環状型信頼モデルに基づく企業向け認証システム ASE の提案

中　根　康　平

インターネットでは，公開鍵によるセキュリティ基盤 PKI(Public Key Infrastructure) が広く利用されている．PKI は，公開鍵の認証局 (CA) が階層構造になっており，最上位機関として root CA が存在する．しかし，root CA 自身の公開鍵を証明する機関がないため，root CA 自身で自己署名する．root CA の自己署名は，クライアントサーバがあらかじめ保持しておく．しかし，自己署名では，発行者が正当であることを検証する方法がない．そのため，root CA の公開鍵は，偽装される可能性がある．

また，PKI では，発行した公開鍵証明書の有効性を確認するために，失効情報を管理しなければならない．失効情報は，原則的に増加し続けるため，管理負荷が大きい．本稿では，このような PKI の課題を解決する認証システム ASE(Authentication System for Enterprise Network) を検討し，その実現を試みた．ASE では，信頼関係の構築を環状にする．これにより，全ての公開鍵証明書に第三者の署名がなされることになり，偽造を検出できる．また，公開鍵証明書を発行者自らが保持し管理を行うため，失効情報の管理が不要である．

Researches on Authentication System for Enterprise Network ASE having high security

Kohei Nakane

PKI (Public Key Infrastructure) is widely used in the Internet these days. The public key of the user node is certified by hierarchized CAs (Certificate Authorities), and the most significant CA is called the root CA. There is no organization that can certificate the root CA, so the public key of the root CA is self-signed by itself, and is maintained safely in the verifier. However, the problem is that the self-signed certificate is easily faked. In this paper, we propose a new authentication system called ASE (Authentication System for an Enterprise network). ASE has a ring structure of authentication, and the user terminal signs the public key of the root authentication server, so the fake of the public key can be detected. We have developed the proposed system and obtained a good performance.

1. はじめに

インターネット上には盗聴，不正アクセス，なりすまし，改ざん，否認といったネットワーク固有の脅威が存在する．これらの脅威を回避するため，公開鍵暗号を用いたセキュリティ基盤 PKI(Public Key Infrastructure) が広く利用されている．PKI は，公開鍵暗号方式の暗号化を利用して，ユーザに秘匿を提供し，署名を利用してユーザに認証，完全性，否認や拒否の機能を提供する仕組みである．

PKI では，各ユーザの公開鍵を信頼のおける認証局 (CA：Certificate Authority) が署名し，公開鍵証明書を発行する．CA の公開鍵は，更に上位の CA が証明書を発行する．しかし，最上位の CA(root CA) の公開鍵証明書を発行する機関がないため，root CA 自身で自己署名する．そのため，root CA の公開鍵証明書は，クライアントサーバがあらかじめ信頼できる方法で取得しておき，厳重に管理する必要がある．しかし，自己署名であるために，root CA の公開鍵証明書は偽造される可能性がある．例えば，ウイルス等の悪意あるプログラムが，ユーザが気づかないうちに root CA の証明書を偽造する可能性がある．

また PKI では，発行した公開鍵証明書を被発行者に渡すため，証明書の有効性を確認するために証明書が失効していないかどうか，その都度確認する必要がある．失効の確認に必要な情報は，原則的に増加し続けるため，管理負荷が大きいという課題がある．

本稿では，PKI の課題を解決し，かつ，管理理負荷の少ない企業認証システム ASE(Authentication System for Enterprise Network) を提案する．ASE の特徴は，公開鍵証明書の偽造を防ぐために，信頼関係の構築を環状にする．また，管理負荷を少なくするために，公開鍵証明書は発行者が保持して自ら管理を行うため，失効情報の管理が不要である．

このような新たな認証基盤を，一般のシステムに適用するには，標準化するなどの手順が必要となる．そこで本稿では，企業ネットワークのような閉じたネットワークへの運用を想定し，検討を行った．

以降，2 章で PKI の原理と課題について述べ，3 章で ASE の原理と詳細について述べる．4 章で ASE の実装方式について述べ，5 章でまとめる．

2. PKI とその課題

2.1 PKI の原理

公開鍵暗号では，公開鍵が正しいことが保証されている必要がある．そこで PKI では，公開鍵の正当性を保証するために，各ユーザやサーバの公開鍵を信頼できる認証局 CA が自身の秘密鍵で署名し，公開鍵証明書を作成する．公開鍵証明書の信頼性を確認するには，信頼関係を構築する必要がある．信頼関係の構築とはいくつかの CA が連携し検証対象の公開鍵証明書を確実に検証することである．図 1 に PKI の信頼関係を示す．ユーザやサーバの公開鍵証明書は CA により発行され，CA の公開鍵証明書は更に上位の CA により発行される．root CA の公開鍵は root CA 自身が自己署名する．root CA の公開鍵は信頼点であり，検証者があらかじめ安全な方法で取得し所持しておくことが前提である．

被検証者の公開鍵の有効性を検証するためには，認証パスの構築と検証が必要である．認証パスの構築では，認証の対象となるユーザの公開鍵証明書を取得し，証明書内の情報から，検証者の信頼点となる root CA までの公開鍵証明書まで関連づけられていることを確認する．認証パスの検証では，すべての公開鍵証明書において，署名内容が正しいか，有効期間が切れていないか，失効していないかなどを検証する．認証パスの構築と検証が問題なく終了することにより，公開鍵証明書の有効性検証が終了する．ここで失効とは，ユーザが秘密鍵を紛失した場合や，証明書の内容が変更された場合などに発行済みの公開鍵が無効になることを言う．

失効の確認方法には CRL(Certificate Revocation List) を用いる方法と，OCSP(Online Certificate Status Protocol) を用いる方法がある．前者は各 CA が CRL を発行し，各ユーザは CRL に検証対象の公開鍵証明書が記載されていないことを確認する方法である．CA は CRL を定期的な周期で発行しリポジトリへ保存する．各ユーザは公開鍵証明書の検証をする前にあらかじめリポジトリから CRL を収集しておく必要がある．

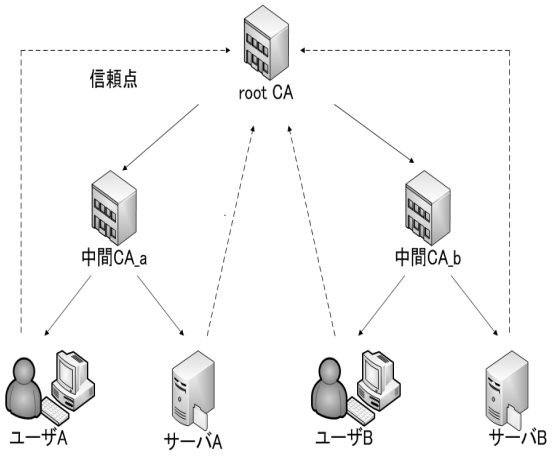

図 1 PKI の信頼関係

後者は公開鍵証明書の検証時に，失効状態を集中管理する OCSP レスポンダに対し，リアルタイムで有効性を確認する方法である．この方法では，OCSP レスポンダが CRL を収集する．検証者は，OCSP レスポンダに対し公開鍵証明書の状態を問い合わせる．OCSP レスポンダはその公開鍵証明書が失効していないか，失効情報との照合を行い，結果をユーザへ返答する．

図 2 は PKI において，ユーザ A がユーザ B の公開鍵を認証するために必要な情報と，それが保持されている場所を示す．root CA が中間 CA となる CA_b に公開鍵証明書を発行し，CA_b がそれを保持している．CA_b がユーザ B の公開鍵証明書を発行し，ユーザ B がそれを保持している．ユーザ B が所持する証明書の中には，CA_b が署名したユーザ B の証明書，root CA が署名した CA_b の証明書が含まれている．そのため，ユーザ A はユーザ B の公開鍵証明書を取得するだけでよい．この他に，ユーザ A は CA_b と root CA からそれぞれ各 CRL を取得し，ユーザ B および CA_b の公開鍵証明書が失効していないことを確認する必要がある．図 2 では CRL が所定のリポジトリに格納されているため，そこから入手する．

2.2 root CA の公開鍵証明書の偽造

このように，root CA の公開鍵証明書は，root CA 自身が自己署名するが，この公開鍵証明書の発行者が正当であることを検証する方法がない．すなわち，root CA の公開鍵は偽造される可能性がある．Windows では root CA の公開鍵証明書があらかじめユーザ端末のレジストリに保存されているが，このレジストリを直接操作することにより書き換えができる．そのため，ウィルスなど悪意あるプログラムなどにより書き換えられる可能性がある．

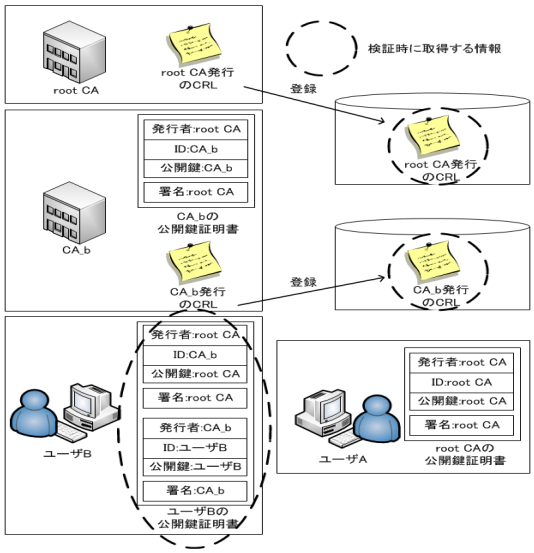

図 2　公開鍵証明書検証に必要なデータ

Root CA の公開鍵証明書が偽造されると下記のように悪用される可能がある．例えば，悪意ある第三者 A が公開鍵を偽造されたユーザ B に何らかの情報を要求する．ユーザ B は A の公開鍵証明書を取得して検証を行うが，その中には偽造された root CA の自己署名が記述されており，検証が問題なく終了する．そこで，ユーザ B は A を信用し秘密データを A に渡してしまうことになる．

このように root CA の公開鍵証明書が偽造された場合，PKI の仕組みの前提が崩れ重大な問題に発展する可能性がある．

2.3　失効情報の管理

PKI では，公開鍵証明書が発行者の手を離れ被発行者が所持している．そのため，特定のユーザの公開鍵証明書を失効させたくても，対象のユーザが公開鍵証明書の削除を行わず使用し続ける可能性がある．よって，公開鍵証明書が失効していないかどうか，失効情報として別途管理する必要がある．CRL は失効情報の管理方法の一つであり，失効情報のリストに CA が署名したものである．

失効情報は原則的に増加し続けるため管理が大変であり，失効情報のデータが大きくなると，有効性の確認時に多くの時間を要する．また CRL は，検証者が公開鍵証明書の検証をする前にあらかじめ収集しておく必要がある．

CRL は一般に定期的に更新される．そのため，公開鍵証明書が失効された場合でも，次回の CRL が発行されるまでは失効情報が利用者に伝わらず，最新の情報が手に入らない場合がある．OCSP を利用する場合においても，OCSP レスポンダの失効情報の更新は CRL を利用することが多く，必ずしも最新の情報であるとは限らない．

3.　提案方式 ASE

本章では提案方式 ASE について説明する．既に広く普及している PKI の仕組み自体を置き換えることは難しいため，以下の説明では，企業ネットワークのような閉じた世界をターゲットとし，検討した結果を述べる．

3.1　概　　要

PKI と ASE の違いは以下の通りである．まず PKI では，信頼関係を root CA の公開鍵証明書を信頼点として階層的に構築するのに対し，ASE では信頼関係を環状にする．即ち，社員がルート認証サーバの公開鍵に署名をし，root CA の公開鍵証明書の偽造を防止する．次に PKI では，公開鍵証明書が発行者の手を離れ，被発行者へ渡されるため，公開鍵証明書の有効性の確認が必要となる．それに対し ASE では，発行者自らが証明書を保持，管理する．検証者は，公開鍵証明書をオンデマンドで収集し，その時点で失効の有無を確認する．この方式により，PKI における失効情報の管理が不要となり，かつ，リアルタイム性の高い認証が可能となる．

3.2　信頼関係の構築

ASE の信頼関係を図 3 に示す．部門認証サーバは社員あるいは共有サーバの公開鍵を保証するための装置で，例えば部単位に設置する．部門認証サーバは複数の階層になっていてもよく，PKI における中間 CA に相当する．ルート認証サーバは企業の最上位に位置づけられるもので，PKI における root CA に相当する．矢印は公開鍵証明書の発行の方向である．ルート認証サーバは部門認証サーバに公開鍵証明書を発行する．部門認証サーバは各部門配下の社員や各サーバに公開鍵証明書を発行する．社員や各サーバはルート認証サーバに公開鍵証明書を発行する．このように信頼関係を環状にすることにより，全ての公開鍵証明書が正しいことを検証できる．環状の信頼関係を一度築けば，全ての公開鍵証明書は偽造を検出できるようになり，安全性が保証される．

また，全ての社員がルート認証サーバに署名するため，管理負荷が増加したり操作ミスが発生する懸念があるが，3. 5 節に述べるように IC カードなどのハードウエアトークンを導入することにより，これらの課題を軽減することができる．

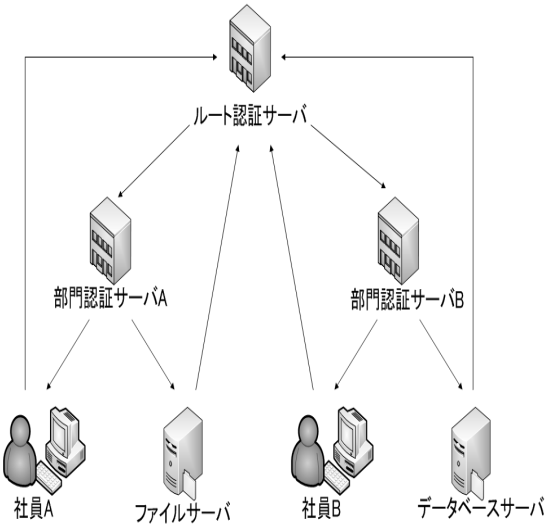

図 3 ASE の信頼関係

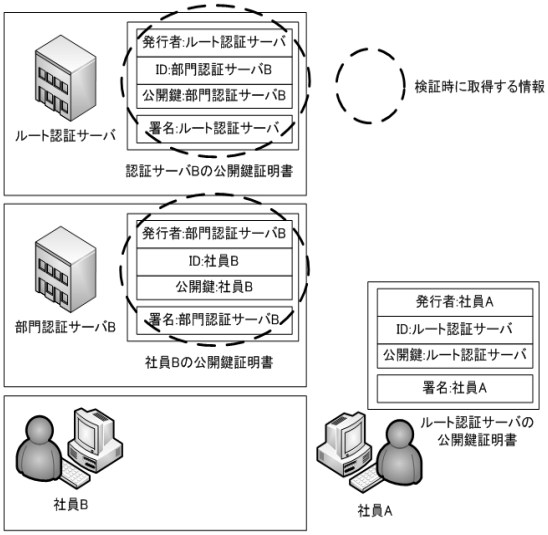

図 4 公開鍵証明書の管理方法

3.3 公開鍵証明書の管理方法

ASE の公開鍵証明書の管理方法を図 4 に示す．図 4 は，ASE において社員 A が社員 B の公開鍵を認証するために必要な情報とそれが保持されている場所を示す．ASE では発行した公開鍵証明書を発行者自身が保持する．即ち，社員 B の公開鍵証明書を発行者の部門認証サーバ B が保持している．また，部門認証サーバ B の公開鍵証明書は発行者のルート認証サーバが保持している．さらに，ルート認証サーバの公開鍵証明書は社員 A が保持している．認証時にはオンデマンドで必要となる公開鍵証明書をすべて収集する．このため，管理方法をこのように改めても特に問題は発生しない．この管理方法により公開鍵証明書が失効した場合は，管理している公開鍵証明書を単に削除するだけで済む．即ち，PKI の失効情報に相当するものは不要である．

3.4 公開鍵証明書の有効性検証

ASE における公開鍵証明書の有効性検証方法を図 5 に示す．必要となる情報は，検証者がオンデマンドで収集する．具体的な検証方法は以下のようになる．

まず，社員 A はルート認証サーバへ社員 B の所属を問い合わせる．ルート認証サーバは社員 B が所属している部門認証サーバ B のアドレスと，その公開鍵証明書を返送する．次に，社員 A は部門認証サーバ B へ社員 B の所属を問い合わせる．部門認証サーバ B は社員 B は自らの部員である旨と社員 B の公開鍵証明書を返送する．

上記手順により必要な情報は揃ったので，認証パスの検証へ移る．認証パスの検証は社員 A の公開鍵でルート認証サーバの公開鍵を検証し，ルート認証サーバの公開鍵証明書で部門認証サーバ B の公開鍵証明書を検証し，さらに部門認証サーバ B の公開鍵で社員 B の公開鍵証明書を検証する．すべての検証が成功した場合，社員 B の公開鍵は信頼することができる．この方法では，問い合わせた公開鍵証明書で最新の状態が確認できるため，失効情報の確認作業は不要となる．上記の手順で社員 B の公開鍵が正しいことが判明したので，社員 B を認証するために，更に下記手順を実行する．

社員 A は共通鍵を作成した後，社員 B の公開鍵で上記共通鍵を暗号化し，社員 B へ送信する．社員 B は自身の秘密鍵で復号し共通鍵を取り出す．次に，共通鍵を取得した旨を共通鍵で暗号化して A に返送する．これらの処理が正しく終了すれば，以後の通信には，上記共通鍵を用いて暗号化通信が可能となる．

3.5 証明書の発行手順

公開鍵証明書の発行手順は以下のように行う．簡単のため，図 3 のように信頼関係は 2 階層とする．まず，社員は自ノードで自身の鍵ペアを生成し，それに対する証明書要求を作成する．作成した証明書要求を部門認証サーバのところまで持っていく．部門認証サーバは受け取った証明書要求を自身の秘密鍵で署名し，公開鍵証明書を作成する．この証明書は，認証サーバがそのまま保持しておく．認証サーバと root CA の関係も同様であり，ルート認証サーバは受け取った証明書要求から公開鍵証明書を作成し，そのまま保持しておく．社員はあらかじめ生成してあるルート認証サーバの証明書要求に署名して，そのまま保持しておく．

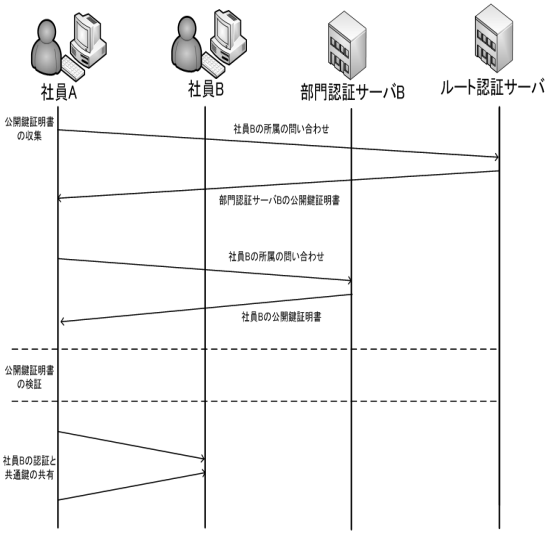

図 5 ASE における公開鍵証明書の検証方法

このような方式は，社員が署名する作業が必要となるため，運用が繁雑になる可能性がある．そこで，以下のように社員が IC カードを保持し，IC カードの発行を認証サーバで行うようにすれば，作業は簡素化される．図 6 に IC カードによる運用を想定した場合に，IC カードが持つべき情報を示す．認証サーバは各社員の ID 情報や鍵ペアとともに，社員の秘密鍵で署名したルート認証サーバの公開鍵証明書を生成し，IC カードに格納する．社員はこの IC カードを受け取る．この方法では，社員の鍵ペアも含めて全て認証サーバが生成することになるため，社員は発行に関する作業が不要になる．また，ルート認証サーバの証明書の生成は，認証サーバにおいてスクリプトを作成しておくなどの工夫ををすることにより，大幅に操作手順を減らすことが可能である．

3.6 ルート認証サーバの処理負荷軽減

ASE は図 5 のような手順によりオンデマンドで証明書を収集するため，すべてのユーザが最初にルート認証サーバへ問い合わせを行う必要があり，ルート認証サーバへかかる処理負荷が多くなる懸念がある．この課題を解決するため，社員は相手の所属が明確で，かつ，ルート認証サーバの証明書を既に保持している場合は，ルート認証サーバを介さず，直接相手の認証サーバに問い合わせる．また，社員のノードはキャッシュを保持し，キャッシュ保持の期間内に同じ問い合わせが発生する場合，問い合わせを行わない．この方法により，検索時にかかるルート認証サーバの負荷を減少させることができる．この方法は DNS(Domain Name System) と同様の考えに基づく仕組みである．

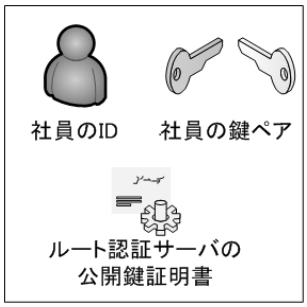

図 6 IC カードが持つべき情報

3.7 PKI との比較

PKI では，root CA の公開鍵証明書が自己署名のため，発行者が正当であることを検証できず，偽造されても検出ができないという課題がある．ASE，は検証者がルート認証サーバの公開鍵証明書を自ら検証できるため偽造が検出できる．

管理負荷としては，導入初期と運用時の 2 種類を考える必要がある．導入初期において，ASE は PKI に比べ各社員がルート認証サーバへ署名を行うなど作業負荷が増える．ただし，IC カードで運用することにより社員に与える負荷はなくなり，管理者の作業負荷もわずかの増加で済む．運用時においては，PKI は失効情報を確実に管理する必要があり管理コストが高くなる．これに対し ASE では，失効時に対象となる公開鍵証明書を削除するだけでよいため管理負荷が軽減される．

PKI の CRL モデルでは，失効情報が定期的に更新されるため，ユーザが最新の有効性を確認できない場合がある．ASE ではオンデマンドで認証パスを構築するためリアルタイム性に優れ，ユーザが最新の有効性を確認できる．

4. 実装と性能評価

4.1 モジュール構成

ASE を実現するため，社員端末用と認証サーバ用の端末に ASE の機能の一部を実装し，動作検証を行った．試作した ASE のモジュール構成を図 7 に示す．ASE が原理的に動作可能であることを示すため，今回は IC カード等の運用は考えず，社員ノード，各認証サーバがそれぞれ証明書を発行することとした．そのため，社員ノードおよび認証サーバには，それぞれ証明書発行モジュールを持たせた．証明書の発行処理は，OpenSSL[2] の関数を利用した．公開鍵証明書の形式は X.509 に準拠するものとする．社員ノードにおける情報取得モジュールは，被検証者の情報と認証

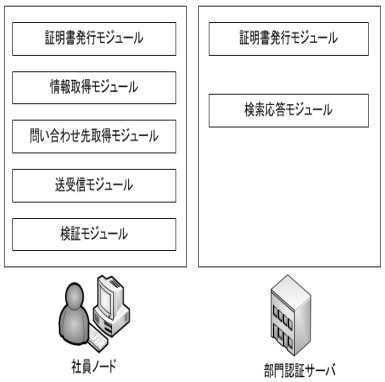

図 7 ASE のモジュール構成

サーバの階層数を取得する処理を行う．問い合わせ先取得モジュールは公開鍵証明書より問い合わせ先認証サーバの名前を取得する処理を行う．送受信モジュールは被検証者の情報を認証サーバ送り，公開鍵証明書を収集する処理を行う．検証モジュールは収集した公開鍵証明書を検証する処理を行う．送受信モジュールと問い合わせ先取得モジュールを複数回実行することにより，必要な公開鍵証明書を全て収集できる．認証サーバにおける検索応答モジュールは，社員からの問い合わせを受け，対応する公開鍵証明書が存在すればそれを送付し，存在しなければその旨を応答する．

4.2 検 証 処 理

検証モジュールは OpenSSL を用いた検証プログラム [3] をもとに作成した．この検証プログラムは，自己署名の公開鍵証明書と被検証者の公開鍵証明書を入力として，被検証者の公開鍵の正当性を確認できる．一階層分の検証処理を行えるが，そのままでは階層構造を実現できないため，被検証者に至るまでの検証が一度にできるように改造した．また，失効の確認など ASE では不要な部分を削除した．

しかし，入力として自己署名の公開鍵証明書が必要であることから，試作システムではルート認証サーバ，部門認証サーバにおいても自己署名の証明書を生成しておき，検証時にこれらの情報も全て収集することとした．図 8 に検証時の処理に置いて収集する情報を示す．

検証処理では，一階層ごとに自己署名の公開鍵証明書と検証したい公開鍵証明書を読み込み，検証を行う．検証に失敗した場合は，以降の階層の検証処理は行わない．しかし，この検証処理だけでは、階層間の検証ができていない．そこで，被検証者に至るまでの検証が成功すると，図 8 に示すように，上位ノードの公開鍵証明書内に記述されている公開鍵と，下位ノードが

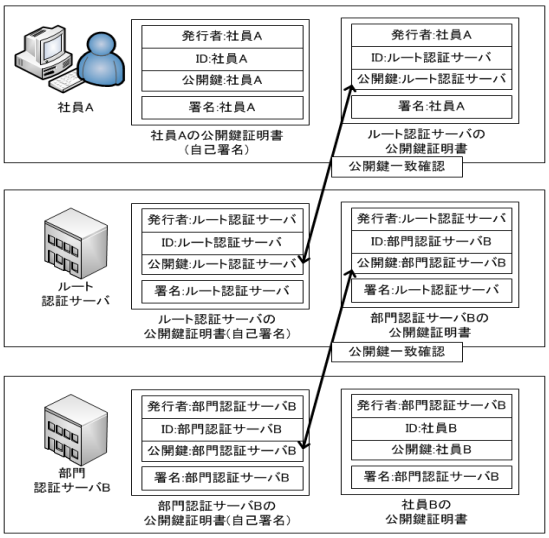

図 8 検証処理

保持する自己署名の公開鍵証明書内に記述されている公開鍵を比較して，一致することを確認する．公開鍵が一致した場合，認証パスの検証が問題なく終了したことになる．

5. ま と め

PKI では，root CA の公開鍵証明書の偽造を検証できないという課題がある．また，失効情報の管理が面倒であり，最新の情報が得られない場合がある．そこで信頼関係を環状にし，公開鍵証明書は発行者が保持して自ら管理を行い，信頼関係の検証をオンデマンドで行う認証システム ASE を提案した．PKI と比較すると，ASE はセキュリティ面，管理負荷の面で優れていると考えられる．今後は，公開鍵証明書の検証プログラムを作成し，実装と評価を行う．

参 考 文 献

1) 坂野文男，保母雅敏，渡邊晃：企業ネットワークにおける管理負荷の少ない認証システム ASE の提案，SCIS2006 シンポジウム論文集（2006）．
2) OpenSSL，”http://www.openssl.org/”
3) John Viega，Matt Messier，Pravir Chandra，齋藤 孝道 翻訳，”OpenSSL-暗号・PKI・SSL/TLS ライブラリの詳細-”，オーム社，2004 年