コンピュータセキュリティ 10−29
(2000. 7. 25)

SPKIによるプライバシ保護手法とその株主優待券の電子的実装への応用

梅澤 健太郎† 齋藤 孝道‡ 奥乃 博‡

†東京理科大学大学院理工学研究科情報科学専攻 ‡東京理科大学理工学部情報科学科

〒274-8510 千葉県野田市山崎２４６１

kentaro@cs.is.noda.sut.ac.jp,{saito,okuno}@is.noda.sut.ac.jp

あらまし

SPKI (Simple Public Key Infrastructure) は, 公開鍵暗号を用いたインフラストラクチャ構築のための技術である. SPKIでは認証と権限管理を別に扱うことができるので, 単純にX.509証明書を用いた場合よりもプライバシーを重視し, サービスが提供できると期待される. 本論文では, SPKI権限証明書を応用した権限管理の実装例として, 株主優待券を取り上げ, その設計, 実装を行ったので報告する. またその過程において, ユーザが自身のIDを秘匿したまま, 相手に年齢などの属性情報を示すための方式の枠組みを考案し実装したので報告する

キーワード プライバシ, SPKI, 権限証明書, アクセスコントロール

Proposal of privacy protection scheme by SPKI and its application to electronic implementation of complimentary tickets for stockholders

Kentaro UMESAWA† Takamichi SAITO† Hiroshi G.OKUNO†

†Dept. of Information Sciences , Science University of Tokyo

2461 Yamazaki, Noda, Chiba 274-8510

kentaro@cs.is.noda.sut.ac.jp, {saito,okuno}@is.noda.sut.ac.jp

Abstract

SPKI is a technology for the infrastructure construction which uses the public key. Because the authority management can be treated separately to the authentification in SPKI, it is expected that service by which privacy is valued can be offered when the X.509 certificate is used. In this paper,we present the design and mounting of the stockholder preferential treatment ticket as an example of mounting the authority management to apply the SPKI authority certificate. Moreover, we present the frame of the method of show the other party attribute information on the age etc. with the user hided own ID secretly.

key words Privacy, SPKI, Authorization Certificate, Access Control

1. プライバシ重視のネットワークサービス

1.1 概　　要

公開鍵暗号を用いた安全な通信のためのはインフラストラクチャの構築が不可欠である．それは PKI (Public Key Infrastructure) と呼ばれ，その枠組みの一つとして，PKIX (PKI with X.509) が提案され，複数の企業や機関でその実装が進められている．PKIX は ID 証明書を発行する母体として認証局 (CA, Certificate Authority) を使用し，証明書としては X.509 を用いる．PKIX は，元々は認証の枠組みとして提案されたが，次のようにその機能を分けて考えると，権限管理にも利用できる．

1) **認証**：公開鍵と ID の結びつきを保証する ID 証明書を用いて，本人であることを示す．
2) **権限管理**：属性証明書 (一般的にはサーバの ACL) を確認し，ID の属性である権限を決定する．

PKIX では，ID と公開鍵，及び，ID と権限の結びつきが前提となっているので，認証 (authentication) と権限 (authority, access rights) が一体化している．つまり，権限と公開鍵とを結び付けるのに ID が必要なため，グローバルな ID を保証する X.509 証明書により権限管理を行う場合，匿名サービスといったプライバシを重視したネットワークサービスの実現は難しい．

我々は，ID を介さず権限と公開鍵が直接結びついている SPKI (Simple Public Key Infrastructure) の権限証明書 (Authority Certificate) を利用することによって，SPKI の特徴であるシンプルな権限管理手法に，プライバシ重視の権限管理，つまり，認証と権限が分離した権限管理の枠組みを追加し，提案してきた [1]．本論文では，『プライバシ重視』を定義を与え，プライバシ重視の枠組みを提案する．さらに，プライバシ重視のネットワークサービスの一例として，単に匿名性を保証するだけでなく，ユーザが名前情報を秘匿したままで，性別，年齢などの情報を相手に対して提示する方法も考案し実装したことを報告する．

1.2 SPKI(SimplePublicKeyInfrastructure)

SPKI は，Carl Ellison の論文 [3] を切っ掛けに始まり，現在，RFC2962, 2963 で規定されている [4),5)]．RFC の中では，名前空間に関する記述が多いが，我々のアイデアは，SPKI 権限証明書が ID を含まないという事実に着目し，シンプルで効率的でかつ，プライバシを重視した権限管理の実現を目指したことにある．SPKI において権限証明書と呼ばれる証明書が用いられる．これは，公開鍵と権限の対応に，権限証明書の発行者がディジタル署名を付加したものである．権限証明書は，公開鍵と権限の結び付きを発行者に対する信頼のもとで保証するものであり，それ自身に ID 情報を含んでいない．具体的な SPKI 権限証明書の様式は，以下のような 5-tuple に発行者のディジタル署名を付加したものとなっている．

$$(I, S, D, A, V)$$

- I：Issuer. 権限証明書の発行者の公開鍵．
- S：Subject. 権限を行使する主体の公開鍵．
- D：Delegation. bool 値．S が更に権限を委譲することが可能かどうかを示している．
- A：Authorization. 権限を表現している．
- V：Validity. 証明書の有効期限．

なお，一般に，権限の生存時間は短かく，種々の権限が存在するが，それらに対応した使い捨ての公開鍵を利用できる．本論文の株主優待券の実装では，SPKI 権限証明書の形式に変更を加えたものを利用する．

2. プライバシの定義

プライバシとは，19 世紀のアメリカの学者によって提案された法概念である．これを「伝統的なプライバシ」と呼ぶ，その意味を「プライベートな情報」と定義する．しかし，ネットワークにおいて伝統的なプライバシの概念を適用しようとすると，電子メールアドレスは公の情報かプライベートな情報なのかといった混乱が生じる．よって，ネットワーク社会におけるプライバシを，伝統的な意味でのプライバシと区別し，「ネットワーク・プライバシ」と呼ぶ．本稿におけるプライバシとはこのネットワーク・プライバシを指す．

では，ネットワーク・プライバシとはどのように定義されるべきものなのだろうか? 憲法学者アラン・F・ウェストは，著書『プライバシーと自由』の中で，「プライバシーとは，個人，グループまたは組織が，自己に関する情報を，いつ，どのように，またどの程度に他人に伝えるのかを自ら決定できる権利である」と定義した．ここから，プライバシの権利とは「自己にかかわる情報について一定のコントロールを及ぼす権利」(**自己情報コントロール権**) であるという考え方が提案されている [2)]．我々は，ネットワーク・プライバシを自己情報コントロール権と定義し，本稿において「自己情報コントロール権を保証する枠組み」として，プライバシ重視の権限管理を提案する．

3. プライバシ重視の権限管理の実装とその考察

3.1 プライバシ重視の権限管理の枠組

プライバシ重視の権限管理とは他のサービスとどう

異なるものなのか？例をあげて考えてみる．SSL に代表される通信路上でのデータ暗号化は，第三者に対して通信路を守るものである．これは，サーバに対してユーザが提供するデータを秘匿するものではない．また PKIX によるアクセス制御を考える．この場合，クライアントが ID を認証データとして提供する必要があるが，ID は本来，権限管理には不必要な情報である．不必要な自己にかかわる情報を，他の主体に対して与えることは望ましいことではない．

我々は，上記の議論と「自己情報コントロール権」という定義をもとに，「プライバシ重視の権限管理」の枠組み [1)6)] を検討した．しかし，ID を秘匿することだけでプライバシが守れるわけではない．自己情報コントロール権という立場からは，自身の情報の一部を明かしつつ，他の情報を秘匿したい場合についても考慮しなければならない．また，現実的には，サービスを提供するサーバが，サービスを受けるクライアントの年齢・性別などの情報に基づいてアクセスコントロールを行いたい場合もあるだろう．つまり，**年齢，性別といった属性情報がある種の権限と結びついている**ことになる．このような機能を実現するためにプライバシ保護サーバを提案する．これは，事前にクライアントの ID と，年齢，性別などの属性情報の登録をうけ，クライアントからの要求時にクライアントの認証を行い，属性と公開鍵の対応を保証する証明書 (以下**特殊権限証明書**と呼ぶ) を発行するものである．プライバシ保護サーバが，権限証明書発行サーバと等価である場合もあるが，特殊権限証明書をより汎用的に利用できるものとするため，概念的に別のものとして扱う．

3.2 登場主体

プライバシ重視という目的を達成するために必要な具体的な構成要素は以下のようになる

(1) **サーバ $\mathcal{S}$** ：クライアントに対してサービスを提供する主体．

(2) **クライアント $\mathcal{C}$** ：サーバに対してサービスを要求する主体で，自己の ID 情報を権限証明書発行サーバ $\mathcal{I}$ に登録してある主体．また自己の ID 情報と，その他の属性についての関係をプライバシ保護サーバ $\mathcal{P}$ に登録してある．

(3) **権限証明書発行サーバ $\mathcal{I}$** ：$\mathcal{S}$ からの委託の下に，ユーザに対して権限証明書を発行する．ACL を保持しており，そこにはユーザの ID 情報と鍵及び権限の対応が記されている．適切なクライアントからの要求を受けて，権限証明書の発行を行う．

(4) **プライバシ保護サーバ $\mathcal{P}$** ：ユーザからの委託の下に，ユーザに対して特殊権限証明書を発行する．ユーザの ID 情報を保持しており，さらにはその他の属性 (年齢，性別など) の情報も保持している．適切なクライアントからの要求を受けて，要求された属性を保証する特殊権限証明書の発行を行う．

クライアントは $\mathcal{I}$ から権限証明書 $Cert$ の発行を受け，$\mathcal{S}$ に対してそれを提示し，権限を保持していることを示す．またクライアントは，$\mathcal{S}$ から属性情報に付いての要求があった場合には，$\mathcal{P}$ から特殊権限証明書 $ACert$ の発行を受け，$\mathcal{S}$ にそれを提示する．$\mathcal{S}$ は，(自身が提出を要求した場合) 特殊権限証明書 $ACert$ を検証することで $\mathcal{C}$ の属性情報を確認し，さらに $\mathcal{I}$ から発行された権限証明書 $Cert$ の検証を行い，権限情報を確認する．$\mathcal{C}$ について属性情報 (自身が提出を要求した場合のみ)，権限情報の検証が成功したら，$\mathcal{S}$ は $\mathcal{C}$ に権限に対応したサービスを返す．$\mathcal{S}$ は属性情報に関して問題が発生した場合は $\mathcal{P}$ に，権限情報について問題が発生した場合は $\mathcal{I}$ に問い合わせを行う．

3.3 処理の流れ

具体的な処理手順は，以下のような流れになる．以下での各ステップは，図 1 と対応している．

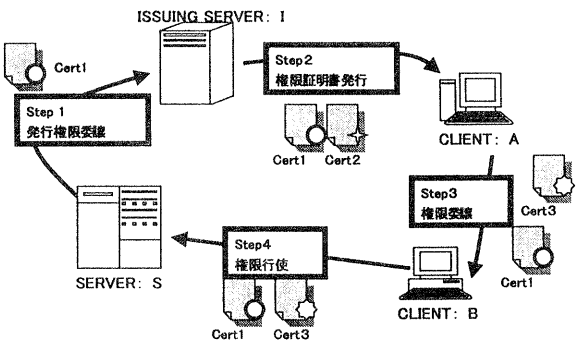

図 1 権限証明書の発行から利用までの処理の概要図

Step1 発行権限委譲： SERVER $\mathcal{S}$ は，ISSUING SERVER $\mathcal{I}$ に自身のリソースに対するアクセスについての権限証明書を発行する権利を与える．$\mathcal{S}$ は，その権限をあらわす証明書 $Cert_1$ を $\mathcal{I}$ に送信する．この結果，$\mathcal{S}$ の保持する ACL には，「$\mathcal{I}$ に対して権限証明書発行権限を与えた」というエントリが追加される．

Step2 権限証明書発行： $\mathcal{I}$ は，CLIENT $\mathcal{A}$ の要求に応じて，権限証明書を発行する．$\mathcal{I}$ は，ACL を保持し，権限証明書を発行する相手の ID と権限の対応付

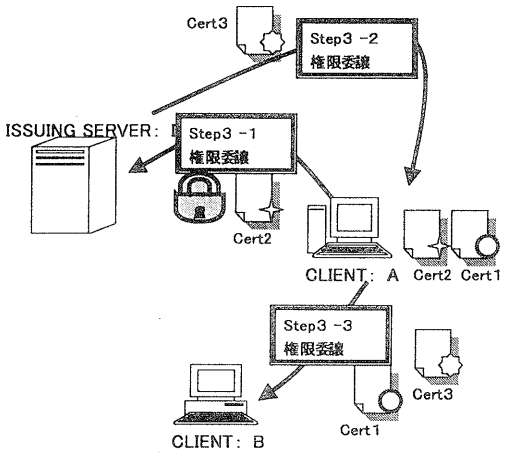

図 2　権限委譲の詳細図

けをエントリとして持つ. また, 権限証明書を発行する相手の ID と公開鍵の対応も保持している. まず, $\mathcal{A}$ の認証を行い $\mathcal{A}$ の ID を確定する. 次に $\mathcal{I}$ の保持する ACL を参照し, ID の属性となる権限を確定する. $\mathcal{A}$ の要求する権限が, ACL の許すものであれば, $\mathcal{A}$ に対して権限証明書 $Cert_2$ を発行し, 認証によって確保したセキュアチャネル上で, $Cert_1, Cert_2$ を $\mathcal{A}$ に対して送信する. 具体的な手順は以下のようになる：

（1） ユーザ $\mathcal{A}$ は, $\mathcal{I}$ に自分の ID を渡す.

（2） $\mathcal{I}$ は, $\mathcal{A}$ の ID に対応する「権限」と「公開鍵」に, 自分の公開鍵, 証明書の有効期限, 及び権限委譲の可否を示す情報を付加して, 権限証明書 $Cert_2$ を作成する.

（3） $\mathcal{I}$ は, デジタル署名を $Cert_2$ に付け, $\mathcal{A}$ に発行する.

ここで, $Cert_2$ を受け取った $\mathcal{A}$ は, $Cert_1$ と $Cert_2$ を $\mathcal{S}$ に提示することで $Cert_2$ の示す権限を行使できる. また $\mathcal{A}$ は, $Cert_2$ の権限委譲項目が可の場合, $Cert_2$ を基にして, $Cert_2$ にある権限の範囲内で別の権限証明書を発行することもできる. ここで注意すべきことは, $\mathcal{I}$ が権限委譲に際しても役割を持つことである.

Step3 権限委譲 : $\mathcal{A}$ は, $Cert_2$ の Delegation が True の場合に, 自分のもつ権限 (またはその一部) を委譲することができる. この部分の詳細は図 2 で示す. 以下の説明は図 2 と対応している.

Step3-1 : $\mathcal{A}$ は, 委譲者となり, $Cert_2$ を基にして, 権限を委譲する相手 $\mathcal{B}$ の公開鍵 $P(\mathcal{B})$, $\mathcal{B}$ に委譲する権限, 有効期限などの情報と自分自身が発行された権限証明書 $Cert_2$ を権限証明書発行サーバ $\mathcal{I}$ に送信する.

Step3-2 : 権限証明書発行サーバ $\mathcal{I}$ は, 権限証明書 $Cert_3$ を作成し, それに対して自分の秘密鍵 $S(\mathcal{I})$ で署名を施す. さらに, 権限証明書 $Cert_2$ を破棄し, $Cert_2$ が無効になった旨を $\mathcal{S}$ に通知する. また, 自身のファイルに, $\mathcal{A}$ の公開鍵 $P(\mathcal{A})$ から $\mathcal{B}$ の公開鍵 $P(\mathcal{B})$ へ権限が移ったことを記録する. そして $\mathcal{I}$ は $\mathcal{A}$ に $Cert_3$ を渡す. ここで, 公開鍵 $P(\mathcal{B})$ は, $\mathcal{I}$ にとって, $\mathcal{B}$ の ID 情報をあらわすものではないことに注意する. $\mathcal{I}$ が知りえる情報は, $\mathcal{A}$ が, $P(\mathcal{B})$ を持つ人物に権限を委譲したということのみである.

Step3-3 : ここで, $Cert_3$ を受け取った $\mathcal{A}$ は, $Cert_1$ と $Cert_2$ を $\mathcal{B}$ に送信する. $\mathcal{B}$ は $\mathcal{A}$ と同様, $Cert_1$, $Cert_3$ を $\mathcal{S}$ に提示することで $Cert_3$ の示す権限を行使できる. また $\mathcal{B}$ は, $Cert_3$ の権限委譲項目が可の場合, $Cert_3$ で指定された権限の範囲内で別の権限証明書を発行することもできる. その場合は, Step3-1 に戻る. この際, 権限証明書発行サーバ $\mathcal{I}$ は, 提出された権限証明書に含まれる公開鍵と, 委譲の際に公開鍵の関係を記したファイルを照らし合わせて, 正当に委譲された主体であることが確認された場合に新しい権限証明書を発行する. この際も, 提出された権限証明書は破棄され, 委譲の際の公開鍵の関係が記録される.

Step4 権限行使 : ユーザ (図 1 での $\mathcal{A}$ または $\mathcal{B}$ は, 権限証明書を $\mathcal{S}$ に対して提示することで自分の保持する権限を行使できる. ここで, $\mathcal{USER}$ に対するサービスの提供は, $\mathcal{USER}$ が提示した権限証明書に含まれるユーザ自身の公開鍵 $P(\mathcal{USER})$ を用いて, 暗号化できる. これを復号できるのは, $P(\mathcal{USER})$ に対応する秘密鍵 $S(\mathcal{USER})$ を保持する主体のみであることに注意する. この部分の詳細は図 3 で示す. 以下の説明は図 3 と対応している.

Step4-0 : ユーザ $\mathcal{USER}$ は, あらかじめプライバシ保護サーバ $\mathcal{P}$ に, 自分自身の ID 情報と, 属性情報 (年齢, 性別,..*etc*) を登録する. さらに, それらの属性情報と結びつける公開鍵 $P(\mathcal{USER})$ を登録しておく.

Step4-1 : $\mathcal{P}$ は, $\mathcal{USER}$ からのリクエストに応じて特殊権限証明書 $ACert$ を発行する. 具体的な手順は以下のようになる.

（1） $\mathcal{P}$ は, $\mathcal{USER}$ を認証する.

（2） $\mathcal{P}$ は, $\mathcal{USER}$ が, どんな情報に関する特殊権限

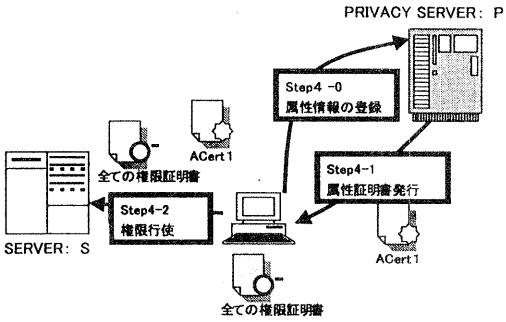

図 3　権限行使の詳細図

証明書を必要としているのかを確認する (ここでは, $\mathcal{USER}$ は年齢に関する特殊権限証明書 $ACert$ を欲しているとする) . この場合, $\mathcal{P}$ は以下のような情報に, 自身の秘密鍵 $S(\mathcal{P})$ を用いて署名する. これが $ACert$ となる.

($P(\mathcal{P})$, $P(\mathcal{USER})$, 年齢情報, 有効期限)

ここでの有効期限は, 証明書自体の有効期限をさしている.

最後に, $\mathcal{P}$ は, $ACert$ を $\mathcal{USER}$ に送信する. この通信は, (1) の認証において成立するセキュアチャネルを用いるため. 通信路中で, 特殊権限証明書 $ACert$ の機密性・完全性は保護される. また $ACert$ は, $\mathcal{P}$ により署名されているので改竄されることはない.

Step4-2 : ユーザは, 自分が持つ権限証明書のうち, 行使する権限に関連したすべての証明書及び (必要な場合は) 特殊権限証明書 $ACert$ を, $\mathcal{S}$ に渡す. まず, $\mathcal{S}$ は, ユーザが, 提出された権限証明書および特殊権限証明書に含まれる公開鍵に対応する秘密鍵を所持していることを, チャレンジなどにより確認する. その後, $\mathcal{S}$ は, 与えられた証明書の正当性を検証する. 具体的な権限証明書の検証の方法は, 論文 [6] を参照してもらうとして, ここでは $ACert$ の検証方法を述べる. $ACert$ の正当性は, $\mathcal{S}$ が $P(\mathcal{P})$ を用いて $ACert$ の署名を検証することで示すことができる.

以上の処理から, 証明書が正当と判断された場合, $\mathcal{S}$ はユーザに対して権限に応じたサービスを提供する.

3.4 実　　装

上述した枠組みを株主優待券の実装に適用した際の詳細について述べる.

3.4.1 導　　入

今回実装したシステムは, 株主優待券の発行及び使用を扱うものである. 株主優待券とは, ある会社の株式を保持する人物に対して発行されるもので, その券の持ち主に対し会社側から何らかのサービスを提供するものである. 通常の株主優待券は, 入場券, 割引券などの役割を持つ, さらに今回の実装のように特殊権限証明書と組み合わせることで, 属性情報の必要なサービス (酒・たばこのネット販売, 男性または女性限定サイトへの入場) の実現も可能になる. ここで, 株主優待券はチケットであるので匿名性を持ち, 委譲可能であること. 及び, 株主優待券として発行された権限はその会社に戻ることに注意する. 優待券は会社から発行されるのではなく, その会社の株式の手続きを代行する証券会社が発行する. これにより, 会社が証券会社に対して発行権限を与えているという解釈が成り立つ. 以上のことは, プライバシ重視の権限管理への要求事項と一致する.

3.4.2 前 提 事 項

ここでは, 前述の枠組みに対して以下のような現実的意味を与える.

- **サーバ $\mathcal{S}$:** ある株式会社 $\mathcal{COM}$ のリソースである. $\mathcal{COM}$ は, $\mathcal{S}$ において自社の株主優待券を保持するものに対してサービスを提供する. この際, 誰が株主優待券を用いたのかということには関心がなく, 株主優待券の偽造などによる損害を防ぐことができればよい. 株主優待券の発行管理は, 株式の販売を代行している証券会社 $\mathcal{FIRM}$ に委託する. この結果, $\mathcal{S}$ は, 株主優待券発行サーバ $\mathcal{I}$ の公開鍵 $P(\mathcal{I})$ と, 株主優待券発行サーバに対して与える権限の対応が記されることになる. ユーザが株主優待券を提出してきたときに, このファイルを参照して, $\mathcal{I}$ の発行した株主優待券が正しいものかを判断する.
- **株主優待券発行サーバ $\mathcal{I}$:** 証券会社 $\mathcal{FIRM}$ に属する. $\mathcal{FIRM}$ は顧客 $\mathcal{C}$ の ID 情報を保持しており, また, 各顧客が株取引に際して用いている ID に関連付けられた公開鍵 $P(\mathcal{C})$ を保持する. さらに, 株主優待券の発行に関して $\mathcal{S}$ から権限を与えられており, 保有株式数に応じて各顧客に対して割り当てられる株主優待券 (権限証明書) の情報をもつ. ここで注意することは, 顧客の実際の ID 情報と $P(\mathcal{C})$ の対応は, $\mathcal{S}$ の知るところではないことである. また $\mathcal{I}$ は, 株主優待券の委譲に際しても役割を持つ.
- **顧客 $\mathcal{C}$:** 証券会社 $\mathcal{FIRM}$ を通じて, 株式会社 $\mathcal{COM}$ の株式を保有する主体である. $\mathcal{FIRM}$ の公開鍵 $P(\mathcal{FIRM})$ は事前に保持してある. また潜在的には, 株主優待券を違う人物に委譲する可能性もあるので, 株主優待券使用者のほかに, 株主優待券委譲者ともなりうる.

- **株主優待券委譲者** $\mathcal{DELEGATER}$ **：** 別の主体 $\mathcal{O}$ に対して, 株主優待券を委譲する主体. $\mathcal{O}$ の公開鍵 $P(\mathcal{O})$ を所持している.
- **株主優待券使用者** $\mathcal{USER}$ **：** 顧客 $\mathcal{C}$ または, $\mathcal{C}$ から株主優待券の委譲を受けた主体である.
- **プライバシ保護サーバ** $\mathcal{P}$ **：** 株主優待券使用者 $\mathcal{USER}$ からの委託の下に, $\mathcal{USER}$ に対して権限証明書を発行する. $\mathcal{USER}$ の ID 情報を保持しており, さらにはその他の属性 (年齢, 性別など) の情報も保持している. $\mathcal{USER}$ が登録した公開鍵 $P(\mathcal{USER})$ と各種属性の関連を保証する証明書 (以下**特殊権限証明書**) の発行を行う.

さらに, 以下のような前提を置く

- $\mathcal{FIRM}$ は, $\mathcal{COM}$ から株主優待券の発行権限をあらわす権限証明書 $Cert_1$ をすでに受け取っている.

以上の前提より, $\mathcal{COM}$ の保有する ACL には, 株主優待券使用者のエントリが存在しない. よって株主優待券の使用において $\mathcal{COM}$ に対する, $\mathcal{USER}$ の ID 流出はないことに注意する.

3.5 実装の処理手順

属性情報の登録： 株主優待券使用者 $\mathcal{USER}$ は, あらかじめプライバシ保護サーバ $\mathcal{P}$ に, 自分自身の ID 情報と, 属性情報 (年齢, 性別,..*etc*) を登録する. さらに, それらの属性情報と結びつける公開鍵 $P(\mathcal{USER})$ を登録しておく.

発行権限委譲： $\mathcal{COM}$ は, $\mathcal{FIRM}$ に自身のリソースに対する株主優待券を発行する権利を与える. 今回の実装においてこのステップは, すでに行われているものとする. この結果, $\mathcal{COM}$ の保持する ACL には, 「$\mathcal{FIRM}$ に対して, 株主優待券の発行権限を与えた」というエントリが追加されることになる.

株主優待券 (権限証明書) 発行： $\mathcal{FIRM}$ は, 顧客 $\mathcal{C}$ の要求に応じて, 株主優待券を発行する. $\mathcal{FIRM}$ は, $\mathcal{C}$ の株取引から $\mathcal{C}$ の保有する株式を把握しており, また $\mathcal{C}$ が取引において用いている公開鍵 $P(C)$ も所有している. よって, $\mathcal{C}$ に対して発行できる株主優待券についての情報と ID の対応付けを ACL エントリとして持つ. また, 株主優待券を発行する相手の ID と公開鍵の対応も保持する. 具体的な発行手順は以下のようになる：

(1) $\mathcal{C}$ は, $\mathcal{FIRM}$ に自分の ID を提示し, 認証を行う.

(2) $\mathcal{FIRM}$ は, $\mathcal{C}$ が保有する株式をもとに, $\mathcal{C}$ に発行できる株主優待券を確認し, 株主優待券 $Cert_2$ を発行する. ここには, $\mathcal{C}$ の公開鍵 $P(\mathcal{C})$ と, 自分の公開鍵 $P(\mathcal{FIRM})$, 株主優待券の権限, 株主優待券の有効期限, 及び株主優待券の委譲の可否を示す情報 (Bool 値), さらに $\mathcal{FIRM}$ が管理するシリアル番号が含まれる.

(3) $\mathcal{FIRM}$ は, 自身の秘密鍵 $S(\mathcal{FIRM})$ を用いてデジタル署名を $Cert_2$ に付加し, $Cert_1$ のコピーとともに $\mathcal{C}$ に発行する.

株主優待券委譲： $\mathcal{C}$ は, $Cert_2$ の Delegation が True の場合に, 自分のもつ株主優待の権利 (またはその一部) を委譲することができる. これは, 株主優待の権利を委譲する権限証明書を発行してもらうことで行われる. 具体的な手順は, Step3 で述べた枠組みをそのまま適用する. 唯一異なる点は, 以前の証明書を破棄した $\mathcal{I}$ は, その破棄された株主優待券のシリアル番号を, $\mathcal{S}$ に通知する点である. これは二重使用を防ぐためである.

株主優待券の利用： 株主優待の権利保持者は, 株主優待券に相当する権限証明書の組を $\mathcal{COM}$ に対して提示することで自分の保持する権限を行使できる. また株主優待券の使用に関して, 20 歳以上であることなどの属性情報の提出を求められる場合は, 特殊権限証明書もあわせて提出する.

(1) (株主優待券の利用にあたって, 年齢制限などがあり, 特殊権限証明書 $ACert$ の提出が求められる場合のみ) プライバシ保護サーバにアクセスし, $ACert$ の発行を受ける.

(2) 株主優待券利用者 $\mathcal{USER}$ は, 株主優待券および特殊権限証明書 $ACert$ (提出が求められる場合のみ) を, $\mathcal{COM}$ に渡す.

(3) $\mathcal{COM}$ は, 与えられた権限証明書の正当性を検証する. まず, (提出を求める場合のみ) 特殊権限証明書 $ACert$ を検証する. 次に株主優待券の検証を行い, 正当と判断された場合, $\mathcal{COM}$ は株主優待券利用者に対して優待券に記載された権限に応じたサービスを提供する.

3.6 株主優待券, 特殊権限証明書の形式

株主優待券

$$(I, S, D, A, V, N)$$

- I : Issuer. 株主優待券発行者の公開鍵.
- S : Subject. 株主優待券を行使する主体の公開鍵.

- D : Delegation. bool 値. S が株主優待券を委譲することが可能かどうかを示している.
- A : Authorization. 権限を表現している.
- V : Validity. 株主優待券の有効期限.
- N : Number. シリアル番号. 株主優待券の委譲・失効管理で使用.

特殊権限証明書

($P(\mathcal{P})$, $P(\mathcal{USER})$, 属性情報, 有効期限)

- $P(\mathcal{P})$: プライバシ保護サーバの公開鍵
- $P(\mathcal{USER})$: ユーザの公開鍵
- 属性情報 : 証明される属性情報の文字列
- 有効期限 : yyyymmddhhmmss の形式の文字列. 開始と終了二つが記載される

3.7 処理の詳細

実装は, **JDK1.2, JSDK2.0, ApacheJServ1.1, IAIK2.51** によって作成した. $\mathcal{COM}$, $\mathcal{FIRM}$ のインターフェースは, Web で提供されると考え, Servlet ととして実装した. 株主優待券の発行, 検証のロジック及び, 株主優待券使用者が復号, 委譲などに用いるツールは, 以前我々が作成したライブラリ [1] の一部を修正し利用した. 具体的には, SPKI 権限証明書の検証・生成を行うためのライブラリ VERIFY, ISSUE を拡張し, 株主優待券および特殊権限証明書の形式も扱えるようした. さらに, 権限証明書の委譲を行うためのアプリケーションであった DELEGATER を, 共通ライブラリ化して Servlet からも利用できるようにした. また, 株主優待券発行に際しての $\mathcal{FIRM}$ と $\mathcal{C}$ のセキュアチャネルの確保及び特殊権限証明書発行に際しての $\mathcal{P}$ と $\mathcal{C}$ のセキュアチャネルの確保には SSL 化した Web サーバと, CA により発行された個人証明書を組み込んだブラウザを使用することを前提としている.

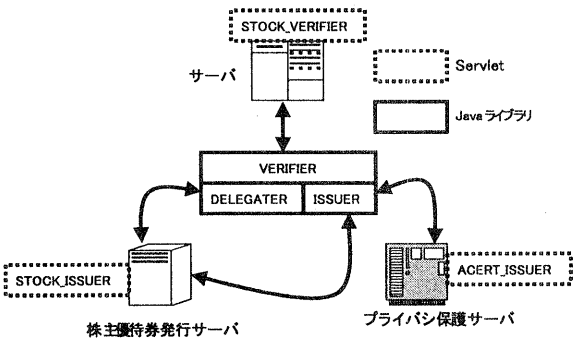

図 4 実装における Java クラスの相関図

4. 考　　察

4.1 安全性に対する考察

まず, 安全性に関して考慮すべき点を, **「通信路中での安全性」**と**「株主優待券にかかわる安全性」**さらに**「特殊権限証明書にかかわる安全性」**について述べる. まず, **「通信路中での安全性」**について議論する. 以下の二つの点に関して考察する.

- **証明書発行サーバ (株主優待券発行サーバ $\mathcal{I}$, プライバシ保護サーバ $\mathcal{P}$) ⇔ クライアント :** ここでの通信は, クライアントが最初に提供する認証データによってセキュアチャネルを張ることで, 完全性・機密性を確保できる.
- **サーバ S ⇔ クライアント :** S から提供されるデータは, 権限証明書に含まれる公開鍵で暗号化されるため, そのデータは, 秘密鍵を所持する主体のみが享受可能.

次に**「株主優待券にかかわる安全性」**から考察する.

- **改竄 :** 発行者によりディジタル署名がなされているので不可.
- **本人のコピーによる二重使用 :** サーバが, 株主優待券に付加されたシリアル番号を記録しておき, それを検証することで防ぐことが可能.
- **委譲と同時の使用 :** この場合, サーバがシリアル番号をチェックした後に, 発行者からも, 委譲により破棄された同じシリアル番号が通知されることになる. このことから検出可能.
- **他人のコピーによる二重使用 :** サーバから提供されるデータは, 本当の所持者が所有する秘密鍵によってのみ復号できるので, その秘密鍵をも入手しない限り不可能. 秘密鍵が盗まれた場合においても, サーバは一度使用されたシリアル番号を記録しているため, 同一の証明書が提示された際に検証できる. そのためサーバ側に実害はない. 秘密鍵が漏洩した場合の責任はユーザにあることに注意する.
- **二重委譲の防止 :** 委譲の際には, 発行サーバに戻し, もとの権限証明書を破棄するため. ある主体が, 同一の一枚の株主優待券を二人以上に対して委譲することは出来ない.
- **破棄された株主優待券の使用 :** これは, 株主優待券の委譲後の利用ともいえる. この場合, 株主優待券を破棄した発行サーバは, そこに含まれるシリアル番号をサーバに対して通知するため, 破棄された株主優待券の使用は検出可能.

最後に, **「特殊権限証明書にかかわる安全性」**につい

て考える．基本的に，特殊権限証明書は，本人の属性情報を保証するものであるため委譲されることはない．よって，委譲に際しての問題点を考える必要はなくなる．特殊権限証明書に関しては，以下の点を考慮する．

- **改竄：** プライバシ保護サーバによりディジタル署名がなされているので不可．
- **他人の特殊権限証明書をコピー使用：** サーバは，クライアントとの通信の際に，チャレンジを行い，特殊権限証明書中の公開鍵に対応する秘密鍵を持つ主体かどうか確認する．
- **過去の特殊権限証明書の使用：** 証明書自身に，有効期限が含まれているのでサーバにより検出可能．

4.2 プライバシ重視の観点からの考察

株主優待券の使用においてこの枠組みを用いることで，ユーザは自身の ID 情報をサーバに対して秘匿することが出来る．また，年齢・性別などの情報を自身の ID を明かすことなく，サーバに対して提示できる．さらに，この場合ユーザが提供する情報は必要最低限の情報でよい．これは，特殊権限証明書は，ユーザが選択した要素の正当性のみを保証することによる．これらのことから，この枠組みは，既存の方式よりもユーザに対して自己情報コントロール権を与えるものであるといえる．

5. ま と め

5.1 PKIX による同様の手法への適用

認証局が，ローカルな (その認証局にとってのみ意味を持つ) ID 証明書に，権限を入れることで同様の枠組みを構築することが出来る．よって PKIX においても，プライバシ保護サーバのような特殊な CA を構成要素として考えることで，今回実装したものと同様の枠組みを構築することが可能である．ただし，この場合，権限証明書を用いる方法と比較して，証明書のサイズが不必要に大きくなることが予想される．これは，特に特殊権限証明書においては不必要に大きくなる．なぜなら，X.509 証明書は多数のフィールドがあり，ユーザが提供したい情報のみを含む特殊権限証明書においては，ほとんどのフィールドは意味をなさないものであるからである．

5.2 最 後 に

本論文においては，「ユーザの自己情報コントロール権を重視した権限管理の枠組み」を提案し，株主優待券に適用し実装を行った．その際，年齢・性別などといった属性情報を提示するための枠組みを新たに与えた．これにより株主優待券に加えて属性情報が必要とされるときでも，必要最低限の情報しか公開する必要がない新しいタイプの株主優待券のサービスを提案した．今回の枠組みにおいて問題となりうるのは，株主優待券の委譲に株主優待券発行サーバが関与する必要があることである．このことから，株主優待券発行サーバの処理速度が遅い場合，株主優待券発行サーバに負荷が集中し，委譲における通信にオーバヘッドが生じる可能性がある．さらに，委譲の枠組みは，クライアント間で行えることが望ましいと考えられる．以上二つの事より，委譲の枠組みに関しては，今後の研究課題の一つである．また，実装においては SPKI 権限証明書を用いているが，PKIX においても，プライバシ保護サーバのような特殊 CA は理論上構築可能であり，PKIX を用いても同様の枠組みが構築できる．証明書のサイズなどの観点からは，SPKI 権限証明書を用いる方式が有利であると考えられる．しかし，PKIX がインフラとして確立した時には，インフラとして構築することの手間を考えると，PKIX によってこれらの枠組みを構築する方が無難であるとも考えられる．この点に関してはこれからも研究を進める必要がある．

6. 謝 辞

本論文の執筆にあたって，NEC 情報通信メディア研究本部の宮内 宏氏，佐古 和恵博士には貴重なコメントをいただきました．記して深く感謝申し上げます．

参 考 文 献

1) 梅澤 健太郎，齊藤　孝道，奥乃 博：SPKI (Simple Public Key Infrastructure) によるプライバシー重視の権限管理の提案と Java を用いた実装，情報処理学会第 60 回全国大会，3Q-03，Mar. 2000.
2) 浜田 良樹："プライバシーの権利とインターネット"，CyberSecurityManagement，JapanCyberSecurityInstitute,Vol3,5,6,2000
3) C. Ellison : Establishing Idetity Without Certification Authority, *Proc. of USENIX Security Symp.*, '96.
4) C. Ellison, *et al.* : SPKI Certificate Theory, RFC2693, May 1999.
5) C.Ellison : SPKI Requirements, RFC2692, Sep. 1999.
6) T. Saito, K. Umesawa, Wu Wen, Hiroshi G. Okuno, Access Control by SPKI Certificate, *JW-ISC2000*,pp.143-150, 沖縄, Jan. 2000.