ間のやりとりは全て暗号化され、さらに暗号化鍵は毎回乱数により発生しているため盗聴に強い点、目的サーバは利用者個々のユーザＩＤやパスワードを管理する必要は無く、それらはＫｅｒｂｅｒｏｓサーバだけが知っていればよい等が長所として指摘されているが、
・一定時間内であれば盗聴した認証情報をそのまま再利用し（リプレイ攻撃）可能。
【００３９】
・米国における暗号製品の輸出制限のため、暗号アルゴリズムとしてのＤＥＳが実装されたＫｅｒｂｅｒｏｓ製品は、日本で利用できない場合がある。
・認証サーバが各利用者の認証情報や暗号化鍵を集中管理するので、悪意の第三者がこの認証サーバへの侵入に成功するとその管理対称ドメインが全滅する。
・全てのマシン、アプリケーションはＫｅｒｂｅｒｏｓ対応が必要で、導入の手間が大きい、
等の短所も指摘されている。
（６）ゼロ知識対話証明方式
この方式は、１９８５年、ＭＩＴのＧｏｌｄｗａｓｓｅｒ，Ｍｉｃａｌｉおよびトロント大学のＲａｃｋｏｆｆにより提案されたもので、ある情報を持っていることをその内容を相手に示すことなく相手に納得させる方式であり、例えば、パスワードを提示することなく真のパスワードを知っていることを相手に証明できる等が利用例である。１９８６年にＦｉａｔとＳｈａｍｉｒによりファイアット・シャミア法が提案され、米国特許４，７４８，６６８号（特開平６３－１０１９８７号）。

【００４０】
クライアントＡ（証明者）がサーバＢ（検証者）へ秘密の情報Ｔ（パスワード等）を転送する場合のゼロ知識対話証明方式によるシーケンスを第９図に示す。ここで、ＡはＺ＝T^2　ｍｏｄ　ｎを完全に知り、ＢはＺとｎのみを知っているとする。ここで、ｎは大きな素数ｐ，ｑの合成数である。この場合、Ｂはｎを素因数分解できなければＴを得ることが極めて困難である。
【００４１】
以下の▲１▼～▲４▼をｋ回繰り返し（対話の所以）Ａの正当性を検証する。
▲１▼Ａは乱数Ｒを選び、Ｘ＝R^2　ｍｏｄ　ｎを計算し、ＸをＢに送る。
▲２▼Ｂはｂ∈｛０，１｝を二者択一的にランダムに選び、ｂをＡに送る。
▲３▼Ａは、Ｙ（Ｙとは、ｂ＝０の場合はＲであり、ｂ＝１の場合はＴＲ　ｍｏｄ　ｎである）をＢに送る。
【００４２】
▲４▼Ｂは、
Ｘ＝　Y^2　ｍｏｄ　ｎ　　ｂ＝０の場合
ＺＸ＝Y^2　ｍｏｄ　ｎ　　ｂ＝１の場合
が成立するかを検査し、これらが成り立てば検査ＯＫとする。ここで、▲３▼及び▲４▼でｂ＝０およびｂ＝１の場合に分けているのは、Ａになりすました悪意のクライアントＡ’はＴの値を知らなくても次のようにして検査に合格できるからである。即ち、
常にｂ＝１であるなら、Ａは▲１▼でＹの値として適当なＹ’を定め、Ｘ＝（Ｙ’）^{2}／Ｚｍｏｄ　ｎを計算し、このＸをＢに送る。次に▲３▼でＹ＝Ｙ’の値を送ると▲４▼の検査は当然合格する。また、この方式では、ｂの値を予想してから検査式を満たすＸとＹを計算できるので繰り返し１回当たりのなりすまし確率は１／２である。従ってこの手順をｋ回繰り返すとなりすまし確率を2^{-k}にできる。
【００４３】
この方式の長所は、事前に秘密の認証情報Ｔをサーバに教える必要がないので、サーバＢの正当な管理者であってもクライアントＡに成りすますことができないことであり、対話シーケンスが冗長である点、認証プロセスが複雑であり、パフォーマンスと認証精度がトレード・オフの関係となる点などが短所である。
〈生体特徴利用〉
次ぎに生体特徴（個人属性）を利用した従来のセキュリティについて説明する。
【００４４】
この手法は、本人の身体的、行動的特徴を認証情報として利用し、端末利用者の正当性を確認する技法である。身体的、行動的特徴としては次のようなものがある。
・身体的特徴
指紋、音声スペクトル、顔のパターン、手形、網膜パターン、耳の形
・行動的特徴
署名、筆記パターン、キーストローク
この方式は、本人にしか持ち得ない唯一の個人属性を認証情報として使用するので、認証が成功した場合の本人識別精度は高いが、正当な本人であるにも係わらず認証が失敗する等、認識確度が１００％ではなく、技術的な改善の余地があり、端末による利用者の認証等、オフライン認証（ローカル認証）では極めて有効であるが、ネットワークをまたがった認証（リモート認証）では盗聴により認証情報を再利用（リプレイ攻撃等）即ち「なりすまし」が可能となるなどの欠点がある。
【００４５】
所有物利用によるセキュリティについて説明する。
〈所有物利用〉
ある特定の物体が認証情報を保持しており、認証する側ではその認証情報を検証することにより、その物体を保持する人間やその物体に認証された人間、或いはその物体と連動して作動するソフトウエアやハードウエア等を正当なエンティティとして認証する。

【００４６】
所有物の例としては次のようなものがある。
・鍵、トークン、バッチ
・電子キー
・磁気カード
・ＩＣカード
・非接触型カード（光式、電磁波式などＩＣカードの発展型と言える）
例えば、端末のロックを解除するための鍵やトークン、電子キーを所持する人間は、その端末の正当な利用者として認証される。
【００４７】
しかし、これらの所有物の紛失や盗難による悪用を防止するため、ネットワークを介した認証では、磁気カードのように所有物でまず利用者識別を行い、更にサーバ（アクセス先のホストコンピュータ等）による暗証番号の検証により利用者の正当性を確認する等、「知識利用」の技法と組み合わせて用いられることが多い。
【００４８】
ＩＣカードではこれがさらに発展し、まずＩＣカード自身がＩＣカードを使用しようとしている人間を暗証番号で検証し、これが成功して初めてネットワークを介したサーバとの認証動作に入る。サーバによるＩＣカード（即ちＩＣカードにより検証された人間等のエンティティ）認証処理は「知識利用」や「暗号利用」の技法を利用して行われる。
【００４９】
この手法は、所有物を厳密に保持すれば第三者による「なりすまし」は一般に困難である点、ＩＣカードは通常、耐タンパー性（Ｔａｍｐｅｒ　Ｆｒｅｅ）を有しており、外部からメモリ内の情報を読み書き不可能な構成となっている。そのため暗号鍵やパスワード等個人に依存した情報を比較的安全に格納、管理できる点、またセキュリティ処理機能そのものをＩＣカード内に組み込むことにより、さらに安全な認証通信が可能となる点などが長所である反面、所有物利用による認証システムでは、大抵の場合、その所有物とクライアントとなる端末との間に専用の入出力機器が必要である点、磁気カード、ＩＣカード等でネットワークを介した認証処理の場合、認証シーケンスそのものは結局「知識利用」や「暗号利用」の技法を使用しているため、当然であるがそれらに特有の短所も付随することになる点などが短所として指摘されている。
【００５０】
上述したように、上記の各種エンティティ認証方式は長所を有する反面、短所も有する。
【００５１】
ところで、エンティティ認証が想定する直接の脅威は、パスワードなどの不正入手による「なりすまし」であるが、この「なりすまし」が成功して、一旦システムに侵入されると、データの改竄やファイル破壊、不正データの生成など様々な不正行為の脅威に晒される。また、このような脅威は外部からの不正アクセスによるもののみならず、システム管理者などの内部犯罪によって引き起こされる可能性もある。
【００５２】
従って、アクセスされる側のシステムにとって、アクセスしてくる実体が何であるかを確認するエンティティ認証は、セキュリティ上の脅威に対する最前線の防衛網と言え、その重要度はシステムの機密度に応じて大きくなる。
【００５３】
ここで、今まで述べてきたエンティティ認証方式の「なりすまし」に対する強度を外部および内部の不正エンティティに対してまとめてみると、図１０に示すようになる。
【００５４】
上記の何れの方式も欠点はあるものの、システムの環境、構成によっては充分実用的なものである。しかしながら、図１０に示すように、システム管理者などのシステムに精通した人間の悪意の内部犯罪による脅威に対しては防衛できないものがほとんどであり、たとえ防衛できる方式であっても認証処理が複雑になるなどの欠点がある。

【００５５】
【発明が解決しようとする課題】
上述したように、エンティティ認証はセキュリティ上の様々な脅威に対する最前線の防衛機能であるが、インターネット時代においては、その適用領域の広範さと相互接続性の観点から、導入が簡単で、従って仕組みが簡単で、かつ、脅威に対して充分に有効な認証方式が望まれる。そこで、上述した各種認証方式の長所、短所に対する検討を踏まえ、新たな認証方式に要求される事項をまとめると次のようになる。
【００５６】
（１）盗聴などによって盗まれた認証情報が第三者によって再利用できないこと。例えば、ワンタイム・パスワード方式（Ｓ／Ｋｅｙ等）はこの要件を満たしているが、ディジタル署名付認証トークン方式はそのタイムスタンプの許容時間内であれば盗聴トークンが再利用できる。
【００５７】
（２）認証サーバに認証情報が保存されないこと。
換言すれば、認証サーバは利用者個々の認証情報を保管する必要がなく、ただログイン時の認証情報が正当か否かを識別できる機能を有すればよい。これによって、たとえ悪意の第三者が認証サーバに侵入できたとしても利用者個々の認証情報を得ることはできない。
【００５８】
（３）認証シーケンスは極力簡単であること。
これにより、システムに対する負荷を最小限にし動作の安定性を得る。従って、チャレンジ・レスポンス方式やゼロ知識対話証明方式のような対話シーケンスを用いない。
【００５９】
（４）認証情報は毎回異なり、しかもその情報は無限に存在すること。
これにより、上記（１）の用件を満足しつつ、既存ワンタイム・パスワード方式（Ｓ／Ｋｅｙ等）のようにパスワードを使い切った場合に、再び初期情報をサーバに再登録するといった定期作業が不要になる。
【００６０】
（５）生体特徴利用のような特殊外部測定機器を必要としないこと。
特殊な機器はインターネットを介した相互運用性を損ない、また導入コストの高騰につながるので、このような外部機器は用いない。
【００６１】
かくして、本発明は、簡単な手順による認証方法を用いて、たとえ認証情報などが第三者に盗まれても、盗まれた認証情報などの第三者による再利用を困難にすることを目的とする。
【００６２】
【課題を解決するための手段】
本発明は、前記の目的を達成する一手段として、以下の構成を備える。
【００６３】
本発明にかかる認証方法は、認証に先立ち、端末装置から送られてくる認証情報を検査するための第一の検査情報をサーバ装置のメモリに保存し、前記サーバ装置に接続された端末装置の認証要求に対して、ワンタイムパスワード方式および公開鍵暗号方式による認証を継続して行う認証方法であって、端末装置から認証要求を受信すると、前記端末装置へ認証情報要求を送信し、前記認証情報要求に応じて、前記端末装置に関連する秘密鍵を用いた暗号化により種情報から生成された認証情報を前記端末装置から受信し、前記認証情報を前記端末装置に関連する公開鍵によって復号することにより第二の検査情報を生成し、前記第二の検査情報と前記メモリに保存された前記端末装置に対応する第一の検査情報とを比較して、前記第一および第二の検査情報が一致する場合に前記認証要求を許可し、前記認証要求を許可した場合、前記認証情報を、次回の認証用として前記端末装置に対応する第一の検査情報に代えて前記サーバ装置のメモリに保存し、前記認証要求が許可された場合、前記認証情報は、次回の認証要求のための種情報として前記端末装置のメモリに

保存され、前記認証要求が許可されなかった場合、前記認証情報は、前記端末装置のメモリおよび前記サーバ装置のメモリに保存されることなく、破棄されることを特徴とする。

【００６４】

本発明にかかる記憶媒体は、サーバ装置を制御して、上記の認証を実行するプログラムコードが格納されたことを特徴とする。

【００６５】

本発明にかかる認証装置は、認証に先立ち、端末装置から送られてくる認証情報を検査するための第一の検査情報を保存するメモリを有し、端末装置の認証要求に対して、ワンタイムパスワード方式および公開鍵暗号方式による認証を継続して行う認証装置であって、端末装置との間で認証に関連する情報の送受信を行う通信手段と、前記メモリおよび前記通信手段を利用して、認証処理を行う認証手段とを備え、前記認証手段は、端末装置から認証要求を受信すると、前記端末装置へ認証情報要求を送信し、前記認証情報要求に応じて、前記端末装置に関連する秘密鍵を用いた暗号化により種情報から生成された認証情報を前記端末装置から受信し、前記認証情報を前記端末装置に関連する公開鍵によって復号することにより第二の検査情報を生成し、前記第二の検査情報と前記メモリに保存された前記端末装置に対応する第一の検査情報とを比較して、前記第一および第二の検査情報が一致する場合に前記認証要求を許可し、前記認証要求が許可された場合、前記認証情報は、次回の認証用として前記端末装置に対応する第一の検査情報に代えて前記メモリに保存されるとともに、次回の認証要求のための種情報として前記端末装置のメモリに保存され、前記認証要求が許可されなかった場合、前記認証情報は、前記端末装置のメモリおよび前記サーバ装置のメモリに保存されることなく、破棄されることを特徴とする。

【００６９】

【実施の形態】

以下添付図面を参照しながら、本発明の好適な実施形態乃至実施例を説明する。

第１１図は本発明にかかる方式が適用されるネットワークの構成を示す

このネットワークでは、複数のクライアント２００，３００…がインターネットによって接続されている。また、このネットワークに認証サーバ１００も接続されている。

【００７０】

クライアント２００がクライアント３００と通信するときは、クライアント２００が認証要求者となり、クライアント３００が認証者となる。本実施形態では、認証者をサーバと呼ぶ。

認証サーバ１００は、複数のクライアントからアクセス可能なデータベースを有するもので、それらクライアントからの認証要求を受けて認証を行うもので、認証サーバと呼ぶ。第１２図を参照。即ち、クライアントとクライアントとが通信を行うときは、一方がサーバとして振る舞う。

【００７１】

本実施形態の認証方法は、本質的には認証局（ＣＡ）の存在を前提としない。クライアントとクライアント間のデータの送受は、認証局（ＣＡ）の介在を必要としなくて直接行われることもあり、認証サーバ１００（例えば、ＣＡ等）を介して行う場合もある。認証者も認証要求者も、人そのものではなく、オペレータ或いはユーザの行為を媒介にして動作するコンピュータ（或いはシステム）である。

【００７２】

第１３図は、認証要求者としてのクライアントＸと認証者として認証サーバＹとの単純化した構成からなるネットワーク（第１１図）における、本発明を適用した認証アルゴリズムの例を示す。第１３図の例では、前提として、公開鍵暗号アルゴリズムを使用する。クライアントＸは自身の秘密鍵K_Sを、またサーバＹは、そのクライアントの秘密鍵K_Sに対応する公開鍵K_P、並びにその公開鍵の証明書CK_pを保持しているものとする。また、S_eは公開鍵暗号アルゴリズムの暗号化関数、S_dは公開鍵暗号アルゴリズムの復号化関数を意味する。

【００７３】

本システムでは、第１３図に示すように、クライアント側は認証情報生成種データファイル２０４を有し、サーバ側はクライアント認証情報検査データファイル１０５を有する。認証情報生成種データファイル２０４は、認証情報を生成するための種となるデータを記憶するファイルである。ここで、本システムでは、「認証情報」は、認証要求者が認証者に認証を要求するために認証要求者が認証者に送る情報を意味し、クライアント側において種データから生成される。この入出力情報はサーバ側において、サーバが有するそのクライアントの検査で田堵照合し、照合がとれれば、そのクライアントを真正な認証要求者と見なすというものである。

【００７４】

第１４図は、サーバＹが有する認証情報検査データファイルの構成を有する。即ち、サーバＹは、各クライアント毎に、「認証情報検査データＤ」と「公開鍵K_P」と「公開鍵証明書CK_p」とを有する。第１４図の例では、サーバＹは、クライアントＸについて検査データD_Xと公開鍵K_{PX}とを有し、クライアントＷについて検査データD_Wと公開鍵K_{PW}とを有する。

【００７５】

第１５図は、本実施形態の認証を実現するために、クライアントＸ及びサーバＹそれぞれにおける処理手順と、これらの間で行われる連絡の手順を示す。第１３図及び第１５図に従って、本実施形態の手順を、クライアントＸがサーバにログインする際の認証を受けようとする場合について説明する。

〈初期情報の登録〉

本実施形態では、ログインに先立って、クライアントは初期種データD_{S0}を設定し、サーバＹにおいて初期検査データD_{s0}を初期的に登録することが必要である。これらの登録は、最初に一回だけ行えば良く、一旦行ってしまえば、その後に登録することは不要である。

【００７６】

この登録作業は、クライアント側ではクライアント自身が行い、サーバ側においては一般的にクライアントのアクセス権限の設定等を伴うので相応の権限を持ったシステム管理者が行うことが好ましい。初期種データD_{S0}は、乱数やクライアントのＥ－ｍａｉｌアドレスや利用者識別名等何でもよい。また、秘密鍵K_sさえ秘密に保たれているならば、初期種データD_{S0}を特に秘密にしておく必要もない、登録後は、登録された旨がクライアントに通知される。

【００７７】

後述するように、種データＤはクライアントにおいて認証情報の生成のために使われる。そして、その認証情報を用いた認証要求が一旦受け入れられると、生成された認証情報は次のログインのための認証要求のための認証情報生成用の種データとして記憶される。また、サーバ側では、受信した認証情報と前もって保存している検査データＤとを比較して照合が得られたならば、その受信した認証情報を、そのクライアントからの次のログインのための検査データとして保存する。従って、本システムでは、認証情報生成種データファイル２０４に記憶されている種データと、サーバ側の検査データファイル１０５に記憶されている検査データとは値として一致しているので、第１３図においては、便宜上、D_{n-1}として表している。種データや検査データを一般的にD_{n-1}と表したのはそれらのデータが前回のログインにおいて生成されたものであるからである。

【００７８】

第１３図の例では、クライアントＸの初期種データはD_{S0}として登録されている。

本実施形態の認証プロトコルは、クライアントＸは、始めての認証セッションでは、認証情報をこの初期種データD_{S0}から生成する。認証が許可された場合には、今まで保存していた種データD_{n-1}をクライアントＸの秘密鍵K_Sで暗号化し、それを次回の認証セッションのための種データD_nとして保存する点に大きな特徴がある。尚、前回の種データD_{n-1}は次回以降のログインでは使用されることはないが、履歴の保持のために保存しておいても良い。

【００７９】
以下、順に、第１３図および第１５図に則して、本実施形態の処理手順を説明する。
・ステップ▲１▼
本実施形態では、認証はログインを行おうとするクライアントが真正なクライアントであるか否かを認証するものである。従って、認証に先立ってサーバへのログインが行われる。本実施形態でのログインは、利用者識別名（Ｕｓｅｒ－ｉｄ等）をサーバＹに送ることによって行われる。このログインメッセージは平文のままでも、暗号文の形式でも良い。
【００８０】
・ステップ▲２▼
ログインメッセージを受け取ったサーバＹは、クライアントＸに対して認証情報要求メッセージを送る。
・ステップ▲３▼
この認証情報要求メッセージを受け取ったクライアントＸは、サーバに返すべき認証情報として、自身が保存している種データＤを自身の秘密鍵K_Sで暗号化してサーバＹに送る。
【００８１】
第１３図の例は、初期登録後の始めて認証セッションの開始であるので、種データはD_{S0}であり、従って、データD_{S0}をクライアントＸの秘密鍵K_Sで暗号化したものD_1がサーバＹに送られる。
・ステップ▲４▼
サーバＹは、クライアントＸから認証情報D_1を受信すると、既に得ているクライアントＸの公開鍵K_Pで復号化する。前述したように、本実施形態の認証情報D_nは公開鍵暗号化アルゴリズムに従って暗号化されている。即ち、クライアントＸを表す筈の認証情報D_1が、クライアントＸの真正な種データD_{S0}をそのクライアントＸの秘密鍵K_Sによって暗号化されたものであれば、その認証情報D_1を公開鍵K_Pによって復号化したものは、公開鍵暗号化アルゴリズムに従えば、クライアントＸの秘密鍵K_Sによって暗号化される前の種データD_{S0}に一致するはずである。
【００８２】
・ステップ▲５▼
それで、サーバＹは、復号化して得た情報D_{S0}と、ファイル１０５から読み出したところのクライアントＸの検査データD_{S0}とを比較照合する。
・ステップ▲６▼
サーバは照合結果をクライアントに返す。
【００８３】
前述したように、照合が一致した場合は、認証を要求したクライアントＸは真正なクライアントであることを意味するから、ログインを許可する旨のメッセージを返す。
また、次回のクライアントＸからのログイン要求に備えて、クライアントＸから受け取ったところの暗号化されている認証情報D_1をファイル１０５内に保存する。サーバＹにおけるこの認証情報の更新（上書き保存）は、ステップ▲５▼における比較結果が一致したときのみ行われる。ファイル１０５内に書き込まれた暗号化認証情報D_1はファイル１０５内では次回のログインのための検査データとして記憶される。
【００８４】
・ステップ▲７▼
サーバからの認証処理結果を受けたクライアントでは、その認証処理結果が許可かまたは拒否であるかを判断する。
・ステップ▲８▼
認証が許可されたものであった場合には、サーバ側に送っていた認証情報D_1を次回のログイン時の種データD_1としてファイル２０４に記憶する。
【００８５】
認証が拒否されたものであった場合（処理結果が所定時間以内に返ってこなかった場合も

含む）には、認証情報D_1を次回のログイン時の種データD_1として使うことはできないので、破棄する。換言すれば、ログインを再試行する場合には、クライアントは種データD_{S0}から認証情報D_1を再度生成する。
以上が、始めてログインが行われたときにおける、ログイン要求に対する認証のための処理手順である。
【００８６】
次回にログインが行われたときには、ステップ▲１▼～▲８▼が繰り返される。
即ち、第１３図に示すように、クライアントＸは、サーバＹからの２回目の認証情報要求に対しては、保存しておいた種データD_1を認証情報としてその秘密鍵K_Sで暗号化して生成し、この暗号化した認証情報D_2をサーバＹに送る。サーバＹは送られてきた認証情報D_2を公開鍵K_Pで復号化して検査データD_1を生成し、この検査データD_1を格納しておいた検査データD_1と比較する。比較の一致がとれれば、ログインを許可する点では、第一回目のログイン時と同じである。
【００８７】
この方式は、一回に限り有効な認証情報を無限に生成できるので、以降これを「無限ワンタイム認証方式」と呼ぶことにする。
この無限ワンタイム認証方式の従来方式に比べ特に強調されるべき利点は次のようである。
（１）次回ログイン時に生成される認証情報は、正当な認証要求者のみが彼の保持する秘密鍵を用いて生成することができるのであり、外部の第三者盗聴者のみならず、サーバの認証情報管理者でさえ、その次回のための認証情報を知ることが出来ない。このことにより、サーバ側の内部悪意者による利用者への「なりすまし」による不正行為、即ち内部犯罪をも防ぐことが可能である。
【００８８】
即ち、認証要求者は、生成種データ（前回ログイン時に使用した認証情報）を自身の秘密鍵で暗号化したものを、認証情報として認証者に送り、認証者は認証要求者から受信した認証情報をその認証要求者の公開鍵で復号化し、認証者側で保存していた検査データと比較して一致したときのみ、認証要求に対して許可を行う。従って、自身の秘密鍵を厳格に保管している限り、認証情報、検査データ、認証情報生成種データのいずれか（或いは全て）が第３者に知られることとなっても、その第３者による当該クライアントのなりすましは不可能である。
【００８９】
さらに、認証者では、１つの認証要求処理プロセスにおいて、検査データ同士の比較を行い、一致しないことが或いは一致したことが確認されるまではその処理プロセスから抜けることはなく、また、一致がとれれば直ちに、検査データの更新を行うので、検査データの更新までのタイムラグは実質的に零であり、従って、第３者が、送信中の認証情報を盗聴し、それをそのまま再利用して認証要求者に成り済ますことの猶予時間は実質的に零である。
（２）認証情報の登録は一回限りで済み、一旦、登録を行えば、クライアントは無限にセキュリティの高い認証情報を生成することが出来る。但し、秘密鍵、公開鍵のペアーを変更した場合は、再度、サーバに登録する必要がある。
（３）クライアントーサーバ間の認証処理は対話シーケンスを持たず、ログイン時に１メッセージ（認証情報）を送信するのみである。従って、サーバ側及びクライアント側で必要とされるプログラムは極めて簡単なものとなる。
（４）クライアント側で認証情報をサーバ側に送ってから、その認証情報を次の認証情報に更新する（即ち、D_nからD_{n+1}に更新）迄の時間間隔が零に等しい。したがって、たとえ通信中に認証情報が盗聴されても、盗聴者がそれを再利用し得る時間的隙間が皆無である。
【００９０】
これに対し、既存方式で認証情報の一部にタイムスタンプを用いるような方式では、サー

バ側で一定の時間許容範囲を設けているため、認証情報を盗聴後即時にその許容範囲時間内で再使用すれば真のクライアントに成りすましてサーバにログインできる（リプレイ攻撃）タイミングが存在し得るが、本方式ではこれが不可能である。
（５）サーバ側において、検査データD_nをサーバ管理者などの内部関係者が盗用し、偽りの認証を試みたとしても、これらの者が用いたD_nは、認証プロセスの中で真正な認証要求者の公開鍵により複合化され生成されたD_{n-1}と比較されるので、認証が成功することはない。即ち、サーバの認証情報を知ることができる内部関係者であっても、真正な認証要求者に成りすますことはできないのである。
【００９１】
【実施例】
上述の無限ワンタイム認証方式を具体化した実施例を以下に説明する。
第１６図は、この実施例のためのサーバ側構成を示す。
このサーバは、ＯＳ１０１として、例えばＷＩＮＤＯＷＳまたはＭＡＣ　ＯＳまたはＵＮＩＸまたはＮＥＴＷＡＲＥを用いる。ネットワーク１０２との通信プロトコルは例えば、ＴＣＰ／ＩＰやＯＳＩやＮＥＴＷＡＲＥを用いる。
【００９２】
検査データファイル１０５は第１４図に関連して説明したファイルの構成を有し、具体的には、クライアントの識別名情報Ｘと、検査データD_{n-1}、公開鍵証明書CK_{px}とを記憶する。公開鍵証明書CK_{px}は、バージョン番号、シリアル番号、発行局名、証明書の有効期限、ユーザ識別子、公開鍵と関連情報等を含む。公開鍵ファイル１０７は、証明機関ＣＡの公開鍵K_{pc}を保存する。この公開鍵K_{pc}はクライアントＸの公開鍵証明書に付されているディジタル署名を検査するのに使用する。
【００９３】
復号処理プログラム１０６は、クライアントＸの公開鍵証明書CK_{px}の検査を行うことによりK_{Px}を得、受信した認証情報D_n（クライアントの秘密鍵K_Sで暗号化されている）を公開鍵K_{Px}で復号化して検査データD_{n-1}を生成する。
第１７図はクライアント側の構成を示す。このクライアントシステムは、ＯＳ２０１として、例えばＷＩＮＤＯＷＳまたはＭＡＣ　ＯＳまたはＵＮＩＸまたはＮＥＴＷＡＲＥを用いる。通信プロトコルは例えばＴＣＰ／ＩＰやＯＳＩやＮＥＴＷＡＲＥを用いる。この場合、クライアント側の通信プロトコルはサーバ側の通信プロトコルに一致させる必要がある。しかし、クライアント側のＯＳはサーバ側のＯＳに一致させる必要はない。秘密鍵ファイル２０６は、当該クライアントＸの秘密鍵K_Sを保存するファイルである。この秘密鍵K_Sは所定の暗号化手順により暗号化されていることが好ましい。
【００９４】
秘密鍵K_Sの暗号化及び復号化、さらに、秘密鍵K_Sを用いた認証情報種データD_{n-1}から認証情報D_nへの暗号化は、認証処理プログラム２０２の支援の下に暗号化処理プログラム２０７によって行われる。
認証情報生成種データファイル２０４はクライアントＸの認証情報生成のための種データを記憶する。
【００９５】
サーバ側の認証処理プログラム１０４は第１５図の右側の制御手順を実行し、クライアント側の認証処理プログラム２０２は同図左側の制御手順を実行する。第１７図の実施例システムの特徴は、秘密鍵K_Sをクライアント側システムのローカルディスク上で暗号化して保管する点に特徴がある。これは、第１２図などで示した実施形態にかかる無限ワンタイム認証方式がクライアントＸが自身の秘密鍵K_Sを厳密に保管することが前提としているために、第１７図のクライアントシステムは、その管理機能を秘密鍵K_Sの暗号化で達成するものである。
【００９６】
暗号化処理プロセス２０７は種々のものが使用可能である。例えば、第１７図システムを使用するユーザにパスワードを要求する手法も簡便であるが、ＤＥＳのような適当な共通

鍵暗号方式を用い、クライアントＸのみが知るパスフレーズを鍵としてK_Sを暗号化して保管することが好ましい。この結果、K_Sが第三者に知れることがなくなり、クライアントＸになりすますことは実質上不可能となる。また、特殊機器を必要とせず暗号ソフトウェアをインストールするだけでK_Sを秘密に保管できる、また、外部インターフェイス機器が不要である等の効果を得ることができる。

【００９７】

特に、暗号処理プログラム２０７をプラグインプログラムモジュール化することにより、操作性、拡張性、可変性は飛躍的に向上する。

クライアントの公開鍵K_pをサーバに送る形態は種々の形態が考えられる。

第１６図の例では、サーバ側は、各ログインごとにクライアントからのクライアントＸの公開鍵証明書を得ることを前提としている。即ち、たとえば、クライアントがサーバに送る認証情報と共にクライアントＸの公開鍵証明書CKp_xを送る。

【００９８】

サーバ側の認証処理プログラム１０４は、クライアントＸのログインメッセージを受信すると、クライアントに認証情報要求メッセージを返し、このメッセージに対してクライアントＸから送信される認証情報を受信すると、認証情報と共に送信されてきた利用者Ｘの公開鍵証明書に付されている証明局ＣＡのディジタル署名を、証明局ＣＡの公開鍵K_{pc}（ファイル１０７中に保存されている）を用いて検査する。検査が確認されたならば、その公開鍵証明書はクライアントＸの正当な公開鍵証明書であることが確認される。また、クライアントＸの公開鍵証明書CK_{px}をファイル１０５に保存する。プログラム１０６は、データファイル１０５にアクセスして、公開鍵証明書CK_{PX}中のクライアントＸの公開鍵K_{px}を取り出す。

【００９９】

〈変形例〉

本発明はその趣旨を逸脱しない範囲で種々変形が可能である。

第１変形例：　たとえば、１６図の例では、クライアントの公開鍵証明書はログイン毎にクライアントからサーバ側に送信されるようにしていた。本方法では、クライアントの公開鍵は秘密にしておく必要がないので、毎回ログイン毎にクライアントＸの公開鍵証明書を送る必要はない。

【０１００】

そこで、サーバ側のプログラムのログインプロセス中に、クライアントＸからのログインがあったならば、そのＸの公開鍵証明書CK_{px}がすでにファイル１０５中に保管されているか否かを検査する手順を追加することを提案する。その場合、サーバ側は、公開鍵証明書が登録されていないクライアントからのログインがあった場合には、認証情報要求メッセージをそのクライアントに送る前に、公開鍵証明書要求メッセージを送るようにしてもよい。

【０１０１】

第２変形例：　上記実施例は、ログインに使用するクライアント端末がK_Sを保管している端末に限定されるという不便さがある。そこで、秘密鍵K_Sをクライアント端末上ではなくＩＣカード上に保管し、クライアントＸが常時そのカードを持ち歩くことを提案する。

そのためのクライアント側のシステムの構成を第１８図に示す。第１８図のシステムは、ＩＣカード３００に、ユーザのパスワードを記憶するパスワードファイル３０１と、公開鍵証明書を記憶するファイル３０２と、秘密鍵K_Sを記憶するファイル３０４と、特に暗号処理プログラム３０４とを有する点にある。

【０１０２】

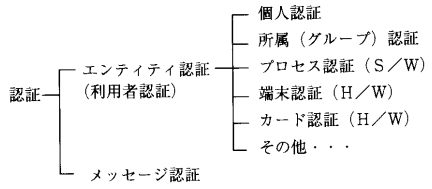
第１８図に示したシステムをクライアント側構成とした場合には、サーバ側構成は第１６図構成を援用することができる。

第１９図は、第１８図のクライアント側の認証処理プログラム３０８（クライアントホスト側）と暗号処理プログラム３０３（クライアントカード側）の連携動作を説明する。

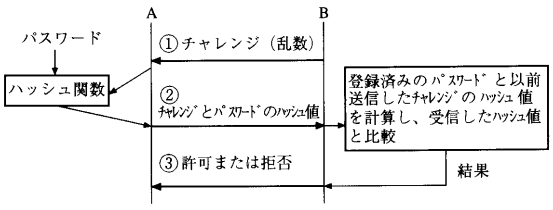
【０１０３】
先ず、ユーザによるログイン（例えば、ＩＣカードを不図示のカードリーダに読み込ませる）があると、暗号処理プログラム３０３は、端末の認証処理プログラム３０８を介してクライアントに対して認証情報の要求メッセージ（パスワードの要求メッセージ）を送る。ユーザが正規のユーザであるならば、正しいパスワードを、端末の不図示のキーボードなどから入力するであろう。そのパスワードが入力されると、プログラム３０８はその入力されたパスワードをインタフェースを介して暗号処理プログラム３０３に送る。暗号処理プログラム３０３は、受け取ったパスワードを、ファイル３０７中に記憶しておいたパスワードと比較する。
【０１０４】
一致しなければ、その旨のメッセージを認証処理プログラム３０８に返すので、認証処理プログラム３０８は当該ログインを拒絶する。
一致が得られれば、カード側の暗号処理プログラムは、クライアントに対してＩＣカードの利用許可を発行するとともに、認証サーバに対して認証要求を行うことが許可されたことを通知する。
【０１０５】
次に、クライアントは認証サーバに対する認証要求を行う。以後の手順は第１３図に説明した通りである。この場合、クライアント側においては、種データD_{n-1}から認証情報を生成するための秘密鍵K_sによる暗号化は全てＩＣカード３００内の暗号処理プログラム３０３によって行われることが重要である。即ち、秘密鍵K_sについてのいかなる情報もホスト側に伝わることはなく、伝わるのは認証情報D_nである。前述したように、認証情報D_nは第３者によって見られてもそれを解読することはできないからである。
【０１０６】
尚、第１８図のクライアント側システムでは、秘密鍵ファイル３０４が認証処理プログラム３０８に対してオープン（漏えい若しくは改ざんの虞）になることは好ましくない。クライアントのホストシステムは不特定の多数のユーザが使用する可能性があり、秘密鍵K_sが生の形でホストシステムに曝されるのは好ましくないからである。そこで、ファイル３０４中の秘密鍵K_sをパスワードファイル３０７中のパスワードによってＤＥＳなどの暗号アルゴリズムに従って暗号化することが好ましい。秘密鍵K_sを暗号化しておけば、ホスト中のたとえば認証処理プログラムが改ざんされて、ファイル３０４から秘密鍵K_sを読み出されても、ＤＥＳにより暗号化されているので、それが解読される可能性は極めて少ない。
【０１０７】
この変形例によれば、K_SがＩＣカードに保存されるので、第三者がクライアント端末を使用してクライアントＸ本人になりすますことは不可能である。換言すれば、クライアント側のシステムは汎用パソコンであっても良く、このパソコンを、クライアントＸ本人以外の者が使用することが可能となる。また、ＩＣカードとインターフェース可能な端末であれば、どのような端末でもクライアント側本体装置として使用可能となる。従って、例えば携帯端末で社外からリモート・ログイン等も可能となる。更に、ログイン処理実行に先立ち、ＩＣカードがクライアントＸ（利用者）を暗証番号等で認証する方式としているので、ＩＣカードを紛失しても第三取得者がクライアントＸになりすますことは困難である。
【０１０８】
第３変形例：　尚、上記実施形態及び実施例さらには変形例では、クライアントの公開鍵はサーバ自身が前もって保存している、或いはサーバがクライアントから取り寄せるという形態を前提としていたが、前述したように、一旦クライアントから送られてきた彼の公開鍵をサーバ側で保管しておき、以後のログインにおいてその保管しておいた公開鍵を証明書とともに流用してもよい。公開鍵は他人に知られても構わないからである。但し、証明書は（従って公開鍵も）有効期間が設定されているから、有効期間経過後のログインに対しては、前述した手法により、公開鍵証明書を再送してもらうことが好ましい。

【０１０９】
第４変形例：　また、上記実施形態及び実施例さらには変形例では、ネットワークの存在を前提としていたが、本発明はネットワークを要件としない。およそ、認証が必要であれば、例えば、ホストと入出力装置との間でも本発明を適用できる。
第５変形例：　上記実施形態では、通信回線（有線であろうが無線であろうが）を介したデータのやり取り時における認証の問題を扱ったが、本発明は、たとえば、カードを利用したドアの開閉装置に適用することも可能である。即ち、この場合は、ロック気候が認証サーバとして振る舞う。
【０１１０】
【発明の効果】
以上説明したように、本発明によれば、簡単な手順による認証方法を用いて、たとえ認証情報などが第三者に盗まれても、盗まれた認証情報などの第三者による再利用を困難にすることができる。例えば、検査情報および種データは毎回変更されるので、リピート攻撃に強く、また、たとえ送信中の認証情報が盗まれても、それを第三者がそのまま再利用する時間はほとんどないので、セキュリティは保たれる。さらに、たとえ種情報、認証情報または検査データが盗まれたとしても、クライアントの秘密鍵が管理されている限り、その盗まれた認証情報を第三者が再利用することは極めて困難である。
【図面の簡単な説明】
【図１】認証の分類を説明する図。
【図２】従来のチャレンジ・レスポンス方式の概要を説明する図。
【図３】公開鍵暗号方式の一般モデルを説明する図。
【図４】従来の認証子照合法を説明する図。
【図５】従来のディジタル証明付き認証トークン方式を説明する図。
【図６】従来のＳＳＨ方式を説明する図。
【図７】従来のＲＰＣ認証の概要を説明する図。
【図８】Ｋｅｒｂｅｒｏｓ認証方式の概要を説明する図。
【図９】零知識対話証明方式の概要を説明する図。
【図１０】従来の各種セキュリティ方式の短所をまとめた図。
【図１１】本発明の実施形態にかかる認証システムの構成を原理的に示す図。
【図１２】本発明の実施形態にかかる認証システムの構成を原理的に示す図。
【図１３】本発明の実施形態による認証手順の動作結果例を説明するフローチャート。
【図１４】本発明の実施形態による認証手順を説明するフローチャート。
【図１５】本発明の実施形態にかかる認証サーバに記憶される認証情報ファイルの構成を説明する図。
【図１６】本発明の実施例のサーバ側のシステム構成を示す図。
【図１７】本発明の実施例のクライアント側のシステム構成を示す図。
【図１８】変形例にかかるクライアント側のシステム構成を示す図。
【図１９】変形例にかかるクライアント側の処理手順を説明するフローチャート。

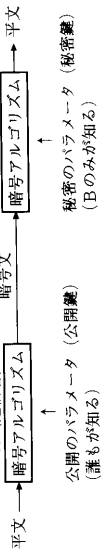
【図１】


認証
エンティティ認証
（利用者認証）
メッセージ認証
個人認証
所属（グループ）認証
プロセス認証（S／W）
端末認証（H／W）
カード認証（H／W）
その他・・・

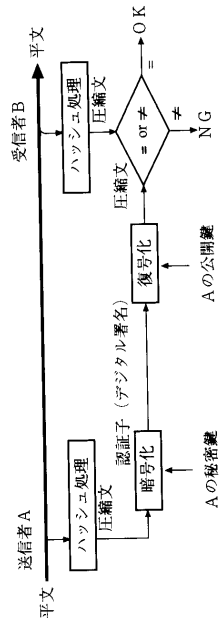
【図２】


A
B
パスワード
①チャレンジ（乱数）
ハッシュ関数
②
チャレンジとパスワードのハッシュ値
登録済みのパスワードと以前送信したチャレンジのハッシュ値を計算し、受信したハッシュ値と比較
③許可または拒否
結果

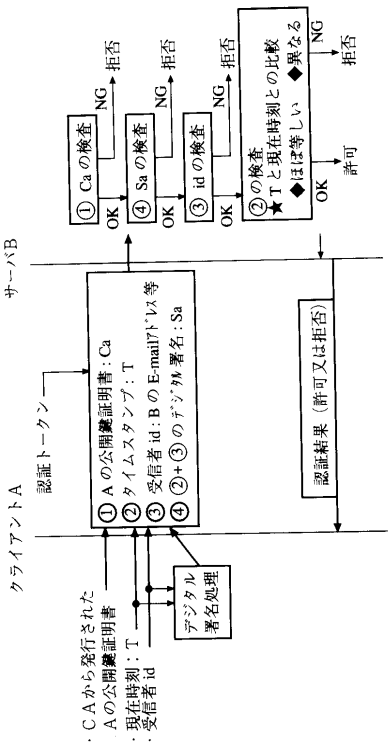
【図３】


平文
B（受信者）
暗号アルゴリズム
秘密のパラメータ（秘密鍵）
（Bのみが知る）
暗号文
A（送信者）
暗号アルゴリズム
公開のパラメータ（公開鍵）
（誰もが知る）
平文

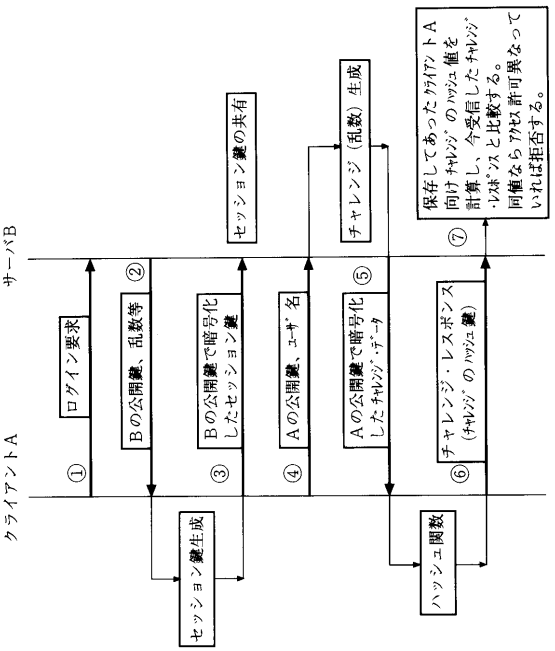
【図４】


受信者B
平文
ハッシュ処理
圧縮文
＝ or ≠
＝
OK
≠
NG
圧縮文
復号化
Aの公開鍵
認証子（デジタル署名）
送信者A
ハッシュ処理
圧縮文
暗号化
Aの秘密鍵
平文

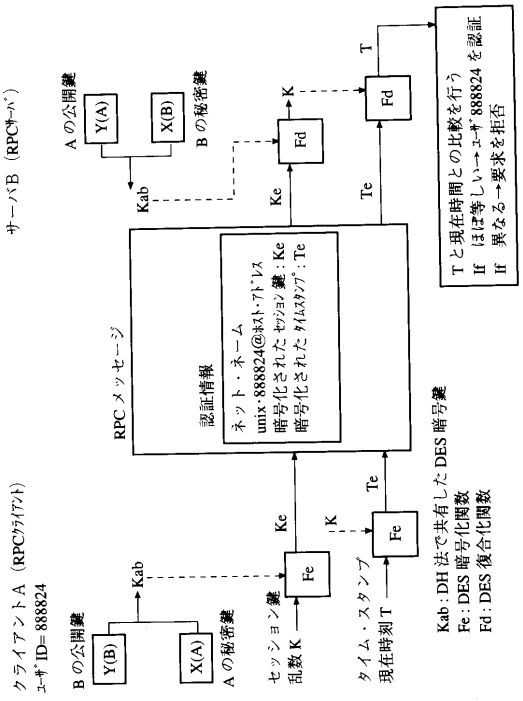
【図５】


サーバB
クライアントA
認証トークン
①Aの公開鍵証明書：Ca
②タイムスタンプ：T
③受信者id：BのE-mailｱﾄﾞﾚｽ等
④②+③のﾃﾞｼﾞﾀﾙ署名：Sa
・CAから発行された
Aの公開鍵証明書
・現在時刻：T
・受信者id
デジタル
署名処理
①Caの検査
NG
拒否
OK
④Saの検査
NG
拒否
OK
③idの検査
NG
拒否
OK
②の検査
★Tと現在時刻との比較
◆ほぼ等しい
◆異なる
NG
拒否
OK
許可
認証結果（許可又は拒否）

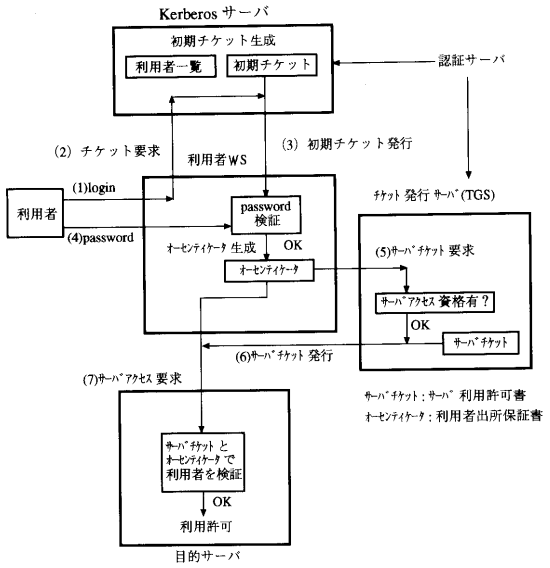
【図６】


サーバB
クライアントA
ログイン要求
Bの公開鍵、乱数等
Bの公開鍵で暗号化したセッション鍵
Aの公開鍵、ユーザ名
Aの公開鍵で暗号化したチャレンジ・データ
チャレンジ・レスポンス（チャレンジのハッシュ鍵）
セッション鍵生成
セッション鍵の共有
チャレンジ（乱数）生成
ハッシュ関数
保存してあったクライアントA向けチャレンジのハッシュ値を計算し、今受信したチャレンジ・レスポンスと比較する。
同値ならアクセス許可異なっていれば拒否する。

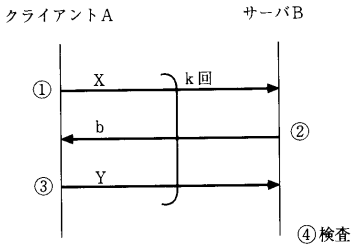
【図７】


サーバB（RPCサーバ）
Aの公開鍵
Y(A)
X(B)
Bの秘密鍵
Kab
Ke
K
Fd
T
RPCメッセージ
認証情報
ネット・ネーム
unix-888824@ホスト・アドレス
暗号化されたセッション鍵：Ke
暗号化されたタイムスタンプ：Te
Tと現在時間との比較を行う
If ほぼ等しい→ユーザ888824を認証
If 異なる→要求を拒否
クライアントA（RPCクライアント）
ユーザID=888824
Bの公開鍵
Y(B)
Kab
Aの秘密鍵
X(A)
セッション鍵
乱数K
Fe
K
Te
タイム・スタンプ
現在時刻T
Kab：DH法で共有したDES暗号鍵
Fe：DES暗号化関数
Fd：DES復合化関数

【図８】


Kerberosサーバ
初期チケット生成
利用者一覧
初期チケット
認証サーバ
(2) チケット要求
利用者WS
(3) 初期チケット発行
(1)login
利用者
(4)password
password検証
オーセンティケータ生成
OK
オーセンティケータ
チケット発行サーバ(TGS)
(5)サーバチケット要求
サーバアクセス資格有？
OK
サーバチケット
(6)サーバチケット発行
(7)サーバアクセス要求
サーバチケットとオーセンティケータで利用者を検証
OK
利用許可
目的サーバ
サーバチケット：サーバ利用許可書
オーセンティケータ：利用者出所保証書

【図９】

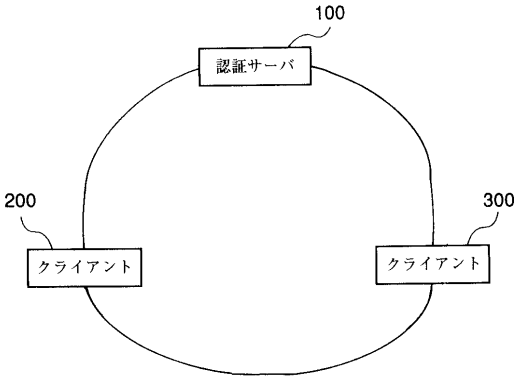

クライアントA
サーバB
①
X
k回
b
②
③
Y
④検査

【図１０】

凡例 ○：なりすまし可
△：一部なりすまし可
×：なりすまし不可

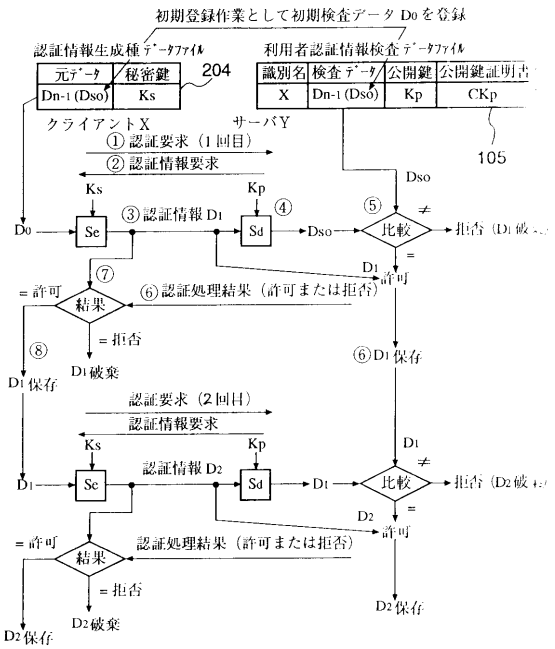
方式	認証方式 ＼ 脅威	「なりすまし」による不正行為者 外部の第三者	「なりすまし」による不正行為者 サーバ管理者	備考
①知識利用	従来のパスワード式	○	○	簡単に盗聴、再利用可能
	ワンタイム・パスワード方式	×	△	盗聴内容を再利用できないが、ﾊﾟｽﾜｰﾄﾞを使い切ったら再登録処理要
	チャレンジ・レスポンス方式	×	○	ｻｰﾊﾞに侵入されたら辞書攻撃にさらされる
②暗号利用	デジタル署名方式	△	○	盗聴した認証情報の再利用が可能
	デジタル署名付認証トークン方式	△	×	一定時間内であれば盗聴した認証情報の再利用が可能
	SSH(Secure SHell) 方式	△	×	同上
	RPC 認定方式	△	○	同上
	Kerberos 方式	△	○	同上、また集中管理する認証ｻｰﾊﾞが破られたらそのﾄﾞﾒｲﾝが全滅する
	ゼロ知識対話証明方式	×	×	対話ｼｰｹﾝｽが冗長 認証ﾌﾟﾛｾｽが複雑になる
③生体計測	指紋、声紋、手形、網膜ﾊﾟﾀｰﾝ、署名、筆記ﾊﾟﾀｰﾝ、ｷｰｽﾄﾛｰｸ	△	○	ｵﾌﾗｲﾝ(ﾛｰｶﾙ)認証では安全性（本人認証精度）が高いがﾈｯﾄﾜｰｸを介した認証処理の場合は、盗聴による再利用が可能で①または②と同様の安全性しかない
④所有物利用	鍵、トークン、バッチ、電子キー、磁気カード、ICカード、非接触型カード	△	○	同上

【図１１】



【図１２】

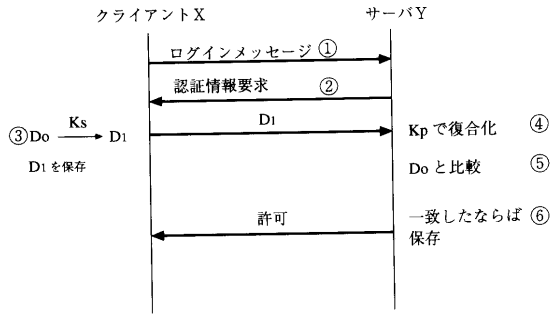
【図１３】



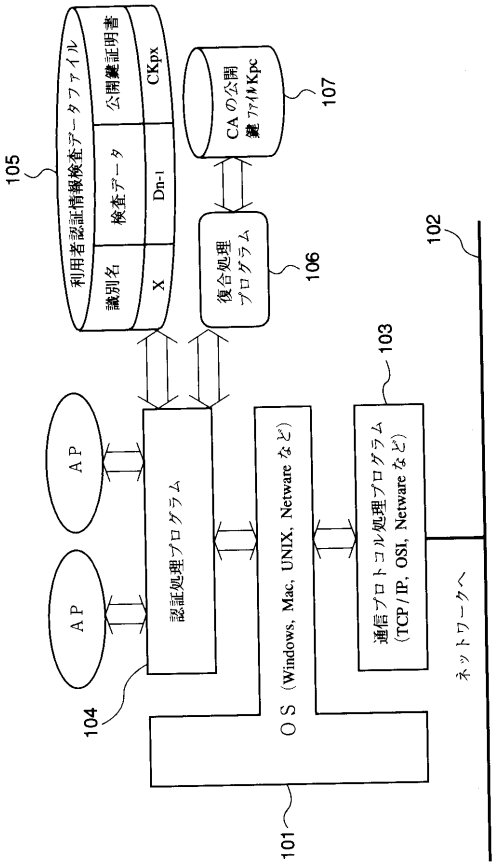
【図１４】

識別名	検査データ	公開鍵
X	Dx = Do	Kpx
W	Dw = Do	Kpy
：	：	：

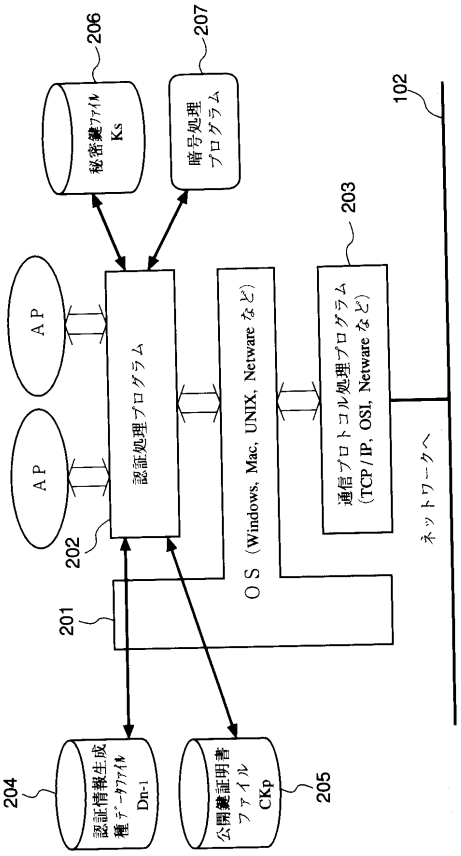
【図１５】



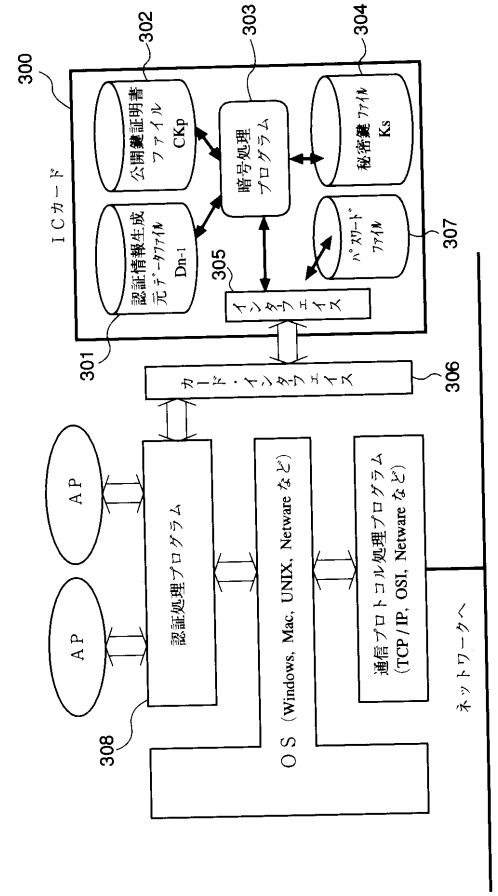
【図１６】



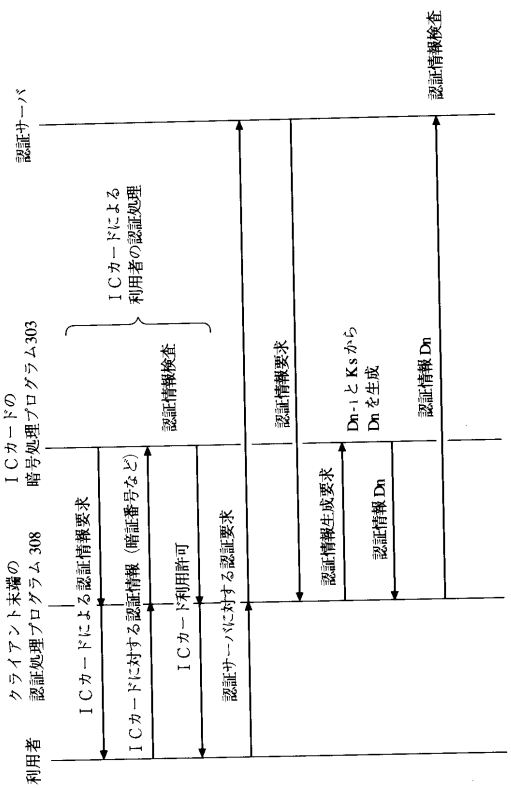
【図１７】



【図１８】



【図１９】



フロントページの続き

(56)参考文献 特開平４－３０６７６０（ＪＰ，Ａ）
特開平５－２１９０５３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
H04L 9/00