平成 14 年度

修士学位論文

信頼の輪モデルに基づいた システム利用権限の委譲による個人認証手法

A Personal Authentication Method using Authority Delegation Based on the Web of Trust Model

1055117　　正岡　元

指導教員　　菊池　豊 助教授

平成 15 年　2 月　24 日

高知工科大学大学院 工学研究科 基盤工学専攻

情報システム工学コース

要 旨

信頼の輪モデルに基づいた
システム利用権限の委譲による個人認証手法

正岡　元

信頼の輪は、個人と個人との信頼関係が複数存在する際に、それらの合成によって構築されるモデルである。本研究では、この信頼の輪をもとにしてシステムにおける利用権限をユーザに委譲することにより、個人認証を行う手法を提案する。

システムにおけるユーザの権限は、そのシステムの管理者が委譲するのが一般的である。NIS などの管理システムを導入した場合でも、NIS 管理者の権限によって、ユーザへの権限委譲が行われる。これらの管理手法では、管理コストが管理者に集中してしまう。

本提案では、従来の管理手法に信頼の輪モデルを導入することで、既に権限をもっているユーザの信頼に基づいて、一部あるいは全部の権限をユーザに与える手法を実現する。この場合管理者がユーザに直接権限を委譲する必要は無く、これにより従来管理者に集中していた管理コストを分散でき、管理者の負担を軽減することが可能となる。

キーワード　信頼の輪, 個人認証, ユーザ管理, 利用権限, 委譲

Abstract

A Personal Authentication Method using Authority Delegation Based on the Web of Trust Model

MASAOKA Hajime

The web of trust is the model built by that composition when trust relations with an individual and an individual exist in the plual. This research proposes a personal authentication method using authority delegation based on the web of trust.

It is general that the administrator of that system gives authority to user. Therefore an administration cost is concentrated on the administrator.

This proposal introduce the web of trust model to an conventional administration method. The method which gives a user a part of the authority is realized based on trust of the user who already has authority.

key words Web of Trust, authentication, user management, authority, delegate

目次

目次

図目次

表目次

第 1 章

はじめに

信頼の輪は、個人と個人との信頼関係が複数存在する際に、それらの合成によって構築される個人認証のモデルである。本研究では、この信頼の輪モデルを基にしてシステムにおける利用権限をユーザに委譲することによって個人認証を行う手法を提案する。

従来のユーザ管理手法では、システムの利用権限はそのシステムの管理者がユーザに対して委譲することが一般的である。そのため、管理コストが管理者に集中してしまう。この管理コストは、ユーザの数やシステムの数にあわせて増加する。さらに従来の手法では、個々のユーザに対して与える権限の制限を、細かく行うことは困難であった。

本研究では、従来の管理手法に信頼の輪モデルを導入することで、既にシステムの利用権限を持っているユーザの信頼に基づいて、一部、あるいは全部の権限をあらたなユーザに与える手法を実現する。この場合管理者はユーザに直接権限を委譲する必要は無い。そのため、管理コストを分散させることが可能であり、管理者の負担を軽減することが可能となる。

まず、信頼について述べ、複数の信頼関係を結合することで複雑な信頼の構造を構成できることを述べる。そして信頼の輪モデルがどのように構築されているかについて述べる。次に従来のユーザ管理手法の問題点を示し、従来の手法に信頼の輪モデルを導入することによって問題点を解決する手法について述べる。そして、この手法を実現するための実装について述べた後、この実装を評価する。評価は、実際にこの手法によって管理コストが軽減されるか、導入のためのコストや、日常の管理業務における維持コストは増加しないか、目的通りに柔軟な権限の委譲が行えるか、などについて行う。

第 2 章

信頼の構造

本研究では、信頼の輪モデルを用いて個人の認証を行う。この章では、この信頼の輪モデルについて述べる。まず個人と個人との間で信頼関係が構築されることを述べ、それが複数の人間がいる場合に信頼関係がどう拡張されるか示す。次に複数の人間があつまってできる信頼関係の 1 つである信頼の輪モデルの概念について述べ、その具体的な実装として、PGP における信頼の輪モデルについて述べる。さらに、信頼の輪とは別の方法で認証基盤を構成する PKI を示す。また本研究の求める信頼の輪に対して、これらのモデルの持つ特徴と不足する点を述べる。

様々な文献における信頼の構造に関する記述では、「信頼」という言葉と「信用」という言葉とがほぼ同義に用いられている。本論文では、すべて信頼に統一する。

2.1 信頼関係の構築

本節では、個人と個人との間で信頼関係ができ、それが複数結合されることによって複雑な構造を形作るまでの流れを述べる。さらに、間接的な信頼によって、直接知らない人物を信頼することが可能であることを述べる。

2.1.1 ユーザ認証における信頼

システムにおける利用権限は、そのシステムの管理者から与えられる。これはつまり、管理者がユーザを信頼している状態である。管理者はユーザを信頼した責任を負うため、ユーザが (故意であれ過失であれ) システムに対して被害を与える人物であるかどうかを判断す

ることになる。ユーザの側では、自分を信頼している管理者を裏切らないように行動することになる。

この関係は、小さなコミュニティでは正常に機能する。しかし、ある程度大きくなったコミュニティでは、様々な問題を引き起こす。大きくなったコミュニティでは、管理者がユーザを直接知らないか、知っていても信頼するに足るかどうかを判断するだけの情報を持っていない可能性がある。その場合、管理者がとる行動の例は以下である。

- 信頼できるかどうか判断できないため、権限を与えない
- 信頼できるかどうか判断できないが、権限を与える
- 判断するための情報を集めた上で判断する

ユーザに利用権限を与えることは必要な事項であることが多く、1 つ目の選択肢を選択することは現実的には困難である。2 つ目の選択肢は、管理者が責任を負う事の負担が大きい。しかし、現実ではこの選択肢が選ばれる事が多々あると考えられる。3 つ目の選択肢は安全性では問題なさそうに見える。しかし、管理者に本来必要でないはずの負担を強いる点では 2 番目と同様であり、いずれの選択肢もそれぞれ管理者に負担をかけるという問題点を含む。

これを解決する方法として、信頼の判断を他人にゆだねるという方法が考えられる。この方法について次節以降に述べる。

2.1.2　信頼の結合

第 2.1.1 節で述べたように、場合によっては直接知らない相手が信頼できるかどうか判断する必要がある。それには、判断を他人に委ねるという方法が利用できることを述べた。この方法について詳しく説明する。

まず、個人同士が信頼している状態を図 2.1 に示す。

この場合 A と B とは互いに信頼しあっており、A は B に対して、B は A に対して責任を負っている状態と呼べる。このような信頼関係が複数存在する状態を図 2.2 に示す。

ここで B と C との間は相互信頼ではなく、B が C を一方的に信頼している状態である。

2.1　信頼関係の構築

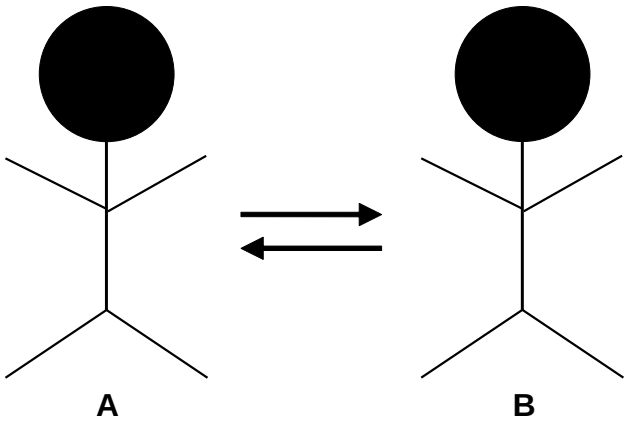

図 2.1　個人と個人との信頼関係

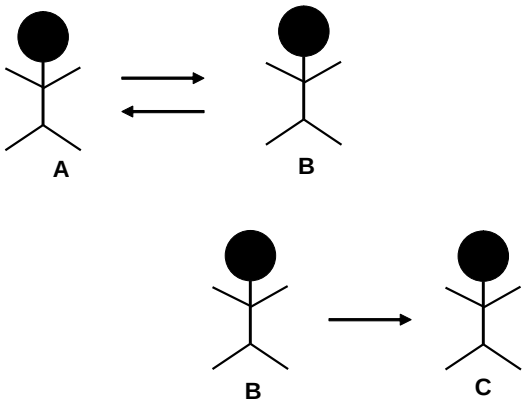

図 2.2　複数の信頼関係

このように人間が複数いてそれぞれの間に信頼関係がある場合、それらを概念的に結合することができる。この場合、両方の関係に B が関わっているので、B を中心に結合すると、図 2.3 のようになる。

この図には、A, B, C の 3 人の信頼関係が示されている。A と B とは互いに信頼しあっており、B は C を一方的に信頼している。

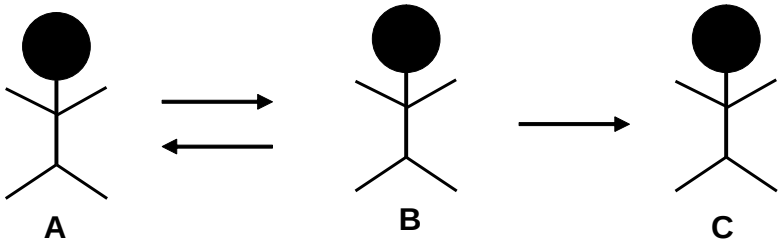

図 2.3　複数の信頼関係の結合

2.1.3　間接的な信頼

次に、第 2.1.2 節で述べたような複数の人物の間に信頼関係が存在している時、直接知らない人物を信頼することができることを示す。

管理者は、人物 A を信頼していると仮定する。その人物 A は、人物 B を信頼している。この場合管理者は、人物 A を通して間接的に人物 B を信頼することが可能である。現実社会ではこの概念は一般に受け入れられており、「彼が信頼できると言うのであればその通りだろう」という判断が日常的に行われている。

これを第 2.1.2 節の図 2.3 に当てはめる。A は C を直接知らないので、信頼することはできない。ここで、A が B を信頼していることに注目する。また、B は C を信頼している。前述の通り、A は、自らの信頼する B が信頼している C を、信頼することができる。ただし、この場合の A の C に対する信頼は A の B に対する信頼より低い。一方、C は B を信頼していないので、C は A を信頼することはできない。よって、A と C との信頼関係は A から C への一方的なものとなる。この状態を図 2.4 に示す。

この間接的な信頼の概念を、管理者とユーザの間に適用することで、管理者が直接知らないユーザに対して権限を与えるための判断を容易にすることができる。この場合責任の所在は直接的な信頼の関係に依存していると考えられる。つまり、前述の例であれば、人物 B に対する責任は管理者ではなく人物 A にあるといえる。

これまで、個人と個人との間の信頼関係が複数あるとき、それを結合できること、さらに

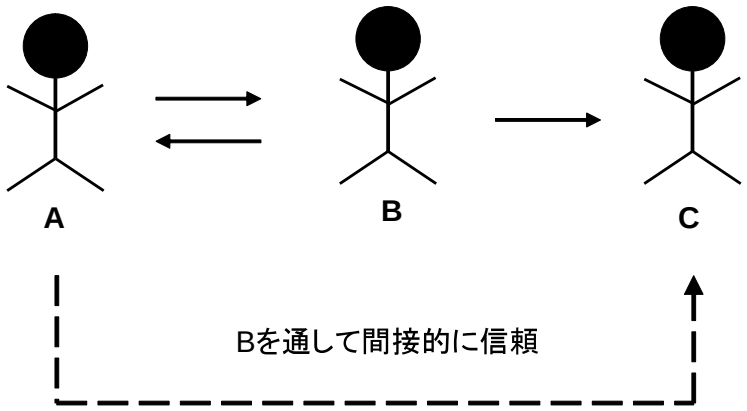

図 2.4　間接的な信頼関係

その時、間に人をはさんで間接的な信頼関係を構築することができることを述べた。この間接的な信頼関係をさらに拡張し、より複雑な信頼関係を構築するモデルについて次節以降に述べる。

2.2　信頼の構造

間接的な信頼を拡張した構造には、大きく分けて以下の 2 種類がある。

- 相互的な信頼関係が存在する構造
- 1 人を頂点とした一方的な信頼の流れのみがある構造

前者における個人の信頼関係は前順序関係であり、互いに信頼しあう関係が存在する。後者のそれは半順序関係であり、個人間の信頼関係は一方的なもののみとなる。ここでは、前者を「信頼の網構造」、後者を「信頼の木構造」と呼んで区別する。

2.2.1　信頼の網構造

信頼の輪とは、個人が個人を信頼する関係が複数存在するとき、それらを結合してできるモデルである。まず、個人が個人を信頼する関係 (図 2.1) が始めにある。複数の個人がいる場合、その関係は複数存在しうる (図 2.2)。さらに、それらを結合することで、間接的な信頼関係 (2.4) が構築できる。

それらの信頼関係の合成により、複雑な、信頼関係の網構造が構成される。信頼の網構造を図 2.5 に示す。この信頼の網構造のことを信頼の輪と呼ぶ。

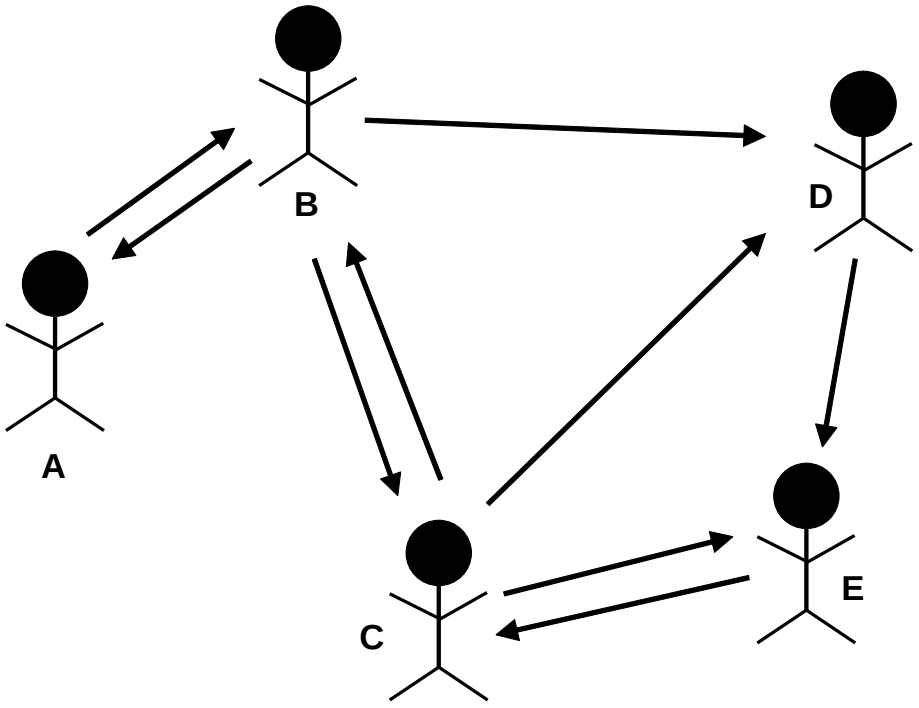

図 2.5　信頼の網構造

信頼の輪では相手を互いに信頼しあう関係が存在する。また、ある 2 人の間にある信頼関係は、間接的な信頼関係も含めて複数存在する可能性がある。図 2.5 の例では、A と B、B と C、C と E の間は互いに信頼しあう関係になっている。また、例えば B と D との間の信頼関係は、B と D との直接的な関係と、C をはさんだ間接的な関係とがある。

この信頼の輪では、個人間の信頼を結合して行くことで、知らない人物を間接的に信頼す

ることができるようになる。個人間の信頼さえ存在すれば、間接的な信頼関係を構築することができるため、この構造を構築することは容易である。

2.2.2　PGP の場合

第 2.2.1 節に述べた信頼の輪モデルでもっとも著名なものが OpenPGP[3][*1]である。OpenPGP （以降、PGP と表記）の実装としては PGPi[*2]と GPG[*3]とがよく知られ、広く利用されている。

PGP は電子メールを暗号化し、または電子署名するためのソフトウェアとして知られている。この電子署名における個人認証に信頼の輪モデルが利用されている。

PGP における電子署名は公開鍵暗号方式における署名によって実現されている。PGP では、送信する電子メールが、正しく本人から送信されたことを示すために電子署名を行う。署名の検証が成功すれば、その電子メールは本人が送信したものであることを示す。メールに署名して送信し、それが検証されるまでの流れを図 2.6 に示す。

まず A はメールを書き (1) 自分の秘密鍵を用いて署名する。(2) 次に、署名したメールを C に送信する。(3)C は、受け取ったメールを、A の公開鍵を用いて検証する。成功すれば、このメールは A から送られたものであると判断できる [8]。失敗した場合は、以下のいずれかの理由が考えられる。

- A が署名に失敗した
- 誰かが A になりすまして送信した
- 誰かが途中でメールの内容を改竄した

署名の検証をするためには、その人物の正しい公開鍵を持っている必要がある。公開鍵は、鍵管理サーバに登録されており、必要な公開鍵を検索して取得することができる（図

[*1] http://www.openpgp.org/
[*2] http://www.pgpi.org/
[*3] http://www.gnupg.org/

2.7)。

他に、Web ページにおいて公開したり、電子メールに添付して渡したり、印刷物での受け渡しも可能である。しかし、手に入れた公開鍵が本人のものであるかどうかはわからない。それを確認するために、鍵の指紋（フィンガープリント）の確認を行う。公開鍵にはそれぞれ異なるフィンガープリントがある。このフィンガープリントと、本人から直接確認したフィンガープリントとを照合することで、公開鍵が本人のものであることを確認することが可能である。こうして得た公開鍵を用いて、送られてきた電子メールの署名を検証することで、その電子メールが本人から送られたものかどうか確認することが可能である。

当然のことながら、フィンガープリントの確認は信頼できる手段で行う必要がある。フィンガープリントを暗号化されないメールで受け取ったとしても、そのフィンガープリントが本人によって送られているという保証はない。理想的には、フィンガープリントの確認は直接会って行うことが望ましい。それが不可能な場合は、複数の手段を利用してフィンガープリントを取得し、フィンガープリントそのものが改竄されてしまう危険性を減らす必要が

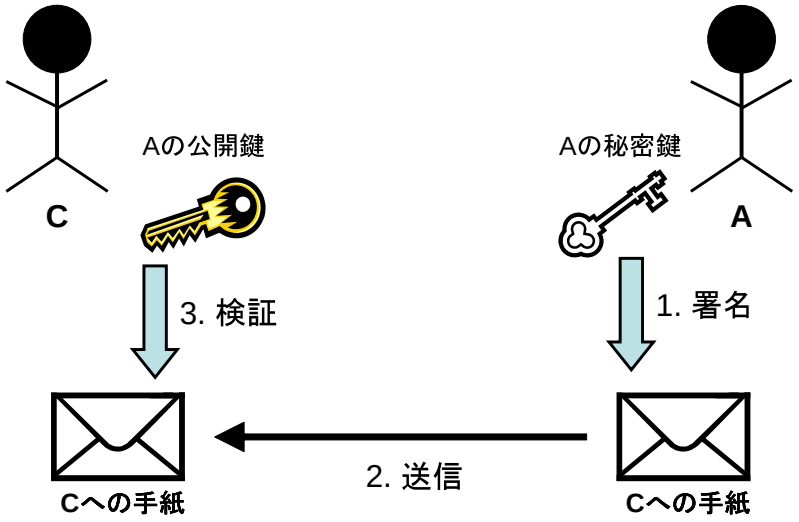

図 2.6　署名と検証

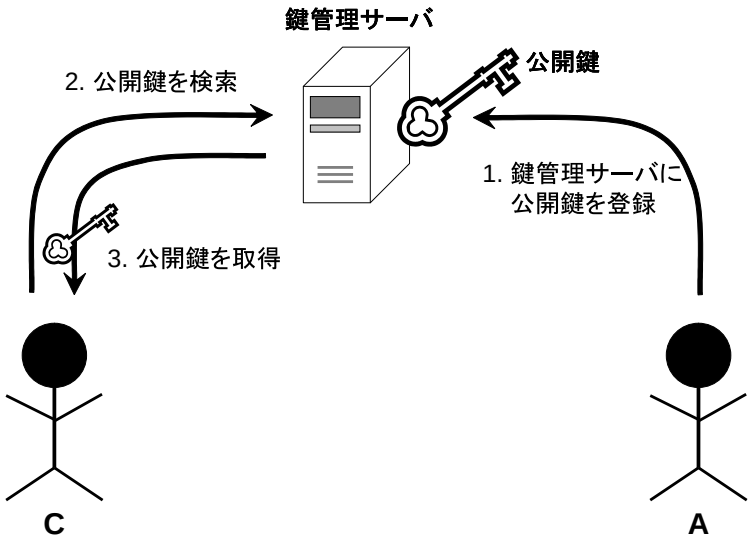

図 2.7　鍵管理サーバ

ある。

フィンガープリントの確認を行えば、高い精度で公開鍵が本人のものであるかどうかを知ることができる。しかし、この方法には以下のような欠点もある。

- 公開鍵の持ち主が直接知らない人であった場合、フィンガープリントの確認を行うのは困難である場合がある
- 公開鍵の持ち主を照合するためにフィンガープリントの確認を必要とすれば、データの送信者と受信者とが同時に通信を行う必要が無いという電子メールの利点が損なわれてしまう

前者では、公開鍵の持ち主について直接知らない場合は、メール以外の連絡先がわからずにそもそもフィンガープリントの確認ができない状況があり得る。この場合、フィンガープリントが正しいことを確認することはできず、公開鍵が本人のものであるかどうか知ることはできない。メール以外の連絡方法があったとしても、相手を直接知らない場合、連絡をとった相手が本人であると判断することができない。

この電子署名を、電子メール本文ではなく、公開鍵に対して行うことができる。これは、その公開鍵は正しい本人の持ち物である、ということを自分が証明する行為である。そのため、公開鍵に署名する際は十分注意して、本人のものでない公開鍵や、確実に本人のものであると確認されていない公開鍵にみだりに署名することは避ける必要がある。

PGP に代表される信頼の輪モデルには、第 3 者機関を必要とせずに信頼の構造を構築できるという利点がある。これは同時に、費用が発生しないという利点にもなる。しかし、個人同士の信頼関係によってしか任意の人物を信頼できるかどうか判断することができないため、やり取りする相手との間に介在する人物が増える、つまりやり取りする相手へのパス長が長くなると、相手を信頼できるかどうかの判断が難しくなってしまうという欠点がある。

2.2.3　信頼の木構造

複数の信頼関係を結合してできる信頼の構造のうち、網構造をとらないものとして信頼の木構造がある。信頼の木構造では、1 人を頂点として末端に至るまで一方的な信頼関係が構築される。信頼の木構造の例を図 2.8 に示す。

信頼の木構造の特徴は、頂点を始点として任意のノードに向かう信頼関係が存在するということである。つまり頂点のノードは、木構造に含まれる任意のノードを信頼していると言える。そのため、頂点のノードを信頼することができれば、木構造に含まれる任意のノードを信頼することができる。

2.2.4　PKI の場合

信頼の木構造の 1 つである PKI について述べる。ここでの PKI は RFC 3280[6] に規格化されているものを指す。RFC 3280 の PKI は X.500 シリーズのディレクトリに基づいている。X.500 シリーズは ITU-T[*4]によって定められた規格であり、認証基盤となるためのディレクトリモデルを定義する。X.500 シリーズをなす主要な規格に X.500[2] と X.509[1]

[*4] http://www.itu.int

とがある。X.500 はディレクトリの概念、モデル、サービスの概要を定義しており、X.509 はディレクトリによる認証フレームワークを定義している [7]。

PKI では、第 2.2.3 節に述べた木構造の特徴を利用して、大規模な認証基盤を構築する。PKI における木構造において、末端ノード以外は第 3 者による認証機関となる。この認証機関のことを認証局 (CA) と呼び、PKI における重要な要素の 1 つである。ディレクトリの頂点にある CA を特にルート (root)CA と呼ぶ。CA の主要な役割は、証明書の作成と個人の認証である。図 2.9 に CA の階層モデルを示す。閉じたコミュニティや、比較的小さなコミュニティである場合は、CA は第 3 者に委託せず、自身で用意することもある。コミュニティ全体を見渡せる場合は、これで十分な信頼性を得ることができる。しかし、コミュニティがある程度の規模になったり、別のコミュニティとのやり取りが必要である場合は、信頼できる第 3 者に対して CA を委託することになり、費用が発生する場合がある。

この図における矢印は信頼関係である。矢印は上から下へ向かうのみなので、上位のノー

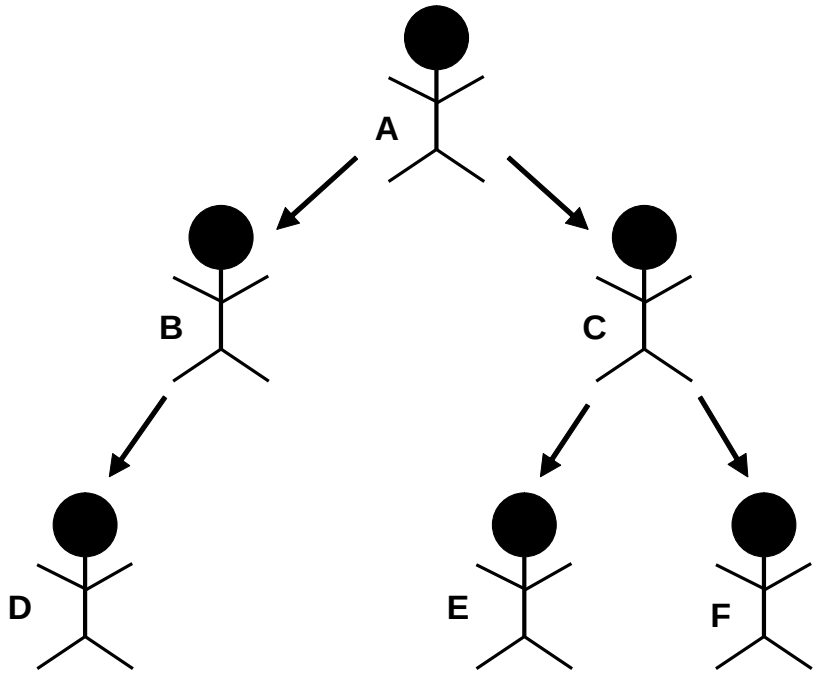

図 2.8　信頼の木構造

ドが、下位のノードを信頼している。CA1 と CA2 とは、それぞれの信頼する S1 から Sn までのノードに対して責任を負い、ルート CA は、CA1 と CA2 とに対して責任を負う。そのため、ルート CA は、階層に属する全てのノードに対して責任を負うことになる。これによって、ルート CA のみを信頼することで、木構造の全体を信頼することのできる構造が構築される。

この階層構造は、ルート CA を中心として統一された信頼の構造を持つという利点がある。また、CA1 や CA2 などのサブ CA にノードに対する信頼を分散させることでルート CA の役割を減らすことにつながり、階層構造が大きくなっても運営することが可能である。しかし階層構造には欠点もある。ルート CA が木構造全体に対する責任を負うことになるため、どの組織がルート CA になるかが問題になる。大きな責任を負うことになるルート CA の運用には様々な費用が必要であるため、PKI 全体が大きくなればなるほど、その利用に対する費用は高価にならざるを得ない。逆に、安価に利用できる PKI では、ルート CA の信頼性は低くなってしまう。しかし、ルート CA を用いて階層化することだけが、複

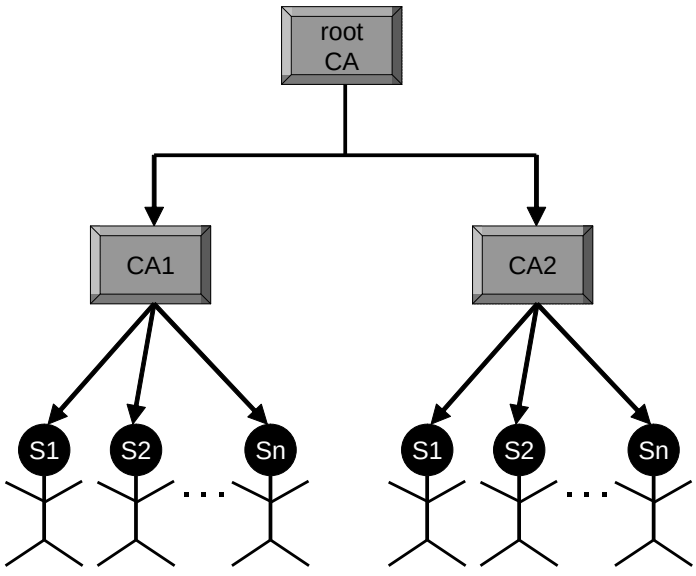

図 2.9　PKI における CA の階層構造

数の CA を連携させる方法ではない。

ルート CA を用いずに複数の CA を連携させる方法の 1 つが、ピア・ツー・ピア・モデル（P2P モデル）である。複数の CA は、互いに信頼しあい、対等の関係を築く。このため、全ての CA がルート CA であるということもできる。図 2.10 に、3 つの CA である CA1、CA2、CA3 が P2P モデルによって連携している様子を示す。

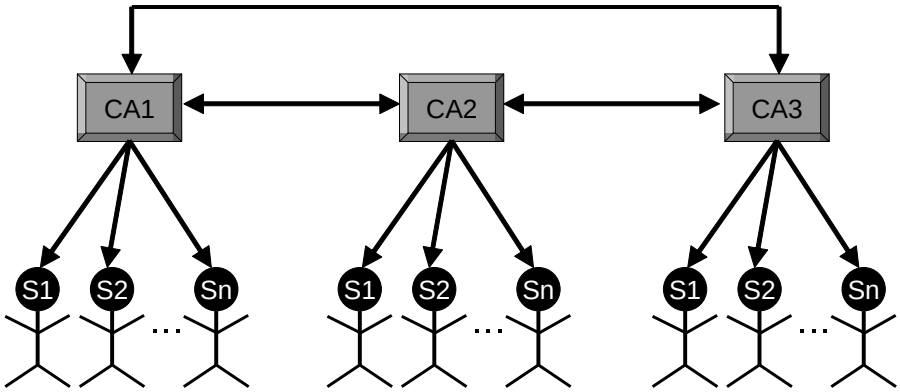

図 2.10　ピア・ツー・ピア・モデルによる CA の連携

図中の矢印は信頼関係を表す。ここで、CA1 と CA2、CA2 と CA3、CA1 と CA3 は、それぞれ互いに信頼しており、全ての CA がルート CA であるとみなすことができる。そのため、どの CA も、自らの下位のノードに対する責任のみを負えばよく、木構造全体に対して責任を負う必要は無い。よってこのモデルでは、比較的運営に必要な費用は少なくできる。しかし、大きな欠点として互いを信頼しあう関係が多くなりすぎるというものがある。このモデルでは、全ての CA が自らを除く他の全ての CA と互いに信頼関係を持つ必要がある。つまり、N 個の CA が存在する場合、必要な信頼関係は $\frac{N^2-N}{2}$ だけ必要になる。そのため CA の数が大きくなるにしたがって非常に多くの信頼関係が必要になり、現実的ではない。とはいえ、必ずしもルート CA を必要とせずに複数の CA を連携させられるという利点はあり、このモデルを基本として多くの信頼関係が必要となる欠点を改善した、ブリッジ・モデ

ルが存在する。

ブリッジモデルでは、図 2.11 に示すように、CA と CA との間にブリッジ CA を設ける。図中の矢印は信頼関係を表す。

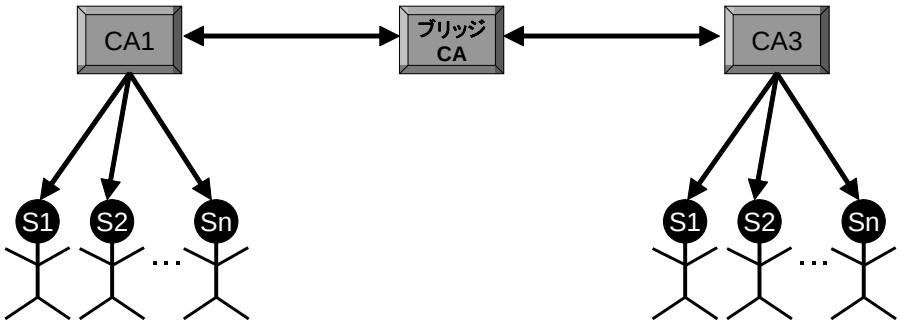

図 2.11　ブリッジモデルによる CA の連携

階層構造では 1 つの CA がルート CA の役割を果たす必要があり、また、ピア・ツー・ピア・モデルでは相互信頼関係の数が非常に多く必要となる。このブリッジモデルでは、中心にブリッジ CA を設ける。CA1 と CA3 とは、それぞれブリッジ CA と信頼関係を結ぶ。つまり、CA1 と CA3 とはブリッジ CA によって互いに信頼関係を結ぶことができることになる。これによって、ブリッジ・モデルによる CA の連携に必要な信頼関係の数は、N 個の CA に対して N となる。このブリッジ・モデルにおいても、全ての CA はルート CA であるということができる。

ピア・ツー・ピア・モデル、ブリッジ・モデルには共通する問題がある。それは、頂点にあって全体を統一するルート CA が存在しないため、各々の CA が下位ノードを信頼する基準がばらばらになってしまう可能性がある、ということである。それが CA 群全体で統一されていないと、ある CA を認証した時に全体を信頼できるとは限らず、複数の CA を連携する価値がない。そのため、これらのモデルを用いて複数の CA を連携する際には、最低限のポリシーと手続きを全ての CA において協調させておく必要がある。

これまで述べてきたような、PKI に代表される信頼の木構造モデルでは、認証は第 3 者機関によって行われる。そのため客観的な判断が可能となるという利点が存在する。しかし、認証のための第 3 者機関を用意するコストが必要であることや、頂点となる第 3 者機関を信頼できない場合には木構造全体を信頼することができないため、その場合にはこのモデルによる認証を利用することができないという欠点がある。

2.3　既存のモデルの評価

第 2.2 節に述べたように、信頼の構造には大きくわけて 2 つの形態がある。また、木構造には、3 つのモデルがある。

本研究では、信頼の構造をシステムのユーザ認証に組み入れることによって、適切な権限をユーザに与える手法を提案する。これは、ユーザが信頼する別のユーザには、もとのユーザの持つ権限の一部を自動的に与えることによって実現される。そのため、信頼の構造は以下に示す項目を可能な限り満たすことが望ましい。

要件 1　ユーザは、任意の人物を信頼することが可能であること

- ユーザは、相手を信頼することで権限を委譲する。権限を委譲したい相手を、構造上の制約により信頼できないことは望ましくない。

要件 2　システムは複数のユーザを信頼できる、つまり信頼の起点となるユーザが複数存在できること

- 提案する手法では、間接的な信頼によって権限を委譲することが可能であるが、委譲される権限は信頼の元となるユーザの持つ権限に依存する。システムは、ただ 1 人のユーザだけでなく、複数人のユーザに対して権限を与えられることが望ましい。

要件 3　小さなコミュニティにおいて採用しやすいために、必要な費用が少ないこと

- 必要な費用が高価であると、小さなコミュニティにおいて導入することは困難になってしまう。必要な費用は、可能な限り少ないことが望ましい。

2.3 既存のモデルの評価

ここで、本章で説明してきた信頼の構造について整理し、比較を行う。まず、比較する点を以下に示す。

基準 1 第 3 者機関の必要性

- 構造を形作るために、コミュニティの外部による機関が必要になるかどうか

基準 2 root CA の必要性

- 木構造において、頂点となる root CA が必要かどうか

基準 3 木構造のスケーラビリティ

- 木構造のスケーラビリティが高いか、低いか

基準 4 双方向の信頼関係の存在

- 互いに相手を信頼しあう、双方向の信頼関係が存在できるかどうか

基準 5 パス長による信頼度の変化

- 認証したい相手との間に入る人の数、つまりパス長が長くなったときに、相手に対する信頼度が変化（例えば低下）するかどうか

基準 6 必要な費用

- 該当する構造によって認証モデルを構築する際に必要な費用が高いか、安いか

これらの基準を元に、各構造を比較した結果を表 2.1 に示す。

構造の形態	基準 1	基準 2	基準 3	基準 4	基準 5	基準 6
網構造（信頼の輪モデル）	不要	―	―	可	する	○
木構造（階層モデル）	必要	必要	○	不可	しない	×
木構造（P2P・モデル）	必要	不要	×	不可	しない	△
木構造（ブリッジ・モデル）	必要	不要	○	不可	しない	△

表 2.1 信頼の構造の評価

表中の「―」は、その項目に該当しないことを示す。基準 3 では、スケーラビリティが高い形態を○、低い形態を×とした。また、基準 6 における必要な費用については、より費用

が少なくてすむ形態を○、費用が多く必要な形態を×、その中間を△とした。

要件 1 にあてはまるのは、基準 4 より網構造だけである。要件 2 には基準 2 より網構造、および木構造のうち P2P・モデルとブリッジモデルとが該当する。要件 3 にあてはまるのは、基準 6 により網構造である。

以上の比較より、3 つの要件を全て満たすことのできる構造は網構造のみである。この理由から、本研究における信頼構造として網構造である信頼の輪モデルを利用することとする。

第 3 章

認証の概念

本研究では、個人認証に対して信頼の概念を取り入れる。本章では、この認証の概念について述べる。まず認証を定義した後に、認証された利用者に与えられる利用権限について例を挙げて述べる。さらに、様々な認証について述べる。

3.1 認証とは

認証とは、今まさにシステムを利用しようとしている人物（以降、利用者と呼ぶ）利用者がシステムにおいて何らかの行動を行うことが許されているユーザであるかどうかをシステムが確認する行為である。後で第 3.2 節に述べるように、システムにおける利用権限はユーザに対して与えられる。しかし、利用者が、利用権限を与えられたユーザと同一人物である、という保証は無い。そのため、利用者がユーザと同一人物であるかどうかを確認する必要がある。

認証する方法は以下の 3 種類があり、このうちの 1 つまたはいくつかを使用する [5]。

1. 知っている事柄（例えば、パスワード）をシステムに知らせる（知識要素）
2. 持っているもの（例えば、カードキー）をシステムに示す（所有要素）
3. 自分の何らかの部分（例えば、指紋）をシステムに計測させる（バイオメトリクス要素）

しかし、これらのどれもが完璧な方法ではない。それぞれの危険性を以下に示す。

1. 通信回線からパスワードは盗聴される可能性がある。また、キーボードでパスワードを入力している際に、後から覗き込まれる可能性がある

2. カードキーは強奪される可能性がある。置き忘れたカードキーを盗まれるかもしれない
3. 指紋のコピーをとって認証を成功させる技術も存在する

様々なサービスにおいて、様々な形態の認証が行われているが、基本的には上記の方法のいずれか 1 つあるいは複数の組み合わせによって行われている。次に、これらの認証に成功した際に与えられる利用権限について述べる。

3.2　UNIX における権限

本研究では、実装の対象として UNIX 互換 OS を選んだ。その理由は、サーバなどの用途で広く使われているものの決して管理が容易とは言えず、改善の余地があると考えたこと、仕様が公開されており、OS そのものに手を加えるような変更も可能であること、である。本節では、この UNIX 互換 OS における様々な権限について述べる。

UNIX OS や UNIX 互換 OS における、システムの利用に関する権限には様々なものがある。ここでは、それらの権限について述べる。本論文では、以降 UNIX OS と UNIX 互換 OS とをあわせて単に UNIX と呼ぶ。UNIX におけるユーザの権限は、以下のように分けることができる。

- ユーザとグループ
- ファイルシステム

3.3　ユーザとグループ

UNIX にはユーザとグループという概念がある。システムの利用者それぞれに対して、システムにおけるユーザが割り当てられる。ユーザは、ユーザ名とユーザ ID とによって構成される。ユーザ ID は UNIX 内部でユーザを識別するために利用される整数である。ユーザ名は利用者がユーザを示すために用いられ、UNIX 内部ではユーザ名はユーザ ID に変換されて解釈される。ユーザ名はアカウントと呼ばれることもある。

3.3.1　ユーザ

ユーザは大きくスーパーユーザ、システム機能に割り当てられる特別なユーザ、そして利用者に割り当てられる一般ユーザに分けられる。スーパーユーザは、UNIX においてもっとも大きな権限を有するユーザであり、基本的にそのシステムに関する全ての権限を持つ。ほぼ全ての UNIX において、スーパーユーザのユーザ名は root であるため、スーパーユーザのことを慣例的に root と呼ぶこともある。

多くの UNIX には、スーパーユーザ以外にも、特別な機能のために用意されたいくつかのアカウントが存在する。例えば以下のアカウントが、システム上の特別な機能を扱うために標準的に用意されている。

root　スーパーユーザ、アカウントの操作やその他全てのシステム機能を実行する

daemon　ネットワーク機能の一部を扱う

ftp　anonymous FTP アクセスに使用する

uucp　UUCP システムを制御する

news　Usenet ニュースに使用する

nobody　特権の無い操作のデフォルトユーザとして使用される

また標準ではないが、サードパーティのアプリケーションの中には特別に用意されたユーザによって実行されるものもある。以下に例を挙げる。

www　WWW サーバである Apache が使用する

pgsql　SQL サーバの PostgreSQL が使用する

bind　name server の bind が使用する

これらのアカウントには、一般的にパスワードを割り当てない。これは、人間のユーザがこれらのアカウントにログインすることを防止している。

一般ユーザのアカウントは、実際にシステムにログインして活動する人間の利用者に対して割り当てられる。一般ユーザは、特別なユーザのみが実行を許された一部を除いて大部分

のアプリケーションを実行することができる。また、一部のシステムにとって重要なファイルを除いて、あるいは他のユーザの所有ファイルであって他人の読み出しを禁止されているファイルを除いて、ファイルを読むことができる。さらに一般ユーザは、自らの所有するファイルを読み、書くことが可能である。これらのファイルの操作に対する権限は第 3.4 節において詳細に述べる。

3.3.2　グループ

UNIX においては、ユーザは必ず 1 つまたは複数のグループに所属している。ユーザがユーザ名とユーザ ID とによって管理されていたように、グループもまたグループ名とグループ ID とによって管理されている。

グループは、複数のユーザをまとめて扱うために使用する。グループを使用することによって、ユーザ群に対して、ファイルやディレクトリ、デバイスの読み出し、書き込み、実行をまとめて管理することが可能である。

多くの UNIX では、システム管理者のグループとして wheel を用意している。このグループに所属しているユーザは su コマンドを利用してスーパーユーザに権限を移行することができる。これによって、システム管理者を複数人で担当することが可能となる。

3.4　UNIX ファイルシステム

UNIX ファイルシステムは、ファイルなどの情報を補助記憶装置に保管する方法を制御すると同時に、どのユーザがどの情報にどのようにアクセスできるかも制御する [5]。

第 3.3.1 節や第 3.3.2 節で述べたファイルの読み出しや書き込みなどの許可、不許可を制御する仕組みはファイルシステムによって実現されている。この機能のことを、パーミッションと呼ぶ。

以下に、パーミッションの例を 2 つ示す。

```
-rw-------
```

```
drwxr-xr-x
```

先頭の文字はファイルの型を表している。通常のファイルは「-」であり、ディレクトリは「d」である。後に続く 9 文字は、3 文字で 1 つのグループとなっていて、そのファイルに対して、誰がどのような権限を持っているかを表している。まず各グループにおいて 3 文字の表すものは、そのファイルに対してなにができるか、ということである。文字の意味を以下に示す。また、これらのパーミッションについて表 3.1 に詳しく示す [5]。

r　読み出しの権限（パーミッション）

w　書き込みの権限（パーミッション）

x　実行の権限（パーミッション）

この 3 文字が集まったグループも同様に 3 つあり、それぞれ順に以下に示す対象のパーミッションを表している。

所有者　ファイルの所有者

グループ　グループに属するユーザ

その他　上記以外の全てのユーザ（スーパーユーザを除く）

前述の例を用いて、これらをまとめたものを図 3.1 に示す。

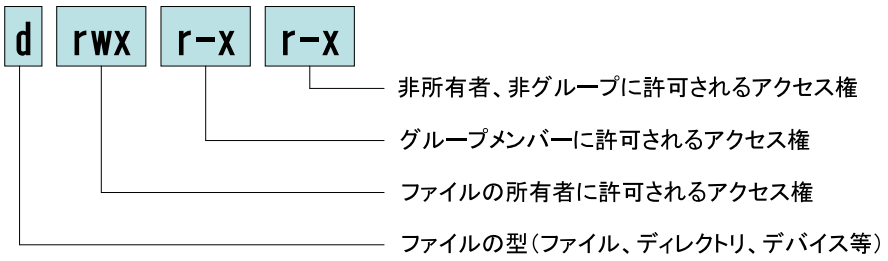

図 3.1　パーミッションの例

3.5　認証の応用

これまで、認証によってどのような権限が与えられるかについて述べてきた。次に、第 3.1 節で基礎的な部分について述べた認証について、実際の使われ方を挙げて述べる。

3.5.1　ログイン認証

UNIX などのワークステーションを利用する際に必要である事から、様々な認証の中でも利用することの多いものである。UNIX におけるログインの流れは以下である。

1. システムがログインプロンプト (login:) を出力
2. ユーザは、プロンプトに対してユーザ名を入力
3. システムがパスワードプロンプト (Password:) を出力
4. ユーザがパスワードを入力
5. システムが、入力されたパスワードが正しいか確認
6. 正しい場合は、該当するユーザ、グループの権限を適用する

文字	パーミッション	意味
r	読み出し	読み出し権限とは、ファイルをオープンし、その内容を読み出せるという意味である。
w	書き込み	書き込み権限とは、ファイルの上書きと内容の修正、および内容を削除できるという意味である。
x	実行	実行権限はプログラムとディレクトリに対して用いられる。実行権限がある場合、パス名を入力すればそのファイルを実行できる。ディレクトリに実行権限がある場合は、cd コマンドを利用してそのディレクトリへ移動することができる。

表 3.1　パーミッションの詳細

第 3.1 節で、認証の方法には 3 種類あることを述べた。この流れの通り、ログイン認証においては利用者が知っている何かをシステムに伝える、知識要素の方法が使用されている。これはログインのためにはユーザ名とパスワードの 2 項目を入力するだけでよいため使いやすいこと、事実上、使用に際して費用がかからない (無償である) こと、などの長所によるものである [7]。ただ最近では、カードキーなどの所有要素、指紋や虹彩によるバイオメトリクス要素を使用したログイン認証も行われるようになってきた。

3.5.2　リモートログイン

第 3.5.1 節で述べたログイン認証は、目の前のシステムに直接ログインする場合である。ネットワークに接続されているシステムでは、離れたところにあるシステムを経由して、リモートログインできる場合がある。

このリモートログインでは、多くの場合直接システムにログインする時とは違うプロセスによって認証が行われる。そのため、認証の手続きも直接ログインする時とは異なる場合がある。

最近よく利用されている ssh というリモートログインプログラムでは、公開鍵暗号方式による認証をすることができる。例えばユーザの公開鍵をリモートのシステムに登録しておくことで、手元の秘密鍵を使用してログインすることが可能である。この場合、システムに登録されたパスワードを入力する必要は無く、秘密鍵に登録されているパスフレーズを入力することになる。

このようなリモートログインであっても、ログイン後に与えられる権限は、直接ログインする場合と基本的に同じである。

3.5.3　Web 認証

Web における認証は、以下の場合に利用される。

- ある Web ページを特定の相手に対してのみ公開したい場合

- 商取引など、通信相手を特定したい場合

前者の場合、公開する相手の範囲が絞られていればよく、通信に際して個々の相手を特定できる必要性はそれほど高く無いこともある。この場合はパスワードのような知識要素を用いることがほとんどであり、相手を特定できる必要性が低い場合はユーザ名とパスワードは複数のユーザで共有することもある。この時得られる権限は、ある Web ページを閲覧することのできる権限である。場合によっては、その Web ページを編集する権限が与えられる場合もある。

後者の場合、通信相手を特定できる必要がある。また、金銭が関わる場合や個人情報がやり取りされる場合もある。そのため認証は厳重に行われ、ほとんどの場合通信は暗号化されて行われる。パスワードによる認証が多くを占めるが、知的要素を用いた認証は盗聴に弱いため、カードキーを用いた所有要素による認証が利用されるケースも増えている。この場合に得られる権限は、相手に本人と認められることであり、これによって相手と取り引きを行うことが可能になる。

3.5.4　SMTP, POP 認証

電子メールを送信する際に使用される SMTP や、受信する際に用いられる POP においても認証が行われる。 SMTP や POP における認証方法はパスワードやパスフレーズによる知識要素が主流である。

POP 認証は電子メールを受信する際に行われる。つまり、認証によって得られる権限はサーバにある電子メールを取得する権限である。電子メールは重要な個人情報が流れることもあり、パスワードの管理には十分気をつける必要がある。そのため、POP による通信を暗号化する仕組みや、電子メールそのものを暗号化してやり取りする仕組みが利用されるようになりつつある。

SMTP のサービスは、今まで認証を必要とせずに利用できることが多かった。しかし、近年になって SPAM と呼ばれる迷惑メールを送信するために利用されてしまうケースが増え

るにしたがって、ほとんどの SMTP サーバは必要の無いメールを扱わないようになってきた。多くの組織において、自組織内からの接続しか受け付けない設定が主流になると、例えば出張や営業などで組織外に出た際にメールを送信することができなくなるという弊害が発生するようになった。この問題を回避するために、必要に応じて SMTP の認証を行い、成功した人にのみ SMTP の接続を許可し、組織外からのメールの送信を可能にする手法ができた。つまり、SMTP 認証によって得られる権限は、外部からメールを送信することのできる権限である。

第 4 章

認証手法

本章では、認証に対して求められている事柄をまとめ、既存の認証手法の問題点を挙げる。さらにそれらを改善する手法を提案し、その提案を実現する認証手法の実装について述べる。

4.1 認証に求めるもの

第 3.1 節で述べたように、認証とは、利用者がシステムにおいて行動を許されているユーザであるかどうか、システムが確認する行為である。認証に成功すると、ユーザはそのシステムにおいて与えられた権限に基づいて行動することができる。

ユーザが各システムにおいて利用するプログラムはそれぞれのシステムにおいて異なる可能性がある。また、システムが与えることのできる権限は、システムによって異なる可能性がある。そのため、新たなユーザを登録するにあたって、管理者は以下にあげる点に注意し、必要以上の権限を与えないようにする必要がある。

- ユーザの求める利用権限を調べる
- ユーザに与える権限を決定する
- 利用する可能性のある全てのシステムにおいて、ユーザに適切な権限を適用する

しかし、既存のユーザ認証の仕組みでは、これらを実現することは困難であったり、大きな管理コストを必要とする。

4.2　既存の認証方式における問題点

既存の認証手法において、第 4.1 節で述べた要求を実現する際に問題となる点を示す。

UNIX におけるログイン認証では、ユーザに対してシステムの利用を許すか許さないか、という両極的な制御しかできない。そのため、必要の無い権限を与えずにシステムの利用を許可することは難しい。ログイン認証の他には、ファイルシステムの機能によって、プログラムの実行権限を操作することは可能である。しかし、各ユーザに対して必要な権限のみを与えるためには、プログラムの実行権限もユーザごとに設定する必要がある。したがって、多くのプログラムの実行権限をそれぞれ設定するための管理コストは非常に大きなものとなってしまう。また、システムの数が増えるにしたがってコストも増大するため、既存の手法によって要求を満たそうとすると、スケールしない。

4.3　改善手法の提案

第 4.2 節で述べたように、既存の認証手法では要求を満たすことができない。本節ではこれを改善する手法を提案する。

ここに、そのシステムを利用することのできるユーザ A と、利用することのできないユーザ B がいると仮定する。ユーザ A がユーザ B を信頼しているとき、A の持つ権限の一部を B に与えることにする。すると、信頼する側の持つ権限に基づいて権限が与えられるため、新たなユーザ B はシステムを利用するために必要な権限を自動的に得ることができる。これを実現するために、第 2.2.1 節で述べた信頼の輪モデルを導入する。これによって、ユーザ A がユーザ B を信頼しているという情報が存在するとき、システムは間接的に B を信頼することが可能である。そして、ユーザ A の持つ権限の一部をユーザ B に適用することが可能である。

4.4 sudo

今回は、権限の一部を与える方法として、sudo[*1]を用いる。sudo は多くの UNIX[*2]または UNIX 互換 OS において動作するアプリケーションであり、一般ユーザに対して super user の権限を委譲するツールとして利用されている。ほとんどの UNIX および UNIX 互換 OS において動作するため特定の OS に依存してしまうことも無く、多様なシステムが存在する環境に導入することも可能である。さらに sudo は設定が非常に柔軟であり、簡単な設定で十分に適切な権限を与えることが可能である。

sudo コマンドの動作メカニズムを図 4.1 に示す。

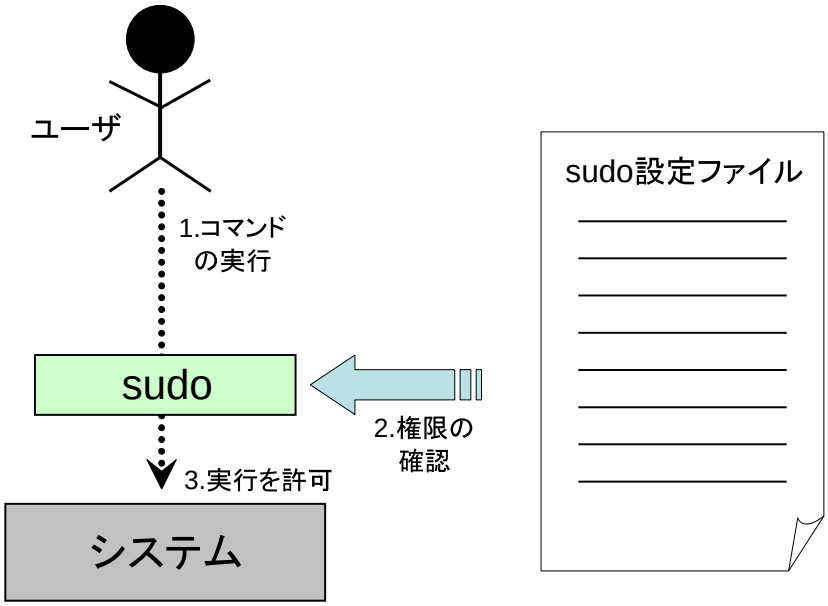

図 4.1 sudo コマンドの動作メカニズム

sudo を実行すると、以下のように動作する。

1. ユーザがコマンドを実行しようと試みる

[*1] http://www.courtesan.com/sudo/
[*2] http://www.opengroup.org/public/prods/xxm0.htm

2. sudo が設定ファイルを読み、権限を確認する
3. 与えられた権限に基づき、コマンドは実行される

ユーザがどんなコマンドを実行することができるのかは、sudo の設定ファイルである sudoers に書かれている。sudoers はテキストファイルであり、エディタで簡単に編集することが可能である。sudoers における設定は、以下の 4 つの項目が中心となる。

- ユーザ
- 実際のコマンド実行者
- ホスト
- コマンド

ユーザは、sudo を実行したユーザである。これは、コマンドを実行するユーザを制限するために利用される。例えば、あるコマンドをユーザ A は実行可能だが、ユーザ B は実行不可能であるようにする場合などである（図 4.2）。

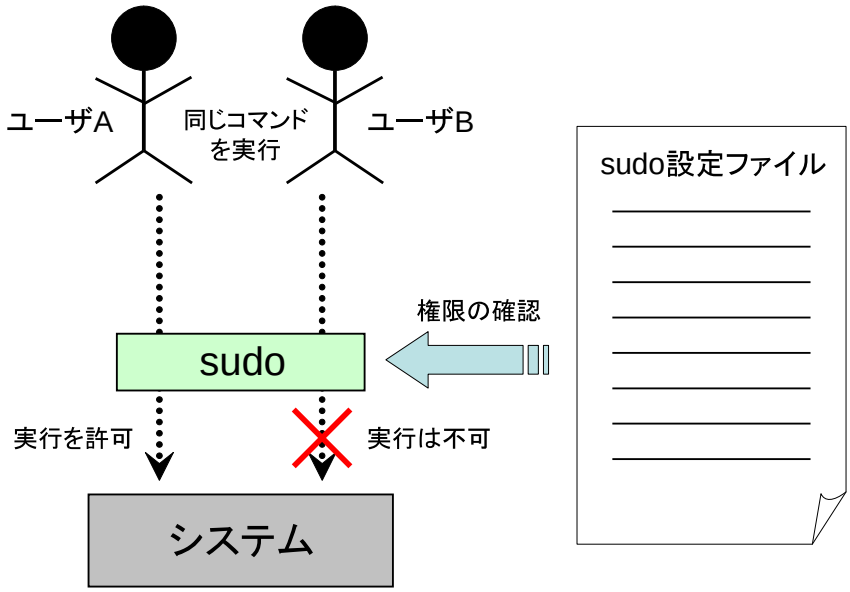

図 4.2　sudo の設定:ユーザ

実際のコマンド実行者とは、sudo によって実行されるコマンドの実行者となるユーザである。これによって、ユーザは、自分とは違うユーザとしてコマンドを実行することができる。コマンドを実行すべきユーザが決まっているものの、別のユーザが実行したい場合などに利用される。あるコマンドが、ユーザ B として実行することが許可されているとする。ユーザ A が、ユーザ B としてコマンドを実行しようとすると許可され、実行できる。しかしユーザ C としてコマンドを実行しようとすると、実行できない（図 4.3）。

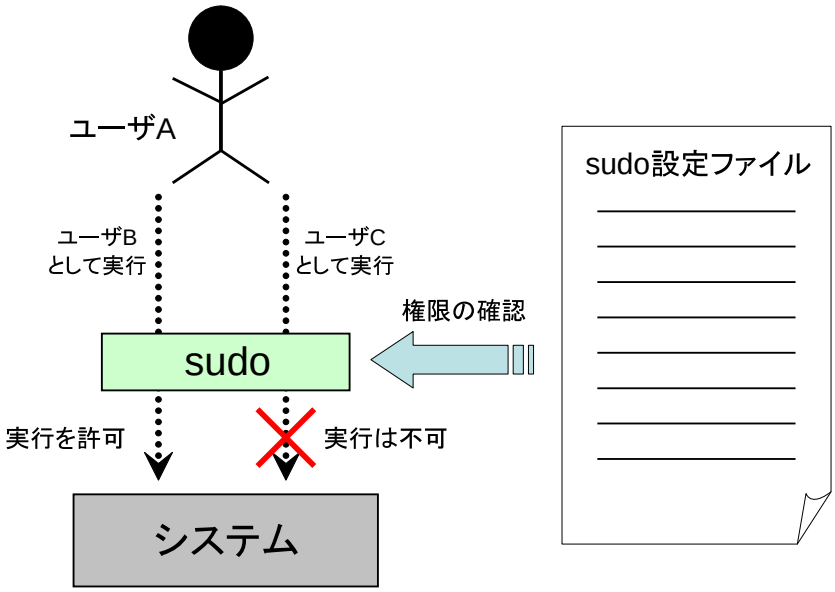

図 4.3　sudo の設定:実際のコマンド実行者

ホストとは、sudo が実行されたホストを表す。これによってホストが複数存在する場合に各々のホストにおいて異なる権限を与えることが可能となる。例えば、ユーザ A があるコマンドを実行しようとしたとき、ホスト A においては許可されるが、ホスト B においては許可されない、という設定が可能である（図 4.4）。そのため、ホストの記述によって、全てのホストにおいて同じ sudoers ファイルを使用することが可能となる。

コマンドとは、sudo を通して実行されるコマンドである。そのシステムに用意されているコマンドのうち、制限されたコマンドだけを実行させることができる。例えば、あるユー

ザ A に対して、コマンド A の実行は許可するが、コマンド B の実行は許可しない設定が可能である（図 4.5）。

4.5　提案の実装

第 4.4 節で述べた sudo を利用して、提案した機能を実装する。この実装の現時点での機能は、ユーザの信頼関係を調べ、適切な権限が適用された sudo の設定ファイルを出力する。この実装は多くのプラットフォームにおいて動作する perl 言語で記述するため、プラットフォームへの依存性を減らすことが可能である。

信頼関係は、公開鍵暗号における電子署名によって記述され、署名を検証することによって信頼関係を確認することが可能である。電子署名の仕組みについて、第 4.6 節に詳しく述べる。提案した手法が動作するメカニズムを図 4.6 に示す。

図中の番号に従って、以下のように動作する。

1. ユーザ A がユーザ B を信頼する

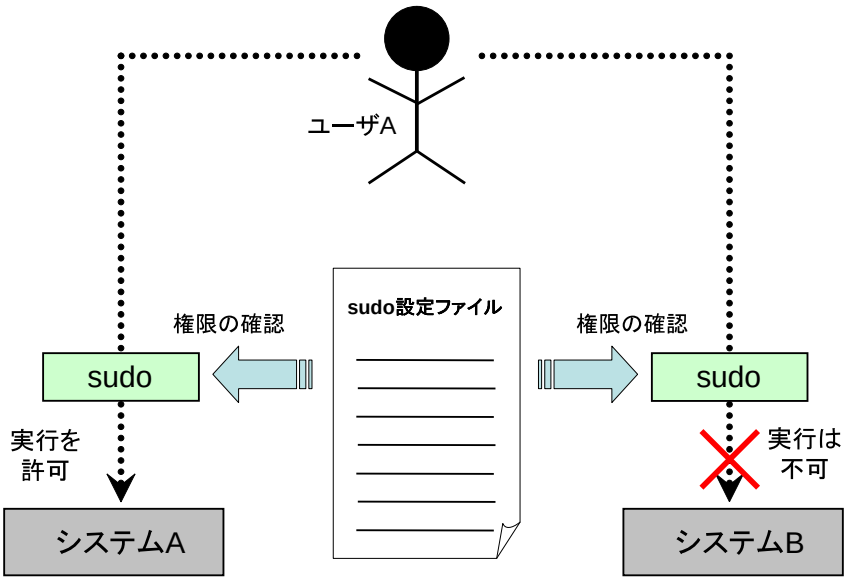

図 4.4　sudo の設定:ホスト

2. 提案された手法により、信頼関係が sudo の設定ファイルに反映される
3. ユーザがコマンドを実行しようと試みる
4. sudo が設定ファイルを読み、権限を確認する
5. 与えられた権限に基づき、コマンドは実行される

この動作について詳細に述べる。

(1) まず、ユーザ A はユーザ B が信頼に値するかどうか判断する。信頼できると判断した場合、その判断に基づいて、ユーザ B を信頼する。つまり、ユーザ A はユーザ B が信頼できるとして電子署名を行う。

(2) PGP の署名によって定義されている信頼関係をチェックし、sudo の設定ファイルを開く。ファイルの中の元のユーザが持つ権限を調べ、信頼されているユーザの設定としてコピーする。

(3) 新たなユーザが、sudo を用いてあるコマンドを実行する（しようと試みる）。

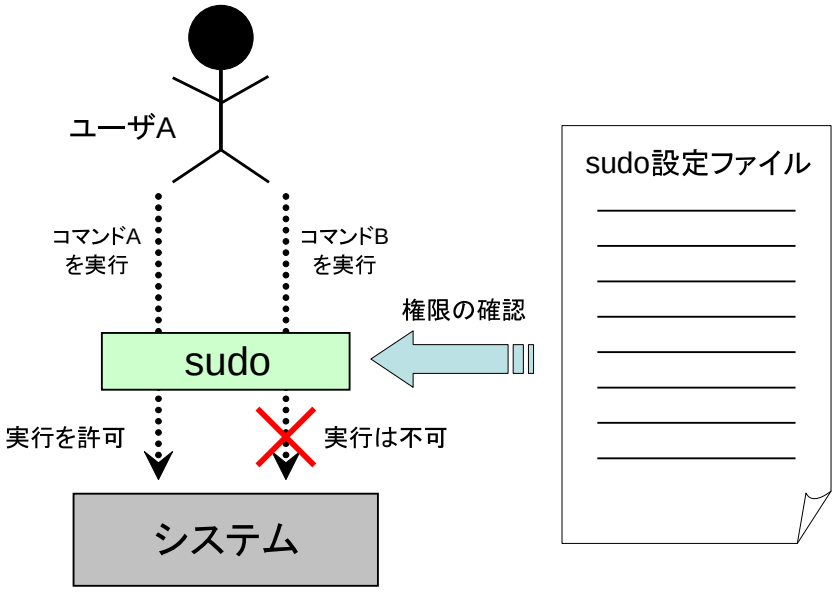

図 4.5　sudo の設定:コマンド

(4) sudo は設定ファイルを読む。ユーザが、そのコマンドを実行する権限を持っているかどうか確認する。

(5)（権限を持っているので、）コマンドは実行される。

4.5.1　信頼関係の設定

信頼関係は、PGP の公開鍵ホルダーによって実現される。PGP における個人認証は信頼の輪によって行われることは第 2.2.2 節に述べた。この信頼の輪は、公開鍵と、それに対する署名とで構成される公開鍵ホルダーによって保持される。この公開鍵ホルダーについている鍵証明書には、次の 2 つのパラメータ（判断基準）がある [4]。

- 有効性

  自分が持っている鍵に書いてある所有者が、本当にその鍵の所有者であるかを表す度合い。

- 信頼度

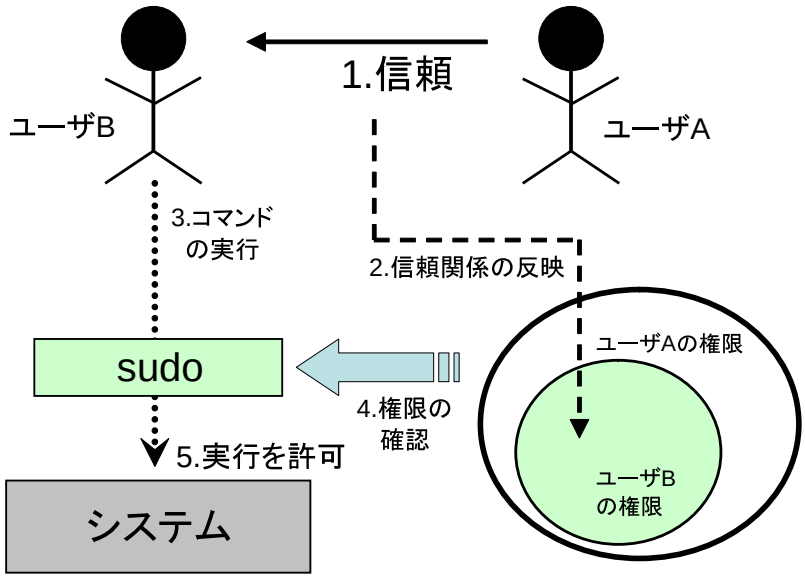

図 4.6　提案手法の動作メカニズム

ある鍵の所有者をどれくらい信頼するかを表す度合い。信頼度が高くなるほど、その鍵の所有者が他の鍵を証明したことをより強く信じるようになる。

信頼の輪におけるパスが遠くなると、鍵の有効性が低くなる。このパラメータをチェックすることにより、元のユーザに対してどの程度権限を減少させるかの基準にすることが可能である。これについては、第 5.2.1 節において述べる。

4.6　電子署名

PGP においても、PKI においても、ノードを信頼する行動に電子署名を用いている。本節ではこの電子署名について述べる。

4.6.1　非対象暗号

まず電子署名の基礎となる公開鍵暗号方式について述べる。公開鍵暗号による暗号化通信を図 4.7 に示す。

まず、図の左側で、平文と公開鍵を公開鍵暗号アルゴリズムに入力する。そして出力された暗号文を、図の左から右へ送信する。図の右側で、暗号文と先に使った公開鍵に対応する秘密鍵を公開鍵暗号アルゴリズムに入力する。すると、もとの平文が出力される [7]。

公開鍵と秘密鍵とは対になっており、公開鍵で暗号化された平文は、対応する秘密鍵によってのみ復号することができる。暗号文を公開鍵で復号することはできない。そのため、誰でも公開鍵で平文を暗号化することが可能であり、その暗号文は対応する秘密鍵を所有する人物にのみ復号可能である。暗号化された通信によって、データの機密性を得ることができる。

逆に、秘密鍵を用いて暗号化された暗号文は対応する公開鍵を持つ誰にでも復号できる。

4.6.2　電子署名

第 4.6.1 節で説明した公開鍵暗号方式の特徴を利用して、電子署名をすることができる。図 4.8 に電子署名の仕組みを示す。

図の右側において、まず平文がハッシュ関数に入力され、ハッシュ値が出力される。ここでハッシュ関数とは、様々な文字列をある一定の固定長の文字列に変換する関数である。ハッシュ値は文書やファイルごとに一意になり、元の文書が 1 文字でも違うと、ハッシュ値は大きく異なる。このハッシュ値と秘密鍵とを非対称暗号アルゴリズムに入力する。その出力を電子署名と呼ぶ。

元の平文と電子署名とが左に送られる。左側において、平文はハッシュ関数に入力され、ハッシュ値を得る。同時に、電子署名と公開鍵とを公開鍵暗号アルゴリズムに入力し、得られたハッシュ値と、先に求めたハッシュ値とが一致するかどうかを確認する。この行為を、署名の検証と呼ぶ。

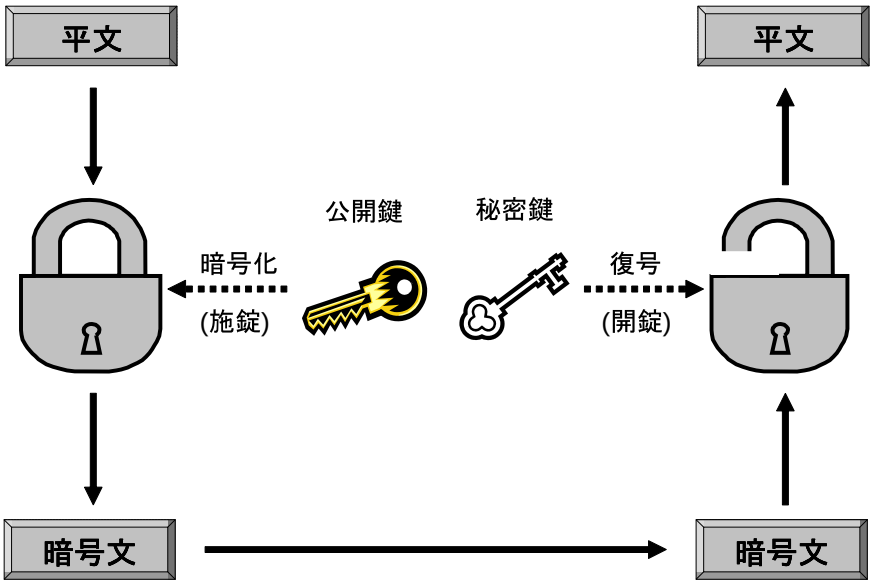

図 4.7　公開鍵暗号方式

検証が成功 (2 つのハッシュ値が一致) したなら、平文は改竄されていないことがわかる。検証できなければ、平文か電子署名かのいずれか、あるいは両方に変更が加えられたことになる。

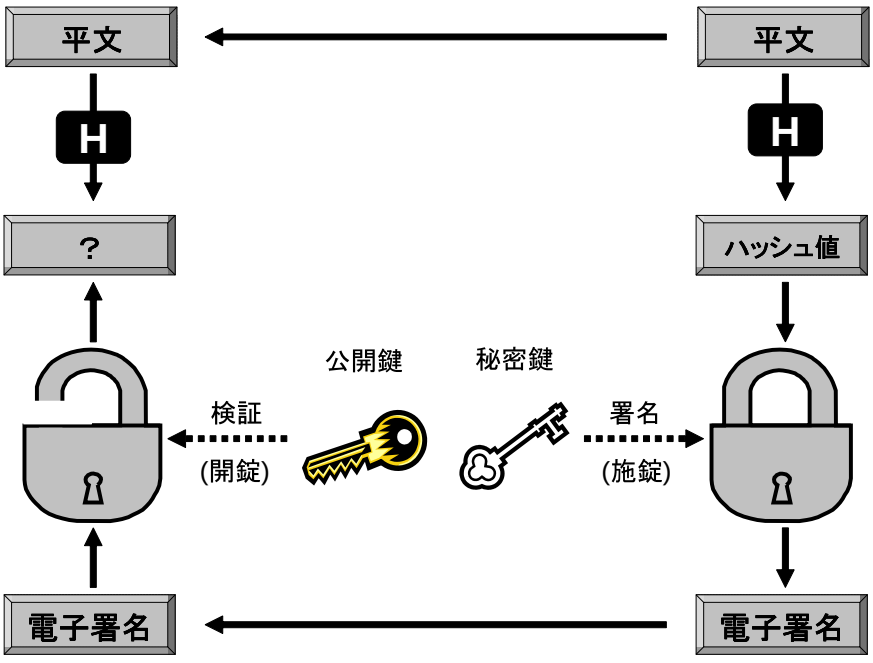

図 4.8　電子署名

第 5 章

提案の評価と考察

本章では、提案した手法が実際に管理コストを下げられるかどうか、評価を行う。また、提案した手法に対する考察を行う。

5.1 提案の評価

本研究における手法の提案には、以下に示す 2 つの段階がある。

- 個人認証に対して信頼の輪を適用した
- 今回の実装の目標として、sudo を選んだ

それぞれの段階における問題点を示し、評価を行う。

5.1.1 信頼の輪モデルの適用

個人認証に信頼の輪モデルを適用する本提案のモデルには、以下の問題点を挙げることができる。

1. 間接的な信頼関係がセキュリティ上の危険を招く可能性がある
2. 間接的な信頼関係に基づいて、ユーザに与える権限を決定するための基準をどうするか

これらの問題点に対する解決策を述べる。

1 つ目の問題点

1 つ目の問題点は、間接的な信頼関係に基づいて与えられるシステムの利用権限の委譲が、セキュリティ上の危険を招く可能性があるということである。従来の手法であれば、システムの管理者が権限を委譲するため、管理者も気付かないうちに権限が与えられてしまうというような問題は起きない。本提案の手法と従来の手法とを、以下の 3 点において比較した結果を、表 5.1 に示す。

(1) 管理者の負担

アカウント管理における管理者の負担の大きさ

(2) 意図しない権限委譲による危険性

管理者の知らないうちに、意図しない権限の委譲が行われてしまう危険性の大きさ

(3) ユーザに与える権限

管理者による設定や他のユーザからの信頼などによってユーザに対して与えられる権限の大きさ

手法	(1) 管理者の負担	(2) 意図しない権限委譲による危険性	(3) ユーザに与える権限
本提案の手法	小さい	大きい	小さい
従来の手法	大きい	小さい	大きい

表 5.1　セキュリティにおける手法の比較

今回の問題である (2) の、意図しない権限委譲による危険性は、管理者がユーザの信頼関係を監視することによってある程度防ぐことが可能である。しかしそれは管理コストの増加を招くため、(1) に挙げた、管理者の負担が小さいという本提案の利点を損ね、当初の目的を果たすとは言い難い。とはいえ、そのユーザに権限を与える必要があるとすれば、(3) に示すように必要最低限の権限のみを与えることのできる本手法は、不必要な権限も与えてし

まう従来の手法に比較して安全であると言える。

2 つ目の問題点

次に、2 つ目の問題点について述べる。本提案では、第 4.3 節に述べた通り、間接的な信頼関係に基づいて、元のユーザの持つ権限の一部、あるいは全部を与える。一部を与える場合も、全部を与える場合も、それぞれに問題があるといえる。問題を以下に挙げる。

1. 一部を与える場合

   一部を与える場合は、元のユーザの持つ権限のうちどれを新しいユーザに委譲し、どれを委譲しないか、を決定する必要がある。当然のことながら最適な解は状況に依存するため、これを決定するのは困難である。

2. 全部を与える場合

   全部を与える場合は、一部を与える場合とは別の問題が発生する。それは、あるユーザが信頼した新しいユーザが、実際にどの程度信頼に値するかわからないことである。第 2.2.2 節に述べたように、間接的な信頼関係を 1 つ経る (パス長が長くなる) ごとに信頼度は低くなって行く。そのため、新しいユーザに対して元のユーザの持つ権限を全部委譲するのは危険性が高いと言える。

前者の一部を与える場合における解決策として、以下の 2 つを挙げる。

- パス長に応じた一定の規則によって権限を減らす
- 元のユーザの持つ権限を解析して、不要な権限を減らす

後者の、全部を与える場合における解決策としては、以下を挙げる。

- ユーザに対して、確かに信頼できると言えるユーザのみを信頼するように指導し、安易に他のユーザを信頼しないようにする

一部の権限を与える場合における解決策は、どちらの場合も何らかの基準を作る必要がある。どのような基準が適当であるか、今後の課題である。

全部を与える場合の解決策は、技術的な解決はできておらず、モデルの問題点をユーザに押し付けてしまっている。そのためユーザにとって使いづらい手法になってしまうと考えられる。全部の権限を与えるモデルの検討も今後の課題である。

5.1.2　sudo の採用

今回目標を置いた実装では、ユーザに対して権限を与える手段として sudo を選択した。これには、以下に挙げる問題点が存在する。

1. どのコマンドを実行するにも sudo を通す必要がある
2. ファイルやデバイスのパーミッションを与えることができない

これらの問題に対する解決策を述べる。

1 つ目の問題

1 つ目の問題点は、この実装では標準で誰にでも実行可能なコマンドを除いて、本手法によって権限を得たどのコマンドを実行するにも sudo を用いる必要があるという点である。sudo を通してコマンドを実行することそのものは大きな手間ではない。しかし、多少なりともタイプ量が増えることは事実であるし、また、今まで慣れてきたコマンド操作が変化してしまうことによる、感覚上の運用コストの増大は否定できない。

この問題は、sudo を使用する限りついてまわる問題である。本提案では、管理コストを低減できることを示してきたものの、そのかわりユーザに対して運用コストの増大を強いることになってしまう。shell の alias 機能を用いることで、見かけ上 sudo を呼び出す操作を隠蔽することは可能である。しかし委譲される権限が増えるにしたがって、alias すべきコマンドも増えて行くため、運用コストの増大は避けられないといえる。これに対しては、

sudo に依存しない実装の開発や sudo の存在を隠蔽する仕組みを考案するなどの解決法が考えられ、今後の課題として取り組む必要がある。

2 つ目の問題

2 つ目の問題点も sudo の実装に依存する問題である。sudo がユーザに対して許可するのは第 4.4 節に述べた通りコマンドの実行である。すなわち、以下に挙げる行動を許可したり制限したりすることができない。

- ファイルやディレクトリに対する、読み出しや書き込み
- デバイスの使用
- ソケットへのアクセス

これも 1 つ目の問題と同じく、sudo の実装に依存する問題であり、基本的に解決策が存在しない。1 つ目の問題の解決策においても述べた通り、sudo に依存しない実装を開発することがこれらの問題の解決策となると考える。そのためには、どのような権限が元のユーザから新たなユーザへ委譲されるべきか、という点についてさらに考察し、必要な権限を委譲するために適した実装を選択すべきである。

5.2　提案の考察

今回の実装における目標は、個人間の信頼関係を個人認証に取り入れることによる権限委譲のモデルが動作し得るかどうかを検証することにあった。そのため、提案している手法の全てを実装するわけではない。

本節では、今回の実装から離れ、個人認証に対して信頼の輪モデルを導入した場合にどのような応用が考えられるか、ということについて考察する。

5.2.1　パス長による信頼度の変化

信頼の輪モデルでは、第 2.2.2 節の通り、信頼関係のパスが長くなればなるほど相手の信頼度が減る。この特徴を利用して、パス長による信頼度の減少にあわせて相手に与える権限を減少させる手法が考えられる。

第 4.5.1 節に述べたように、PGP の公開鍵ホルダーには、信頼関係のパスが長くなるにしたがって減少するパラメータが存在する。このパラメータを利用することで、パス長による信頼度の減少を定義することが可能である。この場合、第 5.1.1 節の 2 つ目の問題において述べた、元のユーザの持つ権限を減少させる基準が必要となる。

5.2.2　複数の信頼関係による権限の合成

信頼の輪モデルでは、図 2.5 に示すように、ある 2 人の間を繋ぐ信頼関係は複数ある場合がある。この図を例にとると、B と E とを結ぶ信頼関係は以下の 3 つある。

- $B \rightarrow D \rightarrow E$
- $B \rightarrow C \rightarrow E$
- $B \rightarrow C \rightarrow D \rightarrow E$

この時、E が C から受け取る権限と、D から受け取る権限とが異なるとする。その場合、2 人の持つ権限が合成されて元の C や D の持つ権限より若干大きい権限を与えることも可能である。

この概念は、新たなユーザが複数のプロジェクトに参加する場合に有効である。先の図を例にすると、C と D とは別のプロジェクトに属している。C は C のプロジェクトにおける権限を、E を信頼することで委譲する。D は、D のプロジェクトにおける権限を、E を信頼することで委譲する。結果として E は、C のプロジェクトにおいても、D のプロジェクトにおいても、活動に必要な権限を得ることができる。信頼関係を複数集めて合成できるようになると、より柔軟な権限の委譲が可能となる。

しかし、柔軟な委譲が可能になるということは裏を返せば関係が複雑になるということである。まず、信頼関係を調べることが難しくなり、謝って必要以上の権限を与えてしまうという可能性がありうる。また、信頼関係が複雑になると、どの権限を誰から委譲されているのか、という流れを追うのが困難になる。このことも、必要以上の権限が割り当てられてしまうことを発見しにくくする。

柔軟な設定を可能にしつつ、全体の信頼関係の見通しをよくする工夫が必要であると考える。

第 6 章

まとめ

本研究では、個人間の信頼関係に基づいて構築された信頼の輪の概念を用いることにより、適切なシステムの利用権限を与える手法を示した。信頼の輪は、個人と個人との信頼関係が複数存在する時、それらを合成することによって構築される個人認証のモデルである。このモデルによって、直接知らない相手を、間接的に信頼することができる。

この手法を用いることで、従来管理者に集中していたユーザ管理のコストを分散することが可能となる。さらに従来の手法では困難であった、ユーザに与える権限を細かく制御することも可能となる。

本手法は、本来不要である権限を割り当ててしまう従来の手法に比べ安全にユーザ管理を行うことができる。

謝辞

本研究を始めるきっかけとなる貴重な御意見を下さいました、独立行政法人 通信総合研究所 非常時通信グループリーダーの大野浩之氏に深く感謝いたします。

本研究を進めるにあたって、貴重な御討論、御助言をいただきました、独立行政法人 通信総合研究所 非常時通信グループのメンバーの方々、ならびに YDC の北島さん、川俣さん、そして東京工業大学大学院の木本雅彦氏にお礼申し上げます。

副指導教員として御指導、御助言を頂きました寺田浩詔副学長に厚くお礼申し上げます。

本研究を進めるにあたり、貴重な討論、ご助言を頂きました菊池研究室の皆様に感謝し、お礼申し上げます。

指導教員の菊池豊助教授には、本研究活動を通して御指導、ご協力いただきました。ここに感謝の意を表します。

参考文献

[1] ITU-T Recommendation X.509 (03/00): Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate, March 2000. X.509.

[2] ITU-T Recommendation X.500 (02/01): Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services, February 2001. X.500.

[3] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. OpenPGP Message Format, November 1998. RFC2440.

[4] Simson Garfinkel. PGP 暗号メールと電子署名. オライリー・ジャパン, May 1997. 山本 和彦 監訳 株式会社 ユニテック 訳.

[5] Simson Garfinkel and Gene Spafford. UNIX & インターネットセキュリティ. オライリー・ジャパン, December 1998. 山口 英 監訳 谷口 功 訳.

[6] R. Housley, W. Polk, W. Ford, and D.Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, April 2002. RFC 3280.

[7] トム・オースティン. PKI Public Key Infrastructure 公開鍵基盤 電子署名法時代のセキュリティ入門. 日経 BP 企画, August 2001. 株式会社ニューコム.

[8] 株式会社クニリサーチインターナショナル. PGP 実践活用ガイド Windows 版. オーム社, December 1998.