PKI の課題を解決する認証システム　ASE の提案

川島　隆太

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 は公開鍵証明機関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，証明書を root CA が自己署名し，これを検証者があらかじめ安全に保存していることが前提になっている．しかし，自己署名では発行者が本当に root CA であることを検証する方法がないため，これを偽造される可能性があり，安全とは言い切れない．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公開鍵証明書の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 an Enterprise network）[1]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て管理を行うため，失効情報の管理が不要である．

Researches on Authentication System for an Enterprise network ASE having high security

Ryuta Kawashima

PKI (Public Key Infrastructure) is widely used in the Internet these days. The public key of the user node is certified by hierarchzed 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 the root CA, 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

インターネットの普及に伴い, 電子商取引や電子申請等の電子化が期待されている. しかし, 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に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：Certification 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, 通常は root CA 自身が自己署名する. そのため, root CA の公開鍵証明書はユーザ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が可能であるという課題がある．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 の証明書を偽造できる可能性がある.

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は証明書が失効していないかどうかをその都度, 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 のセキュリティ上の課題を解決しつつ, 管理理負荷の少ない認証システムの一方式として ASE（Authentication System for an 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ため. 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て検討を行った.

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 が自分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．最上位の CA を root CA と呼ぶ. root CA の公開鍵は root CA 自身が自己署名する．root CA の公開鍵証明書は信頼点であり, 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. マイクロソフト社の Windows では，複数の root CA の公開鍵証明書があらかじめカーネルのレジストリに組み込まれて出荷されている.

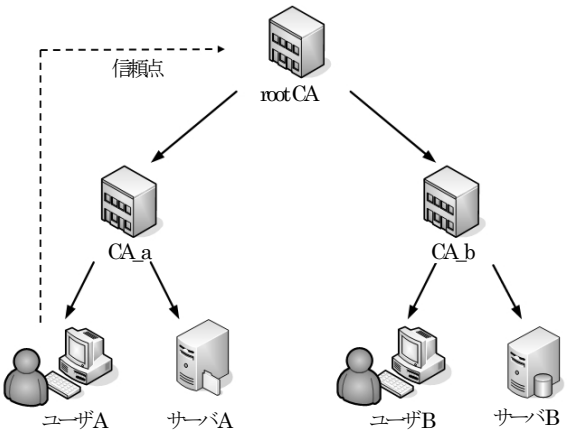

図 1　PKI の信頼関係

公開鍵証明書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．失効とは，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生する.

認証パスの検証の中で実行される失効の確認方法には CRL（Certificate Revocation List）[2]モデルと OCSP（Online Certificate Status Protocol）[3]モデルがある.

CRL モデルは各 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.

OCSP モデル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．OCSP モデルでは，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 の公開鍵を確認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 CA_b に公開鍵証明書を発行し，CA_b が

それを保持する．CA_b がユーザ B の公開鍵証明書を発行し，ユーザ B がそれを保持する．また，ユーザ A は root CA の自己署名による公開鍵証明書をあらかじめ保持している．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，root CA が自己署名した root CA の証明書がすべて含まれている．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 から失効情報 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．CRL は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．

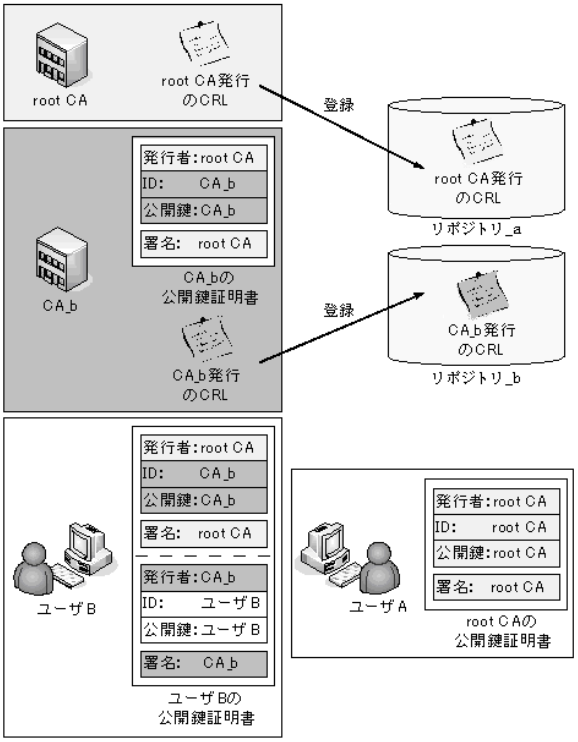

図 2　公開鍵証明書検証に必要なデータ

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．特に，レジストリから直接 root CA の公開鍵証明書を操作するとセキュリティ警告ウィンドウさえ表示されないため, 公開鍵証明書を偽造されたことに気づかない. また自己署名の公開鍵証明書を新たに作成して追加することは容易である. よって, ウィルスなど悪意あるプログラムなどにより同様の操作をされても, ユーザはそのことに気づかない可能性がある.

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，悪意ある第三者が権限を持つユーザになりすまし，当該ユーザへ問い合わせる．このユーザは相手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，このユーザは通信相手を信用して秘密データを悪意ある第三者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を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失効情報管理者の 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，失効情報の確認に CRL モデルを利用する場合は, 検証者が公開鍵証明書の検証をする前に CA から CRL を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 モデル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

本稿では PKI の課題を解決するための一方式として，ASE（Authentication System for an Enterprise network）を考案した．以下に ASE について説明する．現在，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て検討した結果を説明する.

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. 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. 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. ルートサーバは部門ごとに設置された認証サーバに公開鍵証明書を発行する. 認証サーバは各部門の社員や各サーバに公開鍵証明書を発行する. 各社員や各サーバはルート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, 公開鍵証明書の検証時に自分を最上位に位置づけることができる. 即ち, 公開鍵証明書の検証時において, 自分の持つ秘密鍵を信頼点とすることができ, 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.

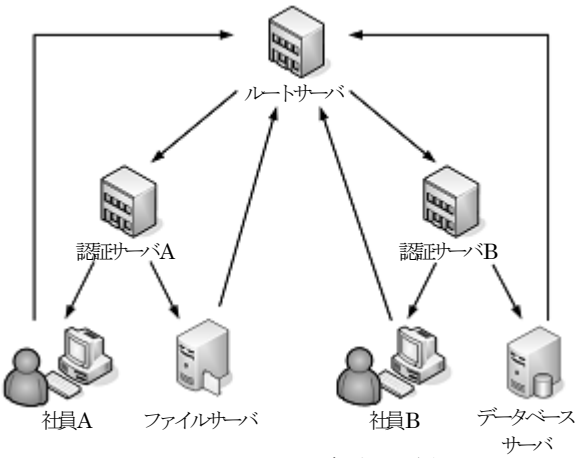

図 3　ASE の信頼関係

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 の公開鍵を確認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, 管理する. 即ち, 社員 B の公開鍵証明書を発行者の認証サーバ B が保持, 管理している. また, 認証サーバ B の公開鍵証明書は発行者のルートサーバが保持, 管理している. さらに, ルートサーバの公開鍵証明書は社員 A が保持, 管理している. 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.

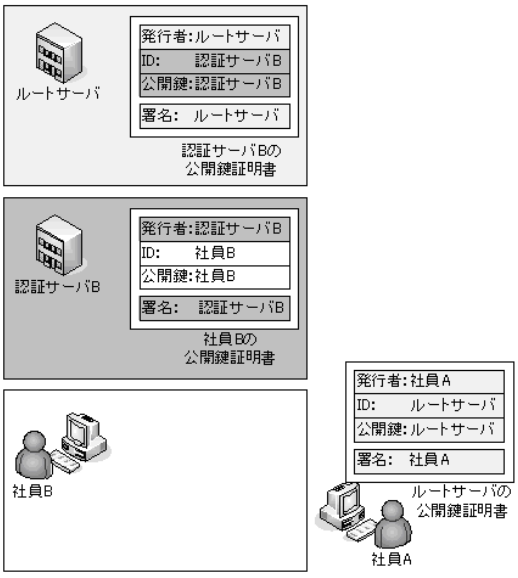

図 4　公開鍵証明書の管理方法

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．

社員 A はまずルートサーバへ社員 B の公開鍵証明書を問い合わせる．問い合わせに対しルートサーバは社員 B が所属している認証サーバ B の公開鍵証明書を返送する．次に，社員 A は認証サーバ B へ社員 B の公開鍵証明書を問い合わせる．問い合わせに対し認証サーバ B は社員 B の公開鍵証明書返送する．

上記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 の公開鍵でルートサーバの公開鍵を検証し，ルートサーバの公開鍵証明書で認証サーバ B の公開鍵証明書を検証し，さらに認証サーバ 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 の公開鍵で共通鍵を暗号化し，社員 B へ作成した暗号文を送る

社員 B は社員 B 自身の秘密鍵を利用して復号し，暗号化された共通鍵を取り出し，共通鍵を利用して社員 A へ共通鍵を取得したことを返答する

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．

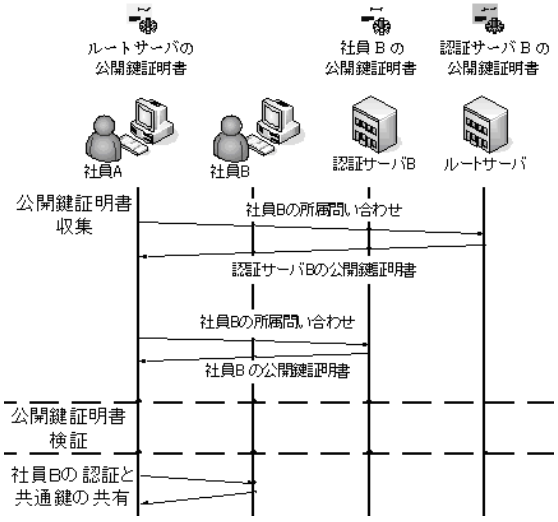

図 5　公開鍵証明書の有効性検証方法

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二階層とする．まず社員は自ノードで自分の鍵ペアを生成し，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認証サーバのところまで持っていく．認証サーバは受け取った証明書要求を自分の秘密鍵で署名し，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サーバの証明書要求に署名して，そのまま保持しておく．

このような方式は社員が署名することになるため，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鍵ペアとともに，社員の秘密鍵で署名したルートサーバの公開鍵証明書を同時に発行し，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．

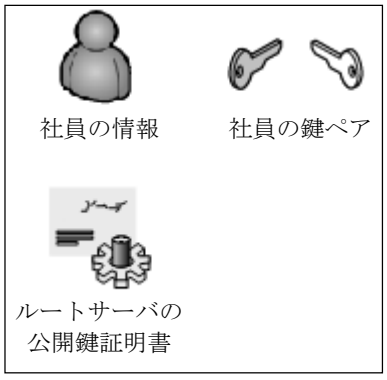

図 6　IC カードが持つべき情報

3.6 ルートサーバの負荷軽減

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サーバへ問い合わせを行う必要があり，ルートサーバへかかる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，かつルートサーバの証明書を既に保持している場合は，ルートサーバを介さず，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は問い合わせを行わない．この方法は DNS（Domain Name System）と同様の仕組みである．この方法により, 検索時にかかるルートサーバの負荷を減少させることができる.

3.7 PKI との比較

表 1 に PKI と ASE の比較を示す．最上位の公開鍵証明書について，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．ASE は検証者がルート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 に比べ各社員が署名を行う分負荷が増える．ただし，IC カードで運用することにより. その負荷は軽減させることができる．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 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.

リアルタイム性について, PKI 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．PKI OCSP モデルでは公開鍵証明書の有効性を検証時に問い合わせるため CRL モデルよりリアルタイム性に優れているが, 失効情報に CRL を利用している場合は，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.

表 1　PKI と ASE の比較

		PKI	ASE
最上位の公開鍵証明書		偽造検出不可能 ×	偽造検出可能 ○
管理負荷	導入初期	○	△
	通常運用時	△	○
リアルタイム性		△	○

４.実装と性能評価

4.1 モジュール構成

ASE を実現するために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た.

ASE のモジュール構成を図 7 に示す．社員ノードにおける証明書発行モジュールは, OpenSSL[4]を利用して公開鍵証明書を発行する. 公開鍵証明書の形式は X.509 に準拠するものとする．情報取得モジュールは，被検証者の情報と認証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公開鍵証明書を応答し，存在しなければその旨を応答する．IC カード発行モジュールは，IC カードに必要な情報を生成し，格納する処理である．本モジュールは今回の試作の範囲外とした.

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[5]をもとに作成した．この検証プログラムは, 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，有効性の確認など ASE では不要な部分を削除した.

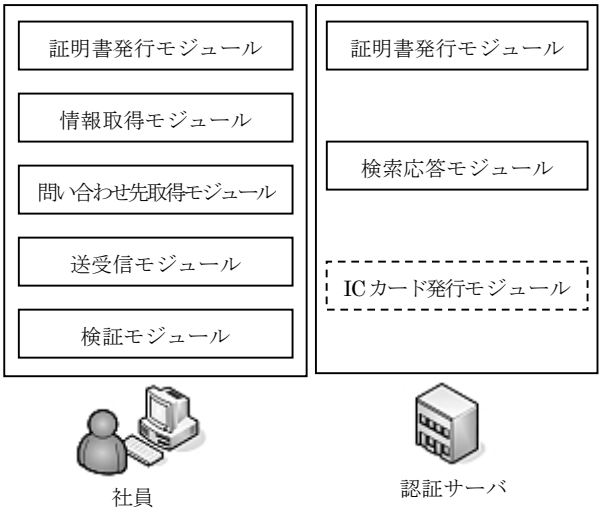

図 7　ASE のモジュール構成

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 のように, 上位ノードが保持する「目的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. この確認をすることで、階層間の検証も行われる. 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.

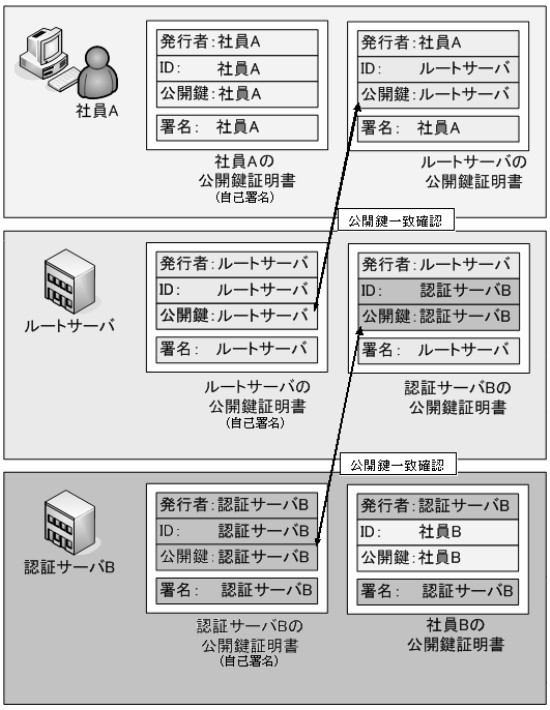

図 8　検証処理

4.3 性能測定

ASE の有効性検証にかかる時間を認証サーバが 2 階層分の場合において測定した. 表 2 に装置仕様を示す. 図 9 にルートサーバ, 認証サーバへの問い合わせから公開鍵証明書検証までの処理時間を示す. RSA の鍵サイズは 1024 ビットとした. 処理時間は 100 回試行した平均の値である. 送受信の際の伝送遅延は考慮していない. 処理時間は, 各プログラムの開始時と終了時に CPU time を取得し, その時間差を求めて測定した.

測定の結果, 社員側の問い合わせデータ作成処理に 0.03ms, 送受信処理に 1.83ms, 検証処理に 4.31ms の時間がかかった. サーバ側の検索応答処理には 0.59ms の時間がかかった. 以上から社員側ノードの処理時間合計 8.03ms であった.

処理時間に大きな影響を及ぼすのは検証時における公開鍵の演算回数である. PKI では, 公開鍵証明書自体の検証時に 2 回, CRL による失効情報の確認時に 2 回の公開鍵演算が行われる. ASE では公開鍵証明書の検証時に 3 回の演算が行われる. よって, 検証時間に関しては, ASE の方が有利である. しかし, 公開鍵が一致しているかの確認をする時間が ASE ではかかる. 全体的にみれば, ASE は PKI と同等の処理時間の性能だと考えられる. ただし, 検証処理を変更できれば ASE はより早い性能を得ることができると考えられる.

表 2　装置仕様

	社員	サーバ
PC Model	Prime Series	OptiPlex GX280
CPU	Core2 Quad Q9550(2.83GHz)	Pentium4(3.40GHz)
メモリ	4096MB	1024MB
ネットワーク	100BASE-T	
OS	Ubuntu 8.10	
暗号化機能	openssl 0.9.8j	

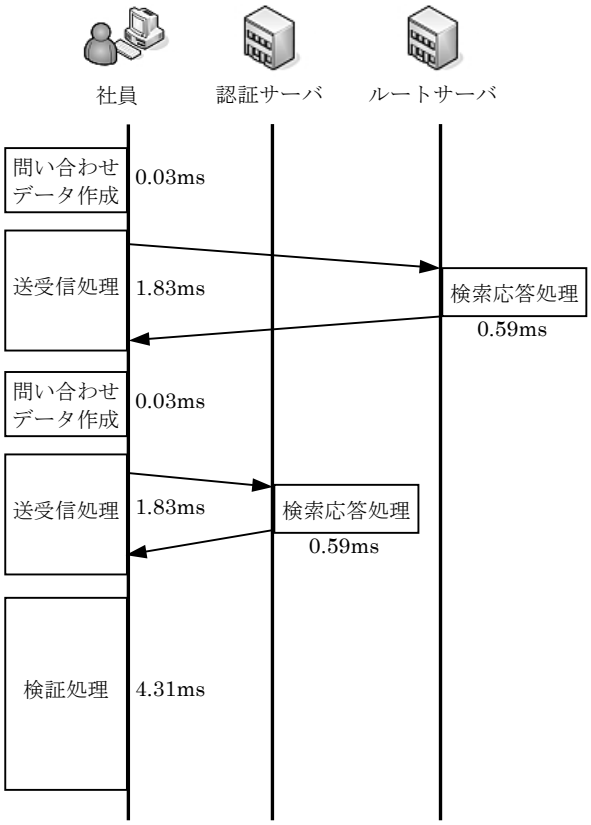

図 9　処理時間

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. 試作して動作検証した結果, 性能的にも問題ないことがわかった. ASE の企業ネットワークのように閉じた世界への導入は十分可能と考えられる.

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] R. Housley, W. Polk, W. Ford, D. ｂSolo, “Internet X.509 Public Key Infrastructure Certificate and CRL Profile”, RFC 3280, April 2002.

[3] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP”, RFC 2560, June 1999.

[4] OpenSSL “http://www.openssl.org/”

[5] John Viega, Matt Messier, Pravir Chandra 共著，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 8 月

謝辞

本研究を行うに当たり，多大なるご指導，ご鞭撻を賜りました渡邊晃教授に心より感謝いたします. また，有益な助言および検討を頂きました渡邊研究室の皆様に深く感謝いたします.