プライバシーを重視したアクセス制御機構の提案

梅澤 健太郎[†] 齋藤 孝道[††] 奥乃 博[††]

プライバシーの権利を「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」（自己情報制御権）と規定し，この自己情報制御権を保証する枠組みとして，サーバに対してクライアントの個人情報開示を自由に制御する方式を提案する．PKIX のような一般的な公開鍵基盤では，認証と権限管理が一体化しているので，自己情報制御権を実現することが難しい．それに対して，本論文では，SPKI で提案された ID を介さない公開鍵と権限とを直接結び付ける権限証明書を使用する．さらに，個人情報証明書という必要な個人情報だけを保証する証明書を提案し，個人情報証明書発行サーバを使用して必要なときに個人情報証明書を発行することを通じて，個人情報の選択的な開示を可能にしている．そして提案方式の適用例として，Java による電子株主優待券のシステムの実装を示す．

Proposal of a Privacy Enhanced Access Control Mechanism

Kentaro Umesawa,[†] Takamichi Saito[††] and Hiroshi G. Okuno[††]

The right of controlling the exposure of personal information is adopted as the definition of the right of privacy in this paper. The privacy-enhanced access control mechanism is attained by the machanism of controlling the exposure of personal information at a server. This mechanism is difficult to realize by the Public Key Infrastructure (PKI) becuase it associates a public key and authority tightly through an identifier, and thus it knows the personal information associated to the identifier. Instead of using PKI's certificates, *Authorization Certificates* of SPKI (Simple Public Key Infrastructure) are used. By requesting an authorization certificate of least necessary information on demand to a personal information certificate issuing server, a user can control to what extent of personal information is exposed. The proposed system is implemented by Java and applied to a system of complimentary tickes for stockholders. This application shows the effectiveness of the proposed privacy-enhanced mechanism.

1. はじめに

安全な通信の下で，最小限の個人の情報だけを提示して，ネットワークサービスを受けたいというのは，多くの個人の要求であろう．SSL に代表される通信路上でのデータの暗号化は，第三者に対して通信路を守るものであり，クライアントが提供した ID などの情報を，サーバから秘匿することを目的とするものではない．このような情報公開制限への要求は，IT 時代のプライバシー保護とも呼ばれているが，その定義は明確ではない．我々は，単純な匿名性ではなく，**プライバシーの権利を「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」**（**自己情報制御権**）という考え方[1)] を採用する．

このような「自己情報制御権を保証する枠組み」として，サーバに対してクライアントの個人情報開示を自由に制御する方式を実現するうえで課題となるのは，公開鍵基盤（Public Key Infrastructure，PKI）との関係である．公開鍵暗号を用いた安全な通信のためにはインフラストラクチャである PKI が不可欠である．その 1 つとして，PKI（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 では，X.509 証明書のような ID 証明書を発行する母体として認証局（Certificate Authority，CA）を使用し，証明書として X.509 を用いる．

アクセス制御における証明書の機能は，それに関係する ID（識別子，たとえば，名前など），公開鍵，権限という 3 つの主体間の関係として規定されるので，一般に次の 2 つの機能に大別できる．

(1) 認証：公開鍵と ID の結び付きを保証する 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

† 東京理科大学大学院理工学研究科情報科学専攻
Graduate School of Sciences and Engineering, Science University of Tokyo
†† 東京理科大学理工学部情報科学科
Department of Information Sciences, Science University of Tokyo

ACL（Access Control List）☆〕を確認し，IDの属性である権限を決定する．

PKIXでは，IDを介して権限と公開鍵を結び付けているので，認証と権限管理が一体化している．その結果，IDを保証するX.509証明書を用いて権限管理を行おうとすると，匿名サービスといったプライバシーを重視したネットワークサービスの実現が難しい．

このような問題点を解決するためのアプローチの1つとして，SPKI（Simple Public Key Infrastructure）の権限証明書（Authorization Certificate）の利用が考えられる．それは，IDを介さず，権限と公開鍵とを直接結び付けるので，認証を分離した簡潔なアクセス制御方式が実現できる[8),9),13),14)]．

本論文では，その匿名アクセス制御方式をベースに，今回新たに，PKIXのような認証基盤の存在を前提とするネットワークモデルにおいて，SPKIの証明書を利用して上述の自己情報制御権を保証するために，サーバに対してクライアントの個人情報開示を自由に制御する方式を提案する．つまり，クライアントは，自分でここまでは公開してもかまわないと考える範囲の個人情報をサーバに与え，サーバがそれに応じたアクセス制御を行うことができるシステムである．また，この方式もこれまで我々の提案してきた匿名アクセス制御方式[8),9),13),14)]と同様に認証とアクセス制御の分離を行い，クライアントのIDからどのようなサービスを享受したかを特定することを困難なものにしている．

以下，関連研究を述べ，2章でSPKIについての背景知識を与える．3章で，提案する方式を概説し，4章で，処理の流れを説明する．5章で，提案する方法を株主優待券利用に応用し，その有効性を示す．6章で，提案方式の安全性について考察し，本論文をまとめる．

1.1 関連研究

崔ら[10)]は，匿名アクセス制御に関する研究として，SPKIの鍵生成と証明書の発行にかかるコストを軽減する方式を提案している．このコスト軽減方式は，我々が提唱する方式と組み合わせて利用することも可能である．ただし，我々の研究の目標は，SPKIを利用することではなく，クライアントの選択的属性情報開示をも含むプライバシー確保が可能な匿名アクセスコントロールの枠組みを提供することにある．

須賀ら[11)]は，X.509属性証明書[5)]を利用したクライアントのプライバシー保護の方式を提案している．この方式では，属性情報の選択的開示を「ドーナツ型属性証明書」と呼ばれるX.509属性証明書を利用することで実現している．これは，多数の属性情報を含む属性証明書の一部分だけを選択開示できる機構である．これに対して，本論文で提案する方式は，必要な属性情報だけを必要なときに属性証明書として発行することを通じて，情報の選択的な開示を可能にしている．我々の方式の適用分野は，限定的な権限利用のアプリケーションを想定しているので，須賀らの方法よりも属性証明書のサイズが小さくなり，また，属性証明書の発行枚数の増加はそれほど問題にならないと考えている．もちろん，このトレードオフは，適用するアプリケーションの規模により左右される．

また，山崎ら[12)]は，ID証明書に対して与信という形で属性情報を付加する方式を提案している．この研究では，属性情報を証明書から切り離して，属性情報管理ディレクトリに属性情報を保存させることを考えている．一方，我々の方式では，クライアント自身が手元に属性情報を入手し，権限証明書との関連により自己情報制御権を確立することを狙っている．山崎らの方式は，認証基盤とともに用いて属性情報によるアクセス制御を行う方法であるが，我々の方式は「匿名性の確保」と「属性情報によるアクセス制御」の両立を目的とする．

2. SPKIの概要

SPKIは，Ellisonの論文[2)]をきっかけに始まり，現在，RFC2962，2963で規定されている[3),4)]．これらのRFCの中では，名前空間に関する記述が中心であり，その多くはグローバルな名前空間とローカルな名前空間の比較であるが，それらの名前空間の定義が記述されているわけではない．したがって，まず，名前空間に関して考察を行い，グローバルな名前空間，ローカルな名前空間という意味を定義する必要がある．

一般的に名前空間は，主体，もしくは，実体を示すID（識別子）の集合である．あるIDと，ある主体Aとの関係を定め，その対応関係を適当な主体の間で共有することにより，その関係を共有した主体の間では，IDと主体Aの対応を識別することができる．逆に，対応関係を共有していない主体にとっては，そのIDが主体Aを指し示すことは知りえない．このように，対応関係が特定の主体だけに開示された名前空間を**ローカルな名前空間**と呼ぶ．一方，対応関係が広く公表された名前空間を**グローバルな名前空間**と呼ぶことにする．この例としては，X.500Name[6)]がある．

我々のアイデアは，SPKIの権限証明書が，クライ

☆ サーバにおける各主体のIDと権限の対応をエントリとするリストを指す．

アントの名前などの ID を含まないという事実に着目し，ローカルな名前空間を共有する主体を制限することで匿名性およびプライバシを重視した簡潔な権限管理の実現を目指したことにある[8),9),13),14)]．SPKI における権限証明書は，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，それ自身に直接的に ID を含んでいない．具体的な **SPKI 権限証明書**（以降，**権限証明書**と呼ぶ）の様式は，以下のような 5-tuple に発行者の電子署名を付加したものとなっている[4)]：

⟨*Issuer*, *Subject*, *Delegation*, *Authorization*, *Validity*⟩

Issuer：権限証明書の発行者の公開鍵，もしくは，その主体自体を示すシンボル．本論文では，発行者の公開鍵とする．

Subject：権限証明書の発行を受ける主体の公開鍵，もしくは，その主体自体を示すシンボル．本論文では，権限を行使する主体の公開鍵とする．

Delegation：ブール値．True，または，False．*Subject* がさらに権限を委譲することが可能かどうかを示す．

Authorization：権限を示す．

Validity：証明書の有効期間を示す．

なお，複数の権限証明書の簡略化に関しては，文献 8)，9)，13)，14) に従う．

3. アクセス制御システムの構成

提案するプライバシーを重視したアクセス制御システムは 2 つの基本機能から構成される：

(1) **匿名アクセス制御**：サービスを享受するクライアントのサーバに対する匿名性は，ID を含まない権限証明書の利用と，証明書の発行過程と行使過程の分離という概念に基づいている．この分離のために善意の第三者として **Issuing Agent**（後述）を導入する．これにより，証明書発行時のクライアント選別と証明書行使時の匿名アクセスを可能にしている．ここで，匿名アクセスといっても anonymous FTP のような無制限な匿名アクセスではないことに注意する．また，Issuing Agent により，だれにどのような権限証明書（権限）を発行したかを管理することもできる．

本来，この枠組みは，クライアント間での権限委譲が可能な枠組みである．しかし，紙面の都合と本論文の目的を考慮のうえ，本論文ではクライアント間での権限委譲に関する議論は割愛する．

(2) **クライアントの個人情報開示の制御**：クライアントの個人情報は ID 証明書と同様に，サーバが信頼できる善意の第三者によって，証明書の形で発行される必要がある．さらに，クライアントの年齢，性別などの個別の情報を個別の証明書として発行する．これにより，必要最低限の情報のみを相手に与えることが可能になる．この機能を実現するために**個人情報証明書発行サーバ**（以降，**Privacy Server** と呼ぶ）を導入する．これは，事前にクライアントから ID，年齢，性別などの個人情報の登録を受け，クライアントからの証明書発行要求時にクライアントの認証を行い，その**個人情報と公開鍵の対応を保証する証明書**（以降，**個人情報証明書**と呼ぶ）を発行する．

Privacy Server は Issuing Agent と同様に，S と C から信頼を受ける主体なので，Privacy Server と Issuing Agent のサービスを同一のマシンで運営してもかまわない．ただし，Privacy Server と Issuing Agent は概念的に異なったものであることに注意する．

3.1 証 明 書

提案システムでは，2 種類の証明書を使用する（ここで，Issuing Agent（後述）を IA，Client（後述）を C，Privacy Server（後述）を PS とする．さらに，主体 X の公開鍵を $P(X)$ で，$P(X)$ に対応する主体 X の秘密鍵を $S(X)$ で表す．また証明書に対して $S(X)$ で電子署名が施されていることを，$\langle \ldots \rangle_{S(X)}$ で表現する）：

(1) **権限証明書**：匿名アクセス制御に用いる証明書は，SPKI 権限証明書のフォーマットを利用し，以下のように定義する：

$$\langle P(IA), P(C), D, A, V \rangle_{S(IA)}$$

ここで，*Delegation* D，*Authorization* A，*Validity* V に関しては 2 章で示した SPKI 権限証明書の定義に従う．

また，提案システムの実装においては，権限証明書にシリアル番号のフィールドを追加することも考えられる．この場合，公開鍵 $P(C)$ の代わりに，公開鍵と比較してデータ量の小さいシリアル番号を権限証明書の識別子として権限証明書の管理に利用する．こうすることにより，公開鍵をキーとする検索処理において，キーとなるデータが小さくなることから，処理量の減少が予想される．

(2) **個人情報証明書**：クライアントの個人情報開示の制御に用いる証明書のフォーマットは，以下のように定義する：

$$\langle P(PS), P(C), SI, V \rangle_{S(PS)}$$

SI は，C の個人情報（たとえば，年齢，性別など）．SI は，C のすべての個人情報ではなく，C の個人情報の一部であることに注意する．また，V は証明書自体の有効期間を示している．

3.2 システムを構成する主体

提案システムは 4 つの主体から構成される：

(1) **Server**：サーバ．Client に対してサービスを提供する主体，その際，Client に対して個人情報を要求する．権限証明書，個人情報証明書を検証できる．

(2) **Client**：クライアント．Server に対してサービスを要求する主体で，自己の ID を Issuing Agent に登録してある主体．また，自己の ID および性別や年齢などの個人情報を Privacy Server に登録してある．

(3) **Issuing Agent**：Server からの委託により，Client に対して権限証明書を発行する主体．Server, Client ともに信頼する存在．Client からの要求を受けて，Client を認証し，権限証明書を発行する．

(4) **Privacy Server**：Client からの委託の下に，Client に対して個人情報証明書を発行する．Server, Client ともに信頼する存在．Client からの要求を受けて，Client を認証し，個人情報証明書を発行する．

ここでの 'Server' と 'Client' は，本論文で提案する方式を構成する各主体を指し，一般のサーバ・クライアントモデルにおけるそれらと区別できるように，前者を英字で，後者をカタカナでそれぞれ表記することにする．

3.3 公開鍵について

提案システムで使用する公開鍵ペアを以下に示す：

(1) $\boldsymbol{P(S)}$, $\boldsymbol{S(S)}$：Server S の ID と対応付けされた公開鍵ペア．

(2) $\boldsymbol{P(IA)}$, $\boldsymbol{S(IA)}$：Issuing Agent IA の ID と対応付けされた公開鍵ペア．

(3) $\boldsymbol{P(PS)}$, $\boldsymbol{S(PS)}$：Privacy Server PS の ID と対応付けされた公開鍵ペア．

(4) $\boldsymbol{P'(C)}$, $\boldsymbol{S'(C)}$：Client C の ID と対応付けされた公開鍵ペア．

(5) $\boldsymbol{P(C)}$, $\boldsymbol{S(C)}$：Client C の権限または個人情報と対応付けられる公開鍵ペア．C が権限ごとに作成する．

(1)〜(4) の公開鍵は，PKIX のような認証基盤で ID との対応が保証されているとする．そのため，すべての主体はある主体の ID と対応付けられた正しい公開鍵を入手することが可能である．これらの公開鍵は ID と対応付けられるため，(5) と比較して有効期間が長い．また，これらの公開鍵の失効管理に関しては，PKIX のような認証基盤の仕組みを利用することを前提としている．

(5) の公開鍵は，Client が権限ごとに作成し，権限または個人情報との対応が IA または PS により保証される．この公開鍵は権限と対応付けられるため，(1)〜(4) と比較して，有効期間が短い．また，この公開鍵の失効管理に関しては，Issuing Agent が Client C から公開鍵 $P(C)$ の失効について通知された際に，Server に公開鍵 $P(C)$ の失効を通知することで行う．

3.4 主体間の通信

ここでは，主体間の通信について説明する．以下で，$P'(C)$ は C の認証用公開鍵であり，$P(C)$ は C が権限ごとに作成する公開鍵であることに注意する．

(1) **Server S と Issuing Agent IA との間の通信**：3.3 節のとおり，S と IA は互いの正しい公開鍵 $P(S)$, $P(IA)$ を獲得可能である．よって，S と IA は通信相手の公開鍵を用いることで，相互認証および通信の暗号化（または通信の暗号化用の共通鍵の共有）を行う[7]．

(2) **Client C と Issuing Agent IA との間の通信**：3.3 節のとおり，C と IA は互いの正しい公開鍵 $P'(C)$, $P(IA)$ を獲得可能である．よって (1) と同様，相互認証および通信の暗号化を行う．

(3) **Client C と Privacy Server PS との間の通信**：3.3 節のとおり，C と PS は互いの正しい公開鍵 $P'(C)$, $P(PS)$ を獲得可能である．よって (1) と同様，相互認証および通信の暗号化を行う．

(4) **Client C と Server S との間の通信**：3.3 節のとおり，C は S の正しい公開鍵 $P(S)$ を入手することが可能である．それに対し，C は S にアクセスする際に直接 ID を提示しないため，S は C の正しい認証用公開鍵 $P'(C)$ を獲得可能できない．しかし，C が S にアクセスする際，公開鍵 $P(C)$ を含む権限証明書を S に送信する（この際，特に権限証明書を暗号化する必要はない）．よって S は $P(C)$ を用いたチャレンジにより，C が正当な権限を持つこと，すなわち $S(C)$ を持つこと，を確認できる．さらに，C は $P(S)$ を，S は $P(C)$ を持つことから通信の暗号化を行う．

3.5 主体の保持する情報

提案するアクセス制御システムでは Server S, Issuing Agent IA, Privacy Server PS は公開鍵ペア以外に以下のような情報を保持する：

(1) **Server S** の保持する情報：

(a) $\boldsymbol{List1_S}$：IA に対して『権限証明書をクライアントに発行する権限』を与えたことが記されているリスト．

(b) $\boldsymbol{List2_S}$：有効期間内に失効した権限証明書に含まれる Client C の公開鍵 $P(C)$ をエントリとするリスト．$P(C)$ は，C から $P(C)$ の失効の通知を受けた IA が S にその旨を通知されたものである．S は，

IA から $P(C)$ の失効の通知を受けた際，そのエントリを記録する．S は $P(C)$ をキーとしてこのリストを検索することで，C から送信された権限証明書に含まれる公開鍵の有効性を確認する．ここで再度 $P(C)$ は C が権限ごとに作成する公開鍵であることに注意する．

(2) **Issuing Agent IA** の保持する情報：

(a) $\boldsymbol{List1_{IA}}$：Client C の ID と権限ごとの公開鍵 $P(C)$ の対応をエントリとするリスト．C の ID は，C が事前に IA に登録するものである．$P(C)$ は，権限証明書の作成時に C が IA に送信したものである．IA が，C に対して $P(C)$ を含む権限証明書を発行する際，そのエントリを記録する．問題の発生時に管理上必要とあれば，IA は $P(C)$ をキーにしてこのリストを検索し，C の ID を特定する．

(b) $\boldsymbol{List2_{IA}}$：Client C の ID と C に発行可能な権限の対応をエントリとするリスト．C の ID は，C が事前に IA に登録するものである．C の権限は，IA が S から与えられるアクセス制御ポリシーを基準として，IA が保持する C の情報をもとに決定される．IA は C の ID をキーとしてこのリストを検索することで，権限証明書に含める権限を決定する．$List2_{IA}$ の具体的な作成方法については，5 章で解説する．

(3) **Privacy Server PS** の保持する情報：

(a) $\boldsymbol{List1_{PS}}$：Client C の ID とその個人情報（年齢，性別など）の対応をエントリとするリスト．C の ID および個人情報は，C が事前に PS に登録するものである．PS は C からの登録があった際，そのエントリを記録する．PS は C の ID をキーとしてこのリストを検索することで，個人情報証明書に含める個人情報を決定する．

4. 具体的な権限行使までの流れ

各主体は 3.5 節で示した情報を保持し，3.4 節で説明した通信路を用いて処理を行う．**図 1** を用いて，具体的な処理の流れを説明する．

Step0：個人情報の登録

C は，前もって C のある個人情報 SI を含むいくつかの個人情報を Privacy Server PS に登録しておく．この際に PS は 3.5 の $List1_{PS}$ に C の ID とそれらの個人情報に関するエントリを追加する．個人情報の登録はオフラインまたは 3.4 節 (3) の通信路を用いて行う．

Step1：発行権限委譲

S は，Issuing Agent IA に，『Client に権限証明書を発行する権限 $Auth1$』を権限証明書 $Cert1$ を用いて委譲する．この通信は，3.4 節 (1) の通信路で行われる．この際に，S は $List1_S$ に『IA に対して権限証明書を発行する権限 $Auth1$ を有効期間 $V1$ で与えた』というエントリを追加する．ここで，$Cert1$ は以下のようになる：

$$\langle P(S), P(IA), \mathrm{True}, Auth1, V1\rangle_{S(S)}$$

また，IA は ID を登録済みの Client に関して $List2_{IA}$ を持つ．ここでは，IA が，C が S に対して $Auth2$ の権限を保持することを決定しているとする．

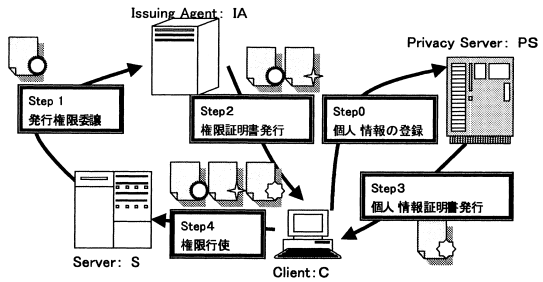

図 1 権限証明書の発行から利用までの処理の概要図
Fig. 1 Outline chart of processing.

Step2：権限証明書発行

C は S において権限を行使するために，IA から権限証明書 $Cert1$，$Cert2$ の発行を受ける．以下で詳細を述べる：

Step2-1：C は $Cert1$，$Cert2$ の発行を受けるために，3.4 節 (2) の通信路を用いて IA にアクセスする．

Step2-2：Step2-1 における C の認証が成功した場合，IA は C に C が権限ごとに作成する公開鍵 $P(C)$ の送信を要求する．この $P(C)$ が $Cert2$ に含められることになる．

Step2-3：C は 3.4 節 (2) の通信路を用いて $P(C)$ を IA に送信する．

Step2-4：IA は，$Cert2$ に含める値を決定し，作成する．つまり，有効期間 $V2$，C がさらなる権限委譲を行えるかを表すブール値 $D2$（ここではつねに False）を，IA が定める．特に権限に関しては，IA は $List2_{IA}$ を C の ID をキーとして参照し，$Cert2$ に含める権限を決定する．この場合 Step1 での前提より，IA は C が権限 $Auth2$ を持つことが分かる．以上から $Cert2$ が作成される：

$$\langle P(IA), P(C), \mathrm{False}, Auth2, V2\rangle_{S(IA)}$$

Step2-5：IA は $Cert1$，$Cert2$ を 3.4 節 (2) の通信路を用いて C に送信する．この際，IA は $P(C)$ と C の ID との対応を，$List1_{IA}$ にエントリとして追加する．

Step3：個人情報証明書発行

C は自分が PS に登録した個人情報の一部である個人情報 SI を S に選択的に開示するために，PS から個人情報証明書 $ACert$ の発行を受ける．以下で詳細を述べる：

Step3-1：C は $ACert$ の発行を受けるために，3.4 節 (3) の通信路を用いて PS にアクセスする．

Step3-2：PS は Step3-1 における C の認証が成功した場合，C に対して $ACert$ に含める個人情報と対応付ける公開鍵 $P(C)$ の送信を要求する．ここで $P(C)$ は，Step2-2 の公開鍵 $P(C)$ と同一のものである．同時に PS は C に登録した個人情報のうち，どの個人情報を $ACert$ に含めるのかを通知するよう要求する．

Step3-3：C は 3.4 節 (3) の通信路を用いて $P(C)$ を PS に送信する．同時に，C はある個人情報 SI に関する $ACert$ を発行してほしい旨を PS に通知する．

Step3-4：PS は $ACert$ を作成する．この際，PS は $ACert$ に含める値（$P(C)$，個人情報，有効期間 $V3$）を決定する．特に，個人情報 SI に関しては，PS が $List1_{PS}$ を C の ID をキーとして参照し，$ACert$ に含める値を決定する．この場合 Step0 より PS は，C がある個人情報 SI を登録してあることが分かる．以上から $ACert$ が作成される：

$$\langle P(PS), P(C), SI, V3\rangle_{S(PS)}$$

たとえば，C の年齢に関する個人情報証明書は以下のようになる：

$\langle P(PS), P(C), \text{`Age} = 19\text{'}, \text{`20001225000000'}\rangle_{S(PS)}$

ここで，これは C が 19 歳であることを示す年齢に関する個人情報証明書で，2000 年 12 月 25 日 0 時 0 分 0 秒まで有効なものである．

Step3-5：PS は 3.4 節 (3) の通信路を用いて $ACert$ を C に送信する．

Step4：権限行使

C は S に自分の保持する権限 $Auth2$ を行使する．以下で詳細を述べる：

Step4-1：C は $Auth2$ を行使するために，3.4 節 (4) の通信路を用いて $Cert1$，$Cert2$，$ACert$ を S に送信する．

Step4-2：S は $Cert1$，$Cert2$ を簡略化する．その結果，C の権限を表す権限証明書 $Cert3$ が一意に定まる．$Cert3$ を以下に示す：

$$\langle P(S), P(C), D', Auth', V'\rangle_{S(S)}$$

ここで，権限証明書の簡略化は文献 8)，9)，13)，14) に従う．

Step4-3：S は $List2_S$ を参照し，$P(C)$ の有効性を確認する．

Step4-4：S は 3.4 節 (4) のとおり，C が $ACert$，$Cert3$ に含まれる権限ごとの公開鍵 $P(C)$ の保持者であることを確認する．

Step4-5：S は $Cert1$，$Cert2$，$Cert3$ の正当性を検証する．この場合，各証明書の電子署名の検証，有効期間の確認を行う（具体的な権限証明書の検証の方法は，文献 8)，9)，13)，14) を参照）．

Step4-6：S は $ACert$ の検証を行う．$ACert$ の正当性は，有効期間 $V3$ の確認および $P(PS)$ を用いた電子署名の検証により確認できる．

Step4-7：Step4-5，Step4-6 の検証が成功した場合，S は 3.4 節 (4) の通信路を用いて，C に $Cert3$ に含まれる権限 $Auth'$ に対応したサービスを提供する．権限証明書の利用が一般的なチケットのように 1 回限りの場合は，S は $List2_S$ に $P(C)$ のエントリを追加する．

5. 『株主優待券の利用』への応用

提案する方式を電子株主優待券の発行，および，その行使に応用する．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，個人情報証明書と組み合わせることで，以前は難しかった個人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，株主優待券は匿名性を持つこと，および，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．株主優待券は会社から発行されるのではなく，その会社の株式の販売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株主優待券の発行権限を与えているという解釈が成り立つ．

5.1 システムの構成

3 章で示したアクセス制御システムと株主優待券の利用のためのシステムの対応を与える．

5.1.1 システムを構成する主体

(1) **サーバ S**：3.2 節の 'Server' に対応する．ある株式会社 $\mathcal{COM}$ のリソースである．$\mathcal{COM}$ は，S を用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，$\mathcal{COM}$ は，だれ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．本実装においては，S が株主 C に提供するサービスは特定の Web ページの

閲覧許可とする．

(2) **株主優待券発行サーバ IA**：3.2 節の 'Issuing Agent' に対応する．証券会社 $\mathcal{FIRM}$ のリソースである．$\mathcal{FIRM}$ は株主 C の ID を保持している．また，株主優待券の発行権限を S から与えられており，保有株式数に応じて各株主に対して割り当てられる株主優待券（権限証明書）の情報を持つ．

(3) **株主 C**：3.2 節の 'Client' に対応する．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．C は株式優待券を行使して，S からサービスを享受する．

(4) **個人情報証明書発行サーバ PS**：3.2 節の 'Privacy Server' に対応する．株主 C からの要求に応じて，個人情報に関する証明書を発行する．PS は，C の ID だけでなく，その他の個人情報（年齢，性別など）を持つ．

5.1.2 株主優待券，個人情報証明書

ここでは，実装システムでの 2 つの証明書の仕様を与える：

株主優待券は，

$$\langle P(IA), P(C), D, A, V, N\rangle_{S(IA)}$$

で定義される．ただし，$P(IA)$ は，株主優待券発行サーバの公開鍵．$P(C)$ は株主が株主優待券ごとに作成する公開鍵．D は株主優待券の委譲許可．A は，具体的に享受できるサービス．V は株主優待券の有効期間．N はシリアル番号であり，株主優待券の失効管理で使用する．シリアル番号を追加する意図は，3.1 節 (1) で記述した証明書管理の負荷削減にある．

本実装において，株主優待券はテキストファイルであり，$P(IA)$，$P(C)$ は Base64 符号化された文字列とする．$S(IA)$ による電子署名も Base64 符号化され，株主優待券と同一のテキストファイルに格納されている．D は True または False という文字列．V は，期限開始・期限終了の 2 つの文字列で yyyymmddhhmmss の形式で記述する．N は，シリアル番号を表す文字列．A は閲覧できる URL のリストであり，$(URL_1, URL_2, \ldots, URL_N)$ のように記述する．今回，A を前述のように記述したが，株主優待券は S が解釈できればよい．そのため，A の記述方法は S と IA の事前の取り決めにより様々な形態が考えられる．また本実装においては，S が C に提供するサービスは特定の Web ページの閲覧許可としたが，提供するサービスは様々なものが考えられる（オンラインショッピングで使用できる割引券など）．

個人情報証明書は，

$$\langle P(PS), P(C), SI, V\rangle_{S(PS)}$$

で定義される．各項目の意味は，3.1 節で定めたものである．実装においては株主優待券と同様テキストファイルである．「$P(PS)$」，「$P(C)$」，「$S(PS)$ による電子署名」は Base64 符号化された文字列として記述する．SI の記述様式は，属性 $X_1, X_2, \ldots, X_N$ に関する個人情報が $x_1, x_2, \ldots, x_N$ である場合，$(X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N)$ と記述する．ここで，$x_1, x_2, \ldots, x_N$ は，C が PS に登録した個人情報のうちこの証明書で開示する一部の個人情報であることに注意する．V の記述様式は，株主優待券と同様とする．

5.2 処 理 系

JDK1.2[15], **JSDK2.0**[15], **ApacheJServ1.1**[16], **IAIK2.51**[17] によって，実装を行った．株主優待券発行サーバ IA，サーバ S，個人情報証明書発行サーバ PS のインタフェースは Servlet[15] を用いて WWW サーバとして実装した．

株主優待券の発行，検証のロジック，および，各主体が暗号化・復号処理などに用いるツールは，以前作成したライブラリ[8),9),13),14)] の一部を修正して利用した．以下で，実装システムの概要を示す．

5.2.1 実装システムの概要

本システムは証券会社が提供するオンライントレードのようなサービスを前提とする．C の認証用公開鍵 $P'(C)$，S の公開鍵 $P(S)$，IA の公開鍵 $P(IA)$，PS の公開鍵 $P(PS)$ は，それぞれの ID との対応を X.509 証明書として CA により保証されているとする．そして，S，PS，IA を X.509 証明書をサーバ証明書として利用することで SSL 対応の Web サーバとして動作させる．また，C は $P'(C)$ に関する X.509 証明書を Web ブラウザに組み込んでおく．この前提の下で，各主体間の通信は以下のように実現される：

(1) **IA と S の通信**：3.4 節 (1) に対応し，SSL 相互認証を用いて行う．

(2) **IA と C，PS と C の通信**：3.4 節 (2)，(3) に対応し，SSL 相互認証を用いて行う．PS および IA に対する C の証明書発行要求は，HTTP の POST メソッドで実現する．この際，発行要求に含まれる公開鍵 $P(C)$ は Base64 符号化された文字列として送信する．また，C は発行された証明書を HTTP を用いたダウンロードにより入手する．

(3) **C と S の通信**：3.4 節 (4) に対応する．この際，C の S に対する匿名性を保つためには，C と S との通信は SSL 相互認証が利用できない．そこで，C と S との通信は以下のように実現する：

(a) C は S にアクセスし，S の X.509 証明書を利

用した SSL サーバ認証を行う．これにより C と S との間で機密性・完全性の確保された通信路を得る．

(b) C は HTTP の POST メソッドにより株主優待券および個人情報証明書を S に送信する．

(c) S は乱数を株主優待券に含まれる公開鍵 $P(C)$ を用いて暗号化し，チャレンジとして C に提示する．

(d) C はチャレンジを $S(C)$ により復号し，POST メソッドにより S に送信する．

(e) S は C からのレスポンスを検証し，C が $S(C)$ を保持することを確認する．

(f) チャレンジ・レスポンスがうまくいった場合に，S は C に株主優待券に記載された権限に応じたサービスを提供する．

5.3 株主優待券の利用

実装システムの処理概要を 4 章と対応させつつ示す．

5.3.1 個人情報の登録

4 章 **Step0** に対応する．株主 C は，あらかじめ個人情報証明書発行サーバ PS に，自分自身の ID と個人情報（年齢，性別，etc.）を登録済みであるとし，PS は $List1_{PS}$ を持つ．

5.3.2 発行権限委譲

4 章 **Step1** に対応する．株式会社 $\mathcal{COM}$ のサーバ S は，証券会社 $\mathcal{FIRM}$ の株主優待券発行サーバ IA に，自身のリソースに対する株主優待券を発行する権限を権限証明書 $Cert1$ の形で与えており，IA は $Cert1$ を保持するとする．

5.3.3 株主優待券発行

4 章 **Step2** に対応する．証券会社 $\mathcal{FIRM}$ の IA は，株主 C の要求に応じて，$Cert1$，株主優待券 $Cert2$ を発行する．$\mathcal{FIRM}$ は，C の株取引から C の保有する株式を把握しており，C に対して発行できる株主優待券についての情報を持つ（この情報は $List2_{IA}$ に相当する）．

5.3.4 株主優待券の利用

4 章 **Step4** に対応する．株主 C は $Cert1$，$Cert2$ を，サーバ S に対して提示することで自分の保持する権限を行使する．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個人情報の提出を求められる場合は，個人情報証明書もあわせて提出する．今，C が株主優待券を利用しようとしていて，S が C の年齢情報を要求しているとする：

(1) この処理は 4 章 **Step3** に対応する．C は個人情報証明書発行サーバ PS にアクセスし，年齢に関する個人情報証明書 $ACert$ を取得する．

(2) C は，$Cert1$，$Cert2$，および，$ACert$ を S に送信する．

(3) S は，与えられた $Cert1$，$Cert2$，$ACert$ を検証する．検証が成功した場合，S は C に対して株主優待券 $Cert2$ に記載された権限に応じたサービスを提供する．

6. 考　　察

6.1 提案方式の再考

我々の提案する方式についての特徴をまとめ，提案する方式の利点について述べる：

(1) **ID を利用しないアクセス制御**：サーバにクライアントの ID を晒さないアクセス制御を可能にする．個人情報の漏洩を考慮したアクセス制御を実現する際，1 つの解決策になるであろう．

(2) **クライアントの自己情報制御権を保証**：(1) の特徴を持ちつつ，クライアントがサーバに対し選択的に個人情報を提示できる．

(3) **権限を持つサーバにより証明書が発行される**：サービスを提供するサーバは，委譲する権限証明書を自分の支配下に置ける．PKIX における CA による発行と違い，証明書は最終的に発行者に戻ってくる．したがって，権限証明書のフォーマットなどの変更を容易にできる．ただし適用されるアプリケーションの規模が大きくなると，相互運用性の問題が生じる可能性もある．

(4) **サーバは独自の名前空間を持たず，信頼できる第三者の持つ名前空間を利用**：権限を持つサーバは，アクセスしてくる主体の ID のデータベースを持たなくてよい．サーバはクライアントの管理から開放されるなどの利点が考えられる．

6.2 安　全　性

提案システムの安全性を 2 つの側面，つまり，『通信路中での安全性』，『権限証明書および個人情報証明書という 2 つの証明書に関わる安全性』から論ずる．

前者に関しては 3.4 節の通信路を用いることで機密性・完全性が確保できるので，以下では後者について考える：

(1) **権限証明書および個人情報証明書の改竄**：証明書に発行者の電子署名が付加されているので証明書に対する改竄は検出可能．

(2) **本人のコピーによる権限証明書および個人情報証明書の二重使用**：Server が使用された証明書に含まれる Client C の権限に応じた公開鍵 $P(C)$ を，その証明書の有効期間内は記録しておき，証明書の検証の際にそのリストを参照することで防止可能．

(3) **複製された権限証明書および個人情報証明書の他者による使用**：Server がサービスを提供する際に，証明書に含まれる公開鍵 $P(C)$ を利用したチャレンジ・レスポンスを行うことで防止可能．ここで秘密鍵 $S(C)$ が漏洩した場合の責任は Client にあることに注意する．

(4) **破棄された権限証明書および個人情報証明書の使用**：権限証明書を破棄する際，Issuing Agent は，そこに含まれる Client C の公開鍵 $P(C)$ を Server に対して通知するため，破棄された証明書の使用は検出可能．ここで，権限証明書と個人情報証明書に含まれる公開鍵 $P(C)$ は同一のものであるため，Server は権限証明書の破棄の有無を確かめるだけでよいことに注意する．

6.3 今後の課題

(1) **適用できるアプリケーション規模の検討**：現在の我々の方式は，権限の発行から行使までが閉じた単一アプリケーションを想定している．また，クライアントが個人情報を個人情報証明書として自分自身で保持する．その結果，アプリケーションが大規模になるとクライアントの証明書管理の問題や，証明書の相互運用などの問題が生じることが予想される．このような問題を解決するためには，本論文で行った定性的な評価以外に，実験システムによる定量的な評価が不可欠である．定量的な評価を通じて，これらの問題を解決する方法を探るとともに，他のアクセス制御方式と比較して適切なアプリケーションの規模を検討していく必要がある．

(2) **個人情報の記述様式**：権限証明書と比較して，個人情報証明書は PKIX の ID 証明書と同様に複数のサーバ，アプリケーションで共通利用される可能性がある．そのため個人情報証明書に含まれる個人情報の記述に関して，統一的な記述様式を検討する必要がある．

(3) **公開鍵生成コストの削減**：提案システムで利用する権限証明書および個人情報証明書に含まれる公開鍵は，それぞれ権限または個人情報ごとに異なったものとすることを前提とする．そのため，Server S は Client C のある権限 $Auth_1$ とそれとは別の権限 $Auth_2$ にそれぞれ対応づけられた 2 つの公開鍵から，2 つの権限を有する主体が同一の主体であることは知りえない．ただし，同一の主体が保持する権限証明書，個人情報証明書の枚数が増えると必要となる公開鍵ペア数も当然増加する．また，C が S に対して同一の権限証明書および個人情報証明書を複数回提示するときに，それに関連付けられた同一の公開鍵から，S が同一の主体からのアクセスであることを推測できるという問題もある．この問題は，1 回の使用を前提とした使い捨ての公開鍵を利用することで解決できる．しかし，この方法も必要となる公開鍵ペア数を増加させる．よって，証明書に含まれる公開鍵を利用した個人の追跡を防止することを考える場合，公開鍵ペアの生成コストを軽減する必要がある．

7. おわりに

本論文では，『クライアントの自己情報制御を重視したアクセス制御の方式』を提案した．この方式によりクライアントの ID を晒さずに，クライアントがよいと思う程度にクライアントの個人情報をサーバに与え，それに応じたアクセス制御を行うことを実現できることを示した．この方式は，認証とアクセス制御の分離と使い捨て公開鍵の利用により，クライアントが享受したサービスの特定を困難なものにしている．さらに，個人情報証明書と権限証明書に同一の使い捨て公開鍵を含ませることにより，個人の特定をせずにサーバに個人情報を渡すことができる．また，提案方式の適用例として，株主優待券のシステムの実装を示した．

謝辞 お忙しい中，懇切丁寧に，洞察力の豊富なコメントをいただきました査読者各位に感謝いたします．

参考文献

1) 浜田良樹：プライバシーの権利とインターネット，*Cyber Security Management*, Vols.3, 5, 6, Japan Cyber Security Institute (2000).
2) Ellison, C.: Establishing Idetity Without Certification Authority, *Proc. USENIX Security Symp.* (1996).
3) Ellison, C.: SPKI Requirements, RFC2692 (1999).
4) Ellison, C., et al.: SPKI Certificate Theory, RFC2693 (1999).
5) Farrell, S., et al.: An Internet Attribute Certificate Profile for Authorization, RFC2026 (2000).
6) CCITT. Recommendation X.500: The directory-overview of concepts, models and services (1988).
7) Schneier, B.: *Applied Cryptography*, 2nd ed., John Wiley & Sons (1996).
8) Saito, T., Umesawa, K., and Okuno, H.G.: Privacy Enhanced Access Control by SPKI, *Proc. 7th Int'l Conf. on Parallel and Distributed Systems*: *Int'l Workshop on Next-Generation Internet Technologies and Applications 2000*

(*NGITA00*), pp.301–306 (2000).
9) Saito, T., Umesawa, K., and Okuno, H.G.: Privacy-Enhanced Access Control by SPKI and Its Application to Web Server, *Proc. IEEE 9th Int'l. Workshops on Enabling Technologies*: *Infrastructure for Collaborative Enterprise* (*WETICE 2000*), pp.201–206 (2000).
10) 崔　浩哲，菊地浩明，中西祥八郎：効率的な匿名権限委託，コンピュータセキュリティシンポジウム 2000, pp.61–66 (2000).
11) 須賀祐治，荒木啓二郎：公開鍵インフラにおける属性証明書の利用について，ソフトウェア・シンポジウム 99 (1999).
12) 山崎重一郎，荒木啓二郎：信用情報と利用ポリシーの管理が可能な相互認証を実現する認証基盤の提案，情報処理学会論文誌，Vol.40, No.1, pp.296–309 (1999).
13) 梅澤健太郎, 齋藤孝道, 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 (2000).
14) 梅澤健太郎，齋藤孝道，奥乃　博：SPKI によるプライバシー重視機能の提案とその株主優待券の電子的発行への応用，情報処理学会セキュリティ研究会 (2000).
15) http://java.sun.com
16) http://java.apache.org
17) http://jcewww.iaik.tu-graz.ac.at

(平成 12 年 11 月 30 日受付)
(平成 13 年 6 月 19 日採録)

梅澤健太郎（学生会員）

1976 年生．2000 年東京理科大学理工学部情報科学科卒業．同年同大学院理工学研究科情報科学専攻修士課程進学．

齋藤　孝道

1995 年東京理科大学理工学部情報科学科卒業．1997 年同大学院理工学研究科情報科学専攻修士課程修了．同年，同学科助手．

奥乃　　博（正会員）

1950 年生．1972 年東京大学教養学部基礎科学科卒業．日本電信電話公社，NTT，科学技術振興事業団北野共生システムプロジェクト，東京理科大学理工学部情報科学科を経て，2001 年 4 月より京都大学大学院情報学研究科知能情報学専攻教授．博士（工学）．この間，スタンフォード大学客員研究員，東京大学工学部客員助教授．推論機構，音環境理解の研究に従事．2001 年度 International Society of Applied Intelligence 最優秀論文賞受賞．本学会英文図書委員．著編書：『インターネット活用術』（岩波書店，1996），“Computational Auditory Scene Analysis”（共編，LEA，1998）等．