プライバシに配慮したWWWにおける個人属性認証・アクセス制御システム

本城信輔[†] 洲崎誠一[†]
齋藤司[††] 三浦信治[†††]

現在普及しているWWWにおける認証・アクセス制御技術は，ユーザのプライバシに配慮していないため，ユーザの追跡が可能な点や，必要以上に個人情報を明らかにしてしまう点が問題であった．本論文では，プライバシに配慮した個人属性による認証・アクセス制御技術を提案する．ここで提案される認証方式は，複数のWWWサーバの結託に対しても，ユーザの追跡を許さない．また，P3Pと呼ばれるプライバシ保護基盤により，ユーザが個人情報の公開を自動制御することが可能となった．これによりユーザの意図しない個人情報の漏洩を防ぐことができる．また，P3Pに加えた独自拡張により，個人情報を公開せずに，情報の品質によってアクセス制御することが可能となった．

Private Attributes Based Authentication and Authorization System on WWW

Shinsuke Honjo,[†] Seiichi Susaki,[†] Tsukasa Saito[††] and Nobuharu Miura[†††]

This paper proposes a new scheme for the private-attributes-based authentication and authorization system on WWW. While the conventional systems have no consideration of user privacy, the system has advantage in protecting it. In the proposed authentication method, WWW sites could not track users, even if they conspire each other. The system also enables users to take part in the authorization without disclosure of unnecessary private information, with the help of the P3P privacy protection technology. With the extension to P3P, the system allows users to access with the attributes of the private information without revealing the information itself.

1. はじめに

現在普及しているWWWにおける認証・アクセス制御システムは，何らかの形でユーザの個人情報を必要とするものが多く見られる．これらは，サービス時のアクセス制御のための情報として用いられる．たとえば，ユーザID・パスワード認証（Basic認証）を実施するシステムの多くは，ユーザの個人情報の登録を受けて，その情報をもとにユーザID発行しアクセス権限を設定する．また，公開鍵証明書や属性証明書による認証を実施するシステム[1)]では，クライアントは公開鍵証明書や属性証明書をWWWサーバに提示し，WWWサーバはこれらに記載されている個人情報をもとにアクセス制御を行う．

上記の技術は，ユーザの特定とアクセス制御を実現するものである．しかし，一方で，アクセス制御は実施したいが，ユーザが必要以上に個人情報を明らかにすることを好まないシステムもあろう．たとえば，アクセスに年齢制限を設けるシステムや，性別に応じたコンテンツを提供するようなシステムにアクセスするユーザは，年齢や性別といった情報は提供するが，それ以外の個人情報の公開の必要性を感じることはないであろう．また，アクセス制御に年齢を要求するシステムと，メール・アドレスを要求するシステムへのアクセスの結果，2つのシステムが結託し情報を収集することによって年齢とメール・アドレスを関連付けてしまう可能性がある場合，ユーザは両方のシステムへのアクセスを躊躇するであろう．したがって，このようなシステムでは，認証行為自体から必要以上に個人

† 日立製作所システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.
†† 日立製作所公共システム事業部
Government & Public Corporation Information Systems Division, Hitachi Ltd.
††† 日立公共システムエンジニアリング
Hitachi Government & Public Corporation System Engineering, Ltd.

情報を開示してしまうことを防いだり，複数システムのアクセスの結果を追跡することによるユーザ情報の収集を不可能にしたりするような機構が必要である．

このような要求に対して，現在普及している認証・アクセス技術を適応することは容易ではない．たとえば，ユーザ ID・パスワード方式のように個人情報の登録を要求するシステムの場合，アクセス権限の設定に必要な情報以外の個人情報の登録を要求される可能性がある．また，公開鍵証明書や属性証明書に関しても，これらに掲載されている個人情報が，システムのアクセス制御の実態によらず認証のたびに提示されるため，必要以上に情報を公開してしまう可能性がある．一方，ユーザ追跡に関しては，特に公開鍵証明書による認証で生じる問題である．ユーザは，認証ごとに同じ証明書あるいは同じ証明書へのポインタを持つ属性証明書を利用するために，システムの結託によるユーザの追跡が容易である．

本論文では，上記の問題を解決するためにプライバシに配慮した WWW における個人情報による認証・アクセス制御システムを提案する．本システムの特徴は次のとおりである．

- ユーザ追跡不可能な認証・アクセス制御機構
  認証・アクセス制御自体から，ユーザが特定できてはならない．本システムの認証・アクセス制御方式は，複数の WWW サーバが結託しても，ユーザを追跡・特定することはできない．
- ユーザ合意による個人情報公開
  個人情報の公開は，ユーザが WWW サイトの個人情報取扱い方針に合意したうえでなされる．本システムでは，P3P [3] を基本とした個人情報公開制御機構を利用している．これにより，意図しない個人情報の流出を防止できる．
- 信頼性のある個人属性によるアクセス制御
  WWW サーバに送信される個人情報は，第三者機関の署名が施された信頼性のある情報である．また，第三者機関が信頼度を設定することも可能である．

以降，2 章では本研究の方針と位置付け，特にプライバシ保護方針について述べる．3 章では，システムの概要を説明する．4 章では，プロトコルについて述べる．5 章では，個人情報の保護の機構に整理する．6 章では，実装や安全性，プライバシ保護に関する評価について述べる．7 章では，本システムの利用例について説明する．8 章では，提案方式と他方式との比較や解決すべき課題について述べる．9 章では，本論文で得られた結果を簡単にまとめる．

2. 本研究の方針と位置付け

2.1 プライバシ保護方針

この節では，ユーザのプライバシ保護のために提案システムが採用した方針について説明する．

2.1.1 追跡不可能な認証・アクセス制御

認証・アクセス制御方式のうち，認証や権限証明に静的な鍵を利用するシステムでは，ユーザの追跡が可能である．つまり，サーバに提供するアクセス権限情報はサービスごとに変更させるのが通常の利用形態であるが，それらに関連付けられる認証用の鍵が固定的であるために，鍵そのものから，それらのアクセス権限情報が関連付けられてしまう恐れがある．アクセス権限情報は，ユーザの何らかの属性を表現していることが多いことを考えれば，プライバシ保護という観点でこのことによる被害は深刻である．また，公開鍵証明書のように鍵に ID 情報が直接関連付けられている場合，証明書そのものからユーザの特定が可能になる．

したがって，認証行為からユーザの特定や追跡を不可能とするには，次の対策をとる必要がある．

- アクセス権限情報を変えるたびに鍵を更新する．
- 鍵そのものに直接 ID 情報を関連付けない．

上の対策をとるという前提に立ち，公開鍵を基本としたシステムと共通鍵を基本としたシステムについてそれぞれ考察してみる．公開鍵を基本とするシステムでは，権限と関連付けられる鍵として公開鍵暗号方式の鍵を利用する．なお，鍵と権限の関連付けに関しては，SPKI [4),5)] のように簡易証明書を使わず，アクセス権限情報を X.509 公開鍵証明書の拡張領域に入れる方が，認証プロトコルとして広く普及している SSL のクライアント認証機構を利用することができるといった利点がある．鍵と ID 情報の関連付けの防止については，アクセス権限を暗号化することと，公開鍵証明書に直接ユーザを特定できる情報を記載しないことで対処できる．一方の鍵の更新についてであるが，よく知られているように，公開鍵・秘密鍵生成にかかる時間はシステム運用上無視できないくらい遅く，したがって実用にあたって何らかの対策が必要である．たとえば，あらかじめシステムが必要十分な公開鍵・秘密鍵を生成しておくことで，頻繁なアクセス権限情報の発行や更新に備えておくことは可能である．ただしこの方法をとっても，どの程度の数の鍵のペアを用意すればよいのかといった運用上解決しなければならない問題は依然として残る．

共通鍵を基本としたシステムでは，権限と関連付けられる鍵として共通鍵暗号方式の鍵や鍵付き MAC の

鍵を利用する．このシステムの特徴は，公開鍵方式の裏返しで，鍵生成にかかる時間は短く実用に適しており，アクセスの多いシステムでも容易に対応可能である．また，共通鍵を利用する方が効率上優れたプロトコルを設計しやすいという利点もある．その反面，権限と鍵を関連付ける方式や認証プロトコルを新たに設計する必要がある．

以上のことから，我々は，共通鍵をもとにしたシステムの方が設計上および運用上の問題の解決が比較的容易であると判断する．したがって以降，本論文では，共通鍵を基本としたシステムとその実装について説明していく．

2.1.2 ユーザ合意による個人情報公開（P3P）

これまで，ユーザが個人情報を WWW サイトに公開する場合，公開する情報の内容はサイトによって決められており，ユーザに交渉の余地はなかった．また，ユーザが個人情報を開示する際，それらを公開するかどうかについて，アクセスごとにユーザが確認する必要があった．サイトによる個人情報の利用促進とユーザのプライバシ保護の両方を満足するには，

- サイトがユーザの個人情報の取扱い方針を明示する，
- ユーザはサイトの方針を吟味して，個人情報の提供の是非を判断する，

といった枠組みが必要である．

W3C[2] で標準化が進められている P3P[3] は，このような枠組みの上でプライバシ保護を実現するものである．P3P では，WWW サイトの個人情報の取扱い方針（Policy）を XML で記載するための仕様を定めている．また P3P Project は，P3P Policy に対するユーザの行動やユーザ側の GUI の動作を設定しておくための APPEL と呼ばれる User Preference の XML 表現の標準化も進めている．XML 化された Policy や Preference は S/W の処理に適しており，個人情報の開示制御を自動化することが可能となる．すなわち，ユーザ側の P3P エージェントは，サイトの P3P Policy を入手して内容を解析し，あらかじめ設定された APPEL をもとに，ユーザの代わりに情報の提供の是非を自動的に判断する．また必要があれば，GUI によりユーザに警告を表示しユーザの注意を促す．提案システムではこの P3P，APPEL を基本にして，ユーザが WWW サイトの個人情報の取扱い方針に合意する機構を実現する．

2.2 関連研究と本研究の位置付け

通信におけるプライバシ保護に関する技術は，様々な立場から研究されている．たとえば，受信者に対して送信者のアドレスを秘匿することを目的とするものに，暗号データを途中経路でリレーする方式[6),7)] や，送信者を Crowds と呼ばれるグループからの送信と見せかける方式[8)] が提案されている．認証においてプライバシを保護する技術には，理論的なアプローチとして，匿名 credential を用いる方式[9),10)] や事後に匿名性を無効化するような方式[11)] が提案されている．また，ユーザがデータベースから情報を取得する際，どの情報に興味があるかデータベースに分らないようにする方式[12)] や，電子商取引において WWW サーバには商品とユーザ情報との関連付けが不可能であるような方式[13)] が提案されている．

また，本研究と同様にアクセス制御システムにおける匿名性を目指す技術として，権限と公開鍵証明書を分離した SPKI による権限管理システム[14)~16)] が提案されている．ところで，このシステムにおいてユーザの追跡を防止するためには，新たに権限証明書を発行するたびに公開鍵証明書の更新が必要になる．したがって，この方式を現実のシステムとして稼動させるには，すでに述べたような公開鍵の生成の問題を解決しなければならない．

本研究は，アクセス制御に個人情報の開示が要求される環境において，個人情報の関連付けやユーザの追跡を不可能とする実用可能なシステムの実装を目的とする．すでに述べたように，提案システムは WWW サーバとクライアントの間で，使い捨ての共通鍵による認証を実施する．そのため，Kerberos[17),18)] の共通鍵システム，特に Davis[19)] による Kerberos の公開鍵拡張を基礎に，普及している PKI 技術や SSL を組み合わせてシステムを設計した．また，提案システムにおいては，個人情報と認証用の共通鍵はチケットと呼ばれるデータに格納されネットワークを流通するが，クライアントにおけるチケット利用の是非の判断に P3P のプライバシ保護機構を利用し，ユーザの意図しない個人情報の公開を防いでいる．ところで，P3P の仕様では個人情報の公開の具体的方法については触れておらず，一般の WWW アプリケーションに対する適応は容易ではないが，本研究で，チケットという具体的な個人情報公開手段に P3P を適応したことにより，その有効性をある程度示すことができた．また提案システムでは，P3P に独自の拡張を加え，個人情報の品質（信頼度と公開度）によってアクセス制御できるように変更を加えた．

3. システムの概要

3.1 システム構成

提案システムでは，個人情報はチケットと呼ばれるXMLデータによってやりとりされる．チケットは後に述べるように，偽造・改ざん・盗聴・不正利用などの攻撃に耐性を持つようにセキュリティ機能を具備したデータである．提案システムは，次のように個人情報登録サーバ/チケット発行サーバ，WWWサーバ，クライアントから構成される．

- 個人情報登録サーバ/チケット発行サーバ（Registry Server/Ticket Granting Server, RS/TGS）
  ユーザの個人情報を管理するサーバ．ユーザはあらかじめこのサーバに個人情報や公開鍵証明書を登録しておく必要がある．また，ユーザの要求に応じてデータベースに登録してあるユーザの個人情報を参照し，必要最低限の情報から構成されるチケットを発行する．
- WWWサーバ
  ユーザの個人情報によりアクセス制御を行うWWWサーバ．WWWサーバは，あらかじめアクセス制御対象のResourceと要求する個人情報およびその条件からなるアクセス制御情報を設定しておく．また，ユーザの個人情報の取扱いに関する方針を記載したP3P Policyも準備しておく．提案システムでは，このP3P PolicyとアクセスPolicyと呼ぶ．Server Policyは，必要に応じてアクセスがあったクライアントに提供される．
- クライアント
  ユーザが操作するWWWブラウザおよびチケット処理S/Wをクライアントと呼ぶ．ユーザはこのクライアントを通じて，個人情報を用いたアクセス制御システムに参加する．クライアントの役割は，TGSからチケットの発行を受けて，WWWサーバへのアクセス要求をチケットとともに送信する．またクライアントはP3Pエージェントとして機能し，WWWサーバから送られるServer Policyの解析を行う．したがって，ユーザはクライアントを通じて個人情報の公開を制御することができる．

3.2 プロトコルの概要

個人情報によるアクセス制御をWWW上で実現するために，HTTPの上で稼動するWWWアプリケーションに透過なプロトコルを新たに設計した．プロトコルのデータはXMLによって表現される．提案システムのプロトコルの概要を**図1**に示す．

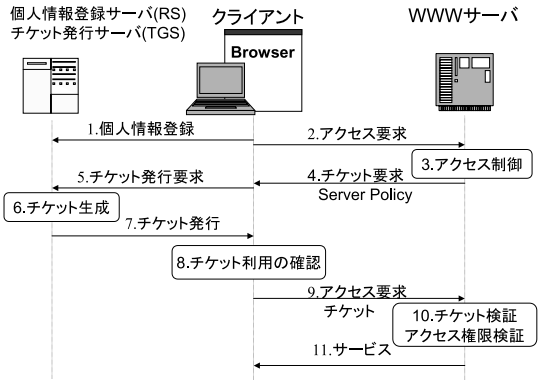

図1 プロトコルの概要
Fig. 1 Overview of the protocol.

1. **個人情報登録** ユーザはあらかじめ個人情報や公開鍵証明書を個人情報登録サーバRSに登録しておく．この個人情報登録サーバRSの役割は個人情報に信頼性を与えることである．
2. **アクセス要求** ユーザはクライアントを操作して，WWWサーバ上のResourceに対するアクセス要求を送信する．
3. **アクセス制御** WWWサーバは，要求されているResourceがアクセス制御対象であるか確認する．通常のWWWサーバの場合は，ここでResourceに対するアクセスを許し，対応するResponseを返す．
4. **チケット要求送信（チケットなし）** アクセスが要求されているResourceがアクセス制御対象である場合は，チケットを要求するメッセージをクライアントへ送信する．同時にWWWサーバの個人情報の取扱い方針やアクセス制御情報を記載したServer Policyを添付する．
5. **チケット発行要求** クライアントは受信したServer Policyを解析し，ユーザにWWWサーバの個人情報の取扱いに関する方針を表示する．ユーザは，これらの情報を閲覧しWWWサーバに要求されている個人情報を開示するか判断する．あるいはP3Pエージェントとして自動的に処理してもよい．ユーザの了承を得てクライアントは，チケット発行サーバTGSへチケット発行要求を送信する．
6. **チケット生成** チケット発行要求を受信したチケット発行サーバTGSは，個人情報登録サーバRSのデータベースに記録してある個人情報を参照し，必要最小限の個人情報とその制御情報から構成されるチケットを発行する．

7. チケット送信 チケット発行サーバ TGS は，クライアントへチケットを送信する．なお，チケット発行要求からチケット送信までのクライアントと TGS とのやりとりは，SSL の相互認証によって確立された安全な通信路により保護される．

8. 個人情報，チケット利用の再確認 クライアントは，ユーザにチケット記載の個人情報を提示する．また，再度ユーザに WWW サーバの個人情報の取扱いに関する方針を表示し，ユーザは情報開示に関する最終確認をする．

9. アクセス要求（チケットあり） クライアントは，チケットを添付してアクセス要求を WWW サーバに送信する．このとき，チケット保持証明を WWW サーバに対して行う．

10. チケット検証，アクセス制御 WWW サーバはチケットの正当性を確認する．さらに，チケット記載の個人情報がアクセスに必要な条件を満足していることを，Server Policy を参照して確認する．

11. サービス アクセス権限が確認できれば，WWW サーバは要求されている Resource のサービスをクライアントに提供する．

上記のプロトコルは，初期状態の場合の説明である．すでに WWW サーバの Server Policy を保持している場合や，すでに利用可能なチケットを保持している場合は，クライアントの判断により適宜省略することが可能である．しかし，このようなキャッシングを行う場合，WWW サーバで Server Policy が更新されても，クライアントは古い Server Policy やそれに基づいて発行された古いチケットを使いつづけてしまうことに注意する．つまり，Server Policy の更新に対応しないと，必要以上に個人情報を公開してしまう可能性がある．したがって，キャッシュはクライアントの方針に応じて，有効期限を設定する必要がある．

4. プロトコル

本章では，クライアントと WWW サーバ，クライアントと RS/TGS 間でやりとりされるプロトコルについて説明する．

これらのプロトコルは HTTP の上で稼動し，プロトコルのデータは XML でエンコードされる．

4.1 プロトコルの設計方針

本システムは，すでに述べたように共通鍵を基本としたチケットを利用するため，Kerberos[17),18)] を参考としてプロトコルを設計した．一方，利便性を考慮し WWW サーバと RS/TGS は公開鍵を保持することとする．Kerberos と PKI の統合については，従来より様々な方法が提案されている．たとえば IETF の Kerberos WG[20)] では，クライアントと KDC の間を公開鍵による認証に置き換えた PKINT[21)]，KDC を介さないでクライアントとサーバ間で直接通信する PKDA[22)] や，PKINT と PKDA を統合した PKTAPP[23)]，Realm 間認証に利用する方法などが提案されている．

ところで Davis[19)] によって，WWW サーバと KDC だけに公開鍵を持たせる Kerberos の PKI 拡張が提案されている．このプロトコルの前提と，本システムの，RS/TGS と WWW サーバが公開鍵を持つがクライアントは WWW サーバに対して共通鍵を利用する，という前提は同じである．したがって Davis のシステム[19)] を参考に，HTTP 上のプロトコルを新たに設計することとした．

4.2 個人情報チケットの構造

個人情報が記載されるチケットの構造を次に示す．

$$Ticket = P_S(S_R(R, L_{Ticket}, PIs, K_{Ticket}, S))$$

ここで，R は RS/TGS の情報，L_{Ticket} はチケットの有効期間，PIs は個人情報を指す．K_{Ticket} は，チケットの保持証明に利用される鍵であり，クライアントも保持している．S は WWW サーバの情報である．$P_S(\cdot)$ は WWW Server S の公開鍵による暗号化，$S_R(\cdot)$ は，RS/TGS R による署名を表現する．ところで鍵 K_{Ticket} は，チケット発行に際して毎回生成・配布され，チケットが使用される期間にのみ利用される．この意味で鍵 K_{Ticket} の利用は一時的に限られ，公開鍵インフラのように長期的に利用するものではない．チケットでは，個人情報 PIs と関連付けられているのがこのような一時的な鍵 K_{Ticket} であるので，X.509 公開鍵証明書や属性証明書，あるいは SPKI のように公開鍵からユーザを追跡することはできない．また，チケットは公開する個人情報が変わるたびに発行されるものであり，アクセス記録やチケットの情報からユーザを追跡することも不可能である．つまり，ユーザはプライバシ保護のために，個人情報 PIs のあり方に注意を払うだけでよい．なお，本システムでは WWW サーバが不正を行うことも考慮し，WWW サーバによるチケットの不正利用を防ぐために WWW サーバ名 S を署名に入れている．また，暗号化された通信路の利用を前提としないので，チケットは個人情報 PIs を含めて暗号化する．

4.3 事前準備

まず，事前準備としてユーザは SSL クライアント認証用の秘密鍵・公開鍵（証明書）を用意し，証明書

を個人情報とともに RS/TGS に登録する．同様に，RS/TGS はチケット署名用の秘密鍵・公開鍵（証明書）cert-rstgs と SSL サーバ証明書を用意しておく．

WWW サーバは，チケット暗号用の秘密鍵・公開鍵（証明書）cert-server を準備しておき，自分で保持しておく．また，後に説明するようにクライアントと SSL 通信をするのであれば，SSL サーバ公開鍵証明書も準備しておく．この場合，チケット暗号用の証明書 cert-server と SSL サーバ証明書の保持者は同一にしておく．

以後各 SSL 認証において，通信相手のルート証明書は信認済みであるものとし，したがって SSL において証明書のチェーンが扱えるものとする．なお，必要に応じてそれぞれは互いの証明書のチェーンの有効性確認を実施してもよい．

4.4 チケット要求プロトコル

クライアントと WWW サーバとのやりとりは**図 2** に示してある．クライアントが，チケットを要求する WWW サーバに request を送信してアクセスすると，サーバはチケット暗号用の証明書 cert-server と Server Policy をクライアントに返す．

なお，中間攻撃や成りすましによって攻撃者が証明書 cert-server の差し替えを行った場合は，攻撃者の鍵で暗号化されたチケットを RS/TGS が生成してしまい，結果として，攻撃者に不正に個人情報を知られてしまうことになる．したがって，これを防ぐため，公開鍵証明書 cert-server にはドメイン・ネームなどの記載があるべきであるし，またクライアントはこの証明書が通信相手の WWW サーバのものであることを確認しなくてはならない．このあたりの議論は，SSL における証明書の確認の議論[24] と同じである．

同様に，Server Policy が改ざんや差し替えの攻撃を受けると，利用者が必要以上に個人情報を開示してしまう可能性がある．もし証明書 cert-server が暗号と署名の両方に利用できるならば，Server Policy に署名を施すことで対応できる．また，クライアントと WWW サーバ間で SSL で保護された通信路を利用してもよい．この場合，クライアントは SSL サーバ証明書の保持者と暗号用の証明書 cert-server の保持者が同じであることを確認する必要がある．なお，送信者の特定が可能なクライアント認証は利用してはならない．

4.5 チケット発行プロトコル

チケット発行プロトコルは，**図 3** に示してある．クライアントは，まず RS/TGS との間に SSL コネクションを確立し，そのうえで発行プロトコルを開始する．SSL はデフォルトのサーバ認証に加えクライアント認証を併用し，クライアントと RS/TGS で相互認証を実施する．クライアント認証は先に RS/TGS に登録してある SSL 用の公開鍵証明書を用いる．この SSL コネクション上において，クライアントはチケット発行要求として，WWW サーバのチケット暗号用の証明書 cert-server，Server Policy，アクセス希望先 URL を RS/TGS に送信する．RS/TGS は発行要求を受けチケットを生成し，応答としてクライアントに $Ticket$，認証用の鍵 K_{Ticket}，個人情報 PIs，有効期限 L_{Ticket}，RS/TGS の公開鍵証明書 cert-rstgs を送信する．この RS/TGS の証明書 cert-rstgs は，チケットの署名検証に利用される．

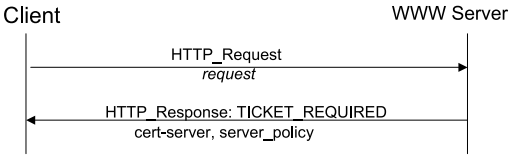

図 2 チケット要求プロトコル
Fig. 2 Ticket request protocol.

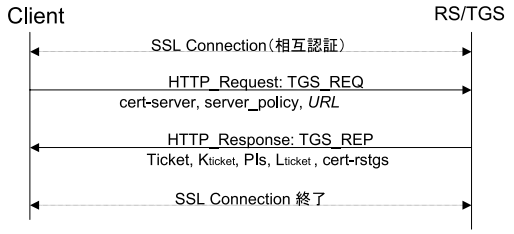

図 3 チケット発行プロトコル
Fig. 3 Ticket granting protocol.

なお，RS/TGS はこのとき必要に応じて CRL や OCSP[25] などによる WWW サーバの公開鍵証明書の有効性確認を実施してもよい．効率性を憂慮しチケット発行の度に有効性確認をするのを避けたい場合は，定期的な有効性確認の結果をキャッシュに保存しておき，ある程度の期間はキャッシュを利用するのが現実的な実装であろう．キャッシュを用いた場合，証明書の確認と実際の失効の間の時間的な差がさらに大きくなることによるリスクを RS/TGS の管理者が承知しておくべきなのはいうまでもない．

また，さらに RS/TGS が独自のポリシで WWW サーバの信頼性を判断してもよい．具体的には，WWW サーバの過去の振舞いや世間的な評判による判断，あるいは，専門の調査機関に基づく信頼性評価の方法があるだろう．

いずれにしても，どこまで WWW サーバの信頼性を検証するかは，RS/TGS が安全性と効率性を斟酌

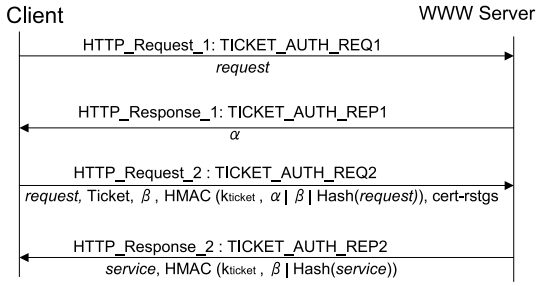

図 4 チケット保持証明
Fig. 4 Proof of possession of ticket.

して採用した運用ポリシに依存する．あるいはクライアントが WWW サーバの信頼性を確認するような運用もあるだろう．

4.6 チケット保持証明プロトコル

クライアント C が WWW サーバ S にチケットを提示してアクセスを行う場合に，第 3 者の不正利用を防ぐためにチケット保持証明を行う必要がある．本システムでは，鍵付き MAC を用いた challenge & response 方式[26)~28)] のチケット保持証明を採用する．**図 4** はその概略を示したものであり，図中の α と β は，それぞれ WWW サーバとクライアントが生成する乱数，HMAC(K_{Ticket}, ·) は，鍵 K_{Ticket} による鍵付き MAC であり，Hash($data$) は $data$ のハッシュを意味する．また $data1|data2$ は，$data1$ と $data2$ の連結を意味する．

HTTP は，Client が Request を送信することによってプロトコルが開始されるので，サーバから challenge を必要とする challenge & response を HTTP の上で実現すると，2 往復必要になってしまう．HTTP のプロトコルモデルに適合するよう 1 往復で済ますには，時刻を暗号化する認証方式を採用すればよい．しかしこのためには，クライアントと WWW サーバの時刻の同期が必要となるが，Internet でそれを期待することは現実的ではない．以上の理由から，本システムでは，乱数による保持証明[26)] を採用することにした．なお，本システムでは乱数を暗号化するのではなく，鍵付き MAC を利用する[27),28)]．HTTP_Request_2 や HTTP_Response_2 において，各 HMAC にそれぞれ Hash($request$)，Hash($service$) が含まれているのは，攻撃者による $request$ や $service$ の改ざんや差し替えを防ぐためであり，また暗号ではなく鍵付き MAC を利用するのはこのためでもある．

WWW サーバによる RS/TGS の信頼性の評価については，RS/TGS が WWW サーバに対して評価するのと同様に，証明書 cert-rstgs の有効性確認機構を利用したり，また信頼する RS/TGS を独自の基準で判断したりしてもよい．

なお，通信内容を特に秘匿したい場合は，SSL による安全な通信路を確保したうえでこのプロトコルを実施するべきである．また，この際，送信者の特定が可能なクライアント認証は利用してはならない．

5. プライバシ保護機能と個人情報の信頼性

5.1 チケットの個人情報

チケットに記載される個人情報は次のように，各個人情報と制御情報から構成される．

```
PIs:= PI+    (+は 1 回以上の繰返し)
PI := (attribute,
       authorize_level, disclose_level)
```

ここで PIs は 1 つ以上の個人情報 PI からなる個人情報の集合である．attribute は P3P のスキーマで表現される個人属性で，たとえば，生年月日の年や性別などの個別情報を表す．authorize_level，disclose_level は，それぞれ承認レベル，開示レベルで次に述べるような意味を持つ．

承認レベル（authorize_level） 承認レベルは，個人情報登録サーバ RS/TGS が各個人属性の信頼性をどの程度承認するかを表現する指標である．たとえば，登録時に公的証明書によって証明された属性の場合は，高い承認レベルを持つが，ユーザの自己申告による場合は低い承認レベルを持つ．

開示レベル（disclose_level） 開示レベルは，個人属性がどの程度開示されているかを表現する．たとえば，個人属性がない場合は最低の開示レベルを持ち，全部開示されている場合は最高の開示レベルを持つ．このレベルは，住所などのように記載範囲を変更できるような属性の場合に有効である．

個人属性，承認レベル，開示レベルを組み合わせることにより，様々な応用が可能となる．たとえば，連絡先そのものは開示したくないが，個人情報登録サーバ RS/TGS に公的証明書によって証明された連絡先が登録されている事実を示したい場合は次のように表現する．

attribute	authorize_level	disclose_level
user.home-info	公的証明書で確認	非公開

ここで user.home-info は，ユーザの連絡先の P3P スキーマである．このようにして，ユーザは個人情報そ

のものを開示しなくても，情報の存在を証明することができる．

5.2 Server Policy

Server Policy は，P3P の仕様を基本にアクセス制御情報を追加するために独自拡張を加えたものである．Server Policy は次のような構造を持つ．

```
Server Policy :=
      (Resource, PIs, Privacy Policy)
```

ここで，Resource はアクセス制御対象を表し，Privacy Policy は P3P に従って記載された個人情報に関する取扱い方針である．また，PIs における各個人属性の authorize_level および disclose_level は，提供される個人属性が持つべき必要最低限の承認レベルおよび開示レベルを表す．たとえば，連絡先として必ず公的証明書による確認を要求する場合は，Server Policy には次のように記載する．

attribute	authorize_level	disclose_level
user.home-info	公的証明書による確認が必要	*

ここで * は，どのレベルでもよいことを表す．

5.3 アクセス制御

WWW サーバは，Server Policy とチケットを照らし合わせてアクセス検証を行う．Server Policy が要求する各個人属性がチケットに記載されており，チケットに記載されている承認レベル，開示レベルが Server Policy の要求を満足すればアクセス可能と判断される．

5.4 個人情報の開示制御とプライバシ保護

本システムは，P3P の個人情報開示制御機構の上に構築されている．クライアントは，WWW サーバから Server Policy を受信すると，サーバの要求する個人情報および開示レベル，承認レベルをユーザに表示する．同時に，サーバの個人情報に関する利用方針をユーザに表示し，個人情報の利用意図を理解させる．したがって，ユーザはサーバの個人情報の取扱い方針を理解したうえでチケットによるアクセスを行うかどうか判断することができる．つまり，ユーザの意図しない個人情報の開示をここで制御することが可能である．また開示レベルの制御機構によって，個人情報の公開を必要最低限に限ることが可能である．

5.5 個人情報の信頼性

個人情報の信頼性は個人情報登録サーバによるデジタル署名によって保証される．したがって，ユーザが個人情報を偽ることによる無制限アクセスの実施は不可能である．また，個人情報にそれぞれ承認レベルが設定されているため，WWW サーバが情報の信頼度を制御することが可能である．たとえば WWW サーバが，厳密な信頼性を要求される情報には高い承認レベルを要求し，一方でニックネームや頻繁に変わる勤め先の部署情報は信頼度がある程度低い承認レベルを設定するということが可能になる．つまり，WWW サーバは用途に応じた品質を持つ個人情報を要求することができる．

6. 評　　価

我々は，すでに提案システムを実装し性能の評価を行っている．ここでは，実装方法とその性能の評価に加え，安全性やプライバシ保護機能についての評価結果を示す．

6.1 実　　装

6.1.1 RS/TGS，WWW サーバ

RS/TGS および WWW サーバは，Linux Kernel 2.2.14 上の Apache HTTP Server 1.3.9 の上で稼動する．RS/TGS は HTTP Server の CGI プログラムとして実装した．また，WWW サーバ側のチケット認証・アクセス制御機構は，Apache HTTP Server の追加モジュールによって実装した．その理由は，既存の WWW アプリケーションに透過に実装するために，CGI ではなく，HTTP Server 内部でプロトコルの処理を行う必要があるからである．暗号ライブラリは OpenSSL 0.9.4，XML Parser は XML4C 3.3.1 を利用した．

6.1.2 クライアント側のチケット処理 S/W

クライアントのチケット処理 S/W は，クライアント側のマシン上で稼動する Proxy 型のプログラムである．通常の HTTP Transaction は通過させるだけであるが，チケット要求などを受けると一連のチケット発行およびチケットによる WWW サーバへの認証・アクセスを自動的に行う．**図 5**，**図 6** に，クライアントのチケット処理 S/W の画面イメージを示す．

6.1.3 データ構造とアルゴリズム

本システムのすべてのデータ，つまり，チケットおよび Server Policy，HTTP 上のプロトコルは，XML で表現されている．また，公開鍵暗号としては 1024bit の RSA，鍵付き MAC は HMAC-SHA1[29] を用いた．

6.1.4 性能の評価

本システムの諸処理の計測の結果を**表 1**，**表 2** に示す．計測は，WWW サーバ，RS/TGS ともに CPU PentiumII 400 MHz メモリ 128 MByte のマシン上で実施した．

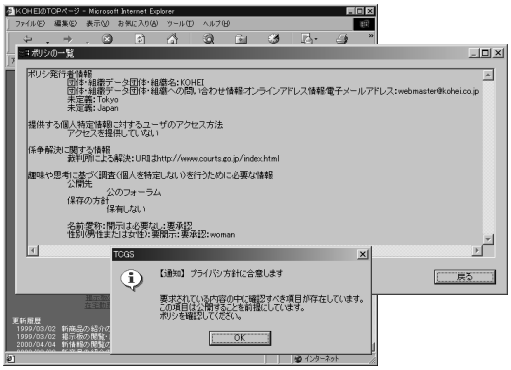

図 5 クライアントの画面イメージ：P3P Policy 情報と APPEL との照合の結果
Fig. 5 Snapshot of the client: P3P policy information and the result of the matching with APPEL.

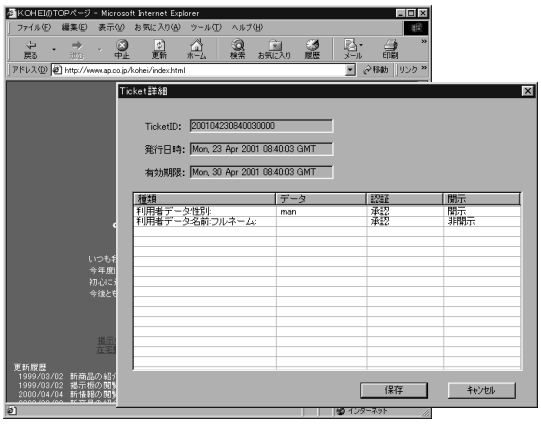

図 6 クライアントの画面イメージ：チケットの情報
Fig. 6 Snapshot of the client: ticket information.

表 1 WWW サーバ側平均処理時間
Table 1 The average time of the processing at the WWW server.

処理	全体	XML 処理 Parse/生成	HMAC 処理	チケット検証	アクセス制御
HTTP_Request_1 の検証	2.1[ms]	1.1[ms]	N/A	N/A	N/A
HTTP_Response_1 の生成	4.4[ms]	2.5[ms]	N/A	N/A	N/A
HTTP_Request_2 の検証	40.9[ms]	12.8 [ms]	1.8[ms]	22.5[ms]	2.2[ms]
HTTP_Response_2 の生成	5.5[ms]	2.8[ms]	1.8[ms]	N/A	N/A

表 2 RS/TGS Ticket 発行処理時間
Table 2 The average time of the processing at the RS/TGS.

処理	全体	XML 生成	鍵生成	暗号化 (共通鍵+公開鍵)	デジタル署名
Ticket 発行処理	31.1[ms]	4.5[ms]	0.1[ms]	6.9[ms]	17.8[ms]

表 1 には，4.6 節で説明したプロトコルの各メッセージに対して，本来の HTTP メッセージの処理以外に要する処理時間を記載している．表から分かるように，XML 関連の処理，特に Parse の処理時間が，どの処理においても処理時間の大部分を占めている．特に HTTP_Request_2 はチケットを含んでいるため数キロバイト程度の大きさになっており，その分 Parse 処理に時間がかかる．全体でみると，WWW サーバにおける HTTP_Request_2 の検証に要する時間が最も大きく，さらにこのうちの 55% の時間をチケット検証に費やす．チケット検証に時間がかかるのは，チケットの署名検証および鍵の復号化などの公開鍵暗号アルゴリズムによる処理があるからである．したがって，チケットの検証を 1 回行いその結果を後のために記録しておけば，2 回目以降のアクセス時においては，処理時間が半分以下になることが期待できる．このような効率化が安全性を損なうのは否めないが，検証結果の保持を数秒という短い時間に限れば，現実的な脅威を小さくすることができる．また，WWW のページは 1 つの HTML ファイルからいくつかの画像などのリンクが貼られていることが多く，同じクライアントからのアクセスが数秒程度の短時間に集中する傾向があることを考慮すれば，数秒程度の結果保持でも大きな効率向上が期待できる．ところで，今回計測した WWW サーバやクライアントにおける処理時間以外にも 1 往往復分のネットワークの遅延時間が加わることに注意しておく．

Ticket 生成処理については，表 2 から分かるように大部分の処理時間をデジタル署名生成に費やす．また共通鍵の生成時間が小さいため全体のチケット生成時間が 31[ms] 程度に済んでおり，本システムの特徴である共通鍵を基本とした設計の正当性を裏付けている．

6.2 安 全 性

6.2.1 チケットの偽造・改ざん・盗聴

チケットには，発行者の RS/TGS の署名がなされているので，チケットの偽造・改ざん攻撃は不可能で

ある．また，検証者である WWW サーバは発行者の RS/TGS の信頼性確認を行う必要があることも，すでに述べた．また，チケットは，正当な行使先である WWW サーバの公開鍵証明書 cert-server により暗号化されているので，攻撃者がチケットを入手しても個人情報は漏洩しない．

6.2.2 チケットの不正利用

チケットの利用にあたっては，クライアントと WWW サーバの間でチケット保持証明プロトコルを稼動し，K_{Ticket} を知っているかどうかの確認が実施される．したがって，単に攻撃者がチケットを入手しても，それを利用することはできない．一方，チケットの発行プロトコルは SSL によって保護されているため，通信路上の盗聴や中間攻撃によって K_{Ticket} を攻撃者が入手することはできない．

また，ホストへの侵入による K_{Ticket} の盗難に配慮し，実装では次のような対応を行った．RS/TGS ではチケット発行後 K_{Ticket} を記憶領域から破棄する．WWW サーバでは，効率上の理由からチケット保持プロトコル終了後も K_{Ticket} を破棄せずに保持することがあるが，10 秒程度の短い時間に限っている（効率化の点での 10 秒という時間の妥当性については，すでに 6.1.4 項で述べた）．また，クライアントは，認証後 K_{Ticket} を記憶領域から破棄するほか，チケットの有効期限内はパスワードによる K_{Ticket} へのアクセス保護を行っている．

6.2.3 不正なチケットの発行

チケット発行において RS/TGS は，クライアントから送信される WWW サーバの暗号用の証明書 cert-server を利用して，チケットを暗号化する．したがって，送信される証明書を攻撃者の証明書に不正に置き換えれば，攻撃者の公開鍵で暗号化された不正なチケットが発行され，攻撃者に個人情報が知られてしまう．

チケット発行プロトコルは，SSL の相互認証により保護されているため，クライアント以外は有効なチケット発行要求を生成できない．したがって，チケット発行プロトコルを攻撃することによるチケットの不正発行は不可能である．

ところで，チケット発行プロトコルでクライアントが送信する WWW サーバの証明書 cert-server は，チケット要求プロトコルにおいて，クライアントが WWW サーバから受け取るものである．したがって，チケット要求プロトコルにおいて，成りすましや中間攻撃により攻撃者が自分の公開鍵証明書に差し替えてしまえば，不正なチケット発行が成功してしまう．これを防ぐため，クライアントは，チケット要求プロトコルで受信する証明書 cert-server が確かに通信している WWW サーバのものであることを確認している．

6.3 プライバシ保護

6.3.1 追跡可能性

すでに述べたように，チケットはアクセス先，記載される個人情報に応じて毎回発行されるので，WWW サーバ結託によるユーザの追跡は不可能である．また，認証方式もチケットに付随した一時的な共通鍵によるものなので，鍵からユーザを特定することもできない．一方，本システムが HTTP を利用するために，その下位層のプロトコル，たとえば IP 層から，クライアントが特定される恐れはある．その対策に，Proxy サーバを経由させることによって送信者のアドレスを秘匿することは有効であり，また現実的でもある．なお，Proxy サーバはクライアントが WWW サーバに対してなすべき安全上の確認（証明書の確認）は，代理で行う必要がある．

また，一方で個人情報登録サーバに対しては，クライアントがチケット暗号化のためにアクセス先の WWW サーバの公開鍵証明書 cert-server を渡すので，ユーザのアクセス先が証明書から特定できてしまうことに注意する必要がある．

6.3.2 P3P による個人情報公開制御

P3P によってユーザが個人情報の公開を制御することが可能になった．しかし，上述したように Server Policy の改ざんや差し替えによる攻撃により必要以上に情報を公開してしまう恐れがある．それを憂慮する場合は，Server Policy に署名を施すか，WWW サーバとクライアント間で SSL で保護された通信を行えばよい．

6.3.3 個人情報の信頼性

RS/TGS は，あらかじめ登録された個人情報をもとにチケットを生成し，また必要に応じて個人情報の信頼度を設定する．また，6.2.1 項で述べたように，チケットは RS/TGS による署名で施されている．したがって，ユーザが個人情報を偽ることによる不正アクセスは，不可能である．

7. 本システムの利用例

7.1 性別によるアクセス制御

たとえば，女性向けの WWW サイトの中には，アクセスを女性にだけに許し有効な広告や意見募集を行いたい場合がある．一方，アクセスする側としては性別以外の情報を提供したくない．このような場合は，本システムの Server Policy の個人情報の条件を次のように設定すればよい．

attribute	authorize_level	disclose_level
user.gender	要承認	要開示

ここで，user.gender はユーザの性別の P3P スキーマである．このアクセス制御方針は，性別が個人情報登録サーバによって保証されていることを要請する．

7.2 掲示板システム

掲示板システムでは，参加する発言者が自由な議論ができるように匿名やハンドルネームなどによる書き込みを認めている事例が多く見られる．このシステムの問題点は，匿名を利用した他人の誹謗・中傷および犯罪目的の利用や，ハンドルネームが自己申告によるため他人による成りすましが，可能であるということにある．この問題を解決するためには，掲示板システムに本システムによるアクセス制御機構を取り入れ，Server Policy の個人情報の条件を次のように設定すればよい．

attribute	authorize_level	disclose_level
user.home-info	公的証明書による確認が必要	*
user.postal.name.nickname	*	要開示

ここで，user.postal.name.nickname はユーザのニックネームの P3P スキーマである．上のアクセス制御方針は次のことを要請している．ユーザの連絡先 user.home-info は，公的証明書で確認をとった正しい情報として個人情報登録サーバに登録されている必要があるが，連絡先の情報そのものは必ずしも公開する必要はない．つまり，問題発生時にサイトが個人情報登録サーバにチケット ID で問い合わせることによってユーザを間接的に特定できることを保証するためのものである．また，ハンドルネームは成りすましを防ぐ必要はあるが，個人情報登録サーバに登録されている情報であればよく，また開示される必要がある．

8. 議　　論

この章では，類似技術との比較，今後の課題として解決すべき点について説明する．

8.1 他方式との比較

8.1.1 Kerberos

Kerberos そのものは，すべての principal が共通鍵を持つことを前提にした鍵配布，認証プロトコルである．Kerberos を本提案システムにそのまま適応することは次の点で困難である．まず，システム参加者全体をいくつかの Realm に分け Realm 間認証を行うための事前作業が必要である．また，WWW サーバはあらかじめ適切な KDC に principal を登録しておかなくてはならない．さらに提案システムの利用形態を考えれば，共通鍵を管理する KDC を Internet 上に公開する必要がある．様々な攻撃者が予想される Internet 上で KDC を安全に運用することの難しさはいうまでもないであろう．一方，Kerberos の PKI 拡張が標準化の議論の過程にあるが，特に，本提案システムのように，クライアントが共通鍵を利用し WWW サーバ，RS/TGS が公開鍵を利用するような方式は，議論も実装もされていない．

8.1.2 SSL の上のチケット転送

提案システムのように独自の認証プロトコルを利用する代わりに，SSL 上で単純にチケットの転送を行う方法も考えられる．つまり，チケットの保持証明をせずに SSL でチケットをクライアントから WWW サーバに転送する方法をとれば，簡潔なシステムが実現できる．しかしこの方法は，通信路上のチケットの盗聴に対しては有効であるが，その他の場所でのチケットの不正取得による不正アクセスを防ぐことはできない．特に WWW サーバにおけるチケットの盗難は，管理者や内部・外部のユーザの犯行を問わず，現実的な問題である．

8.1.3 SPKI とその応用システム

SPKI による方式は，すでに述べたように複数の権限証明書を公開鍵に関連付けているため，公開鍵そのものから，複数の権限情報を関連付けることによるユーザ追跡が可能である．SPKI やその応用システムにおいて，この点を克服するには，ユーザが新しいサービスに参加するたび，公開鍵証明書の発行を受ける必要があり，またそれに対応して権限証明書の発行を受ける必要があるが，これには，これまで述べたように公開鍵・秘密鍵生成の問題がある．したがって，公開鍵と SSL を利用するのであれば，2.1.1 項で説明した公開鍵による提案システムのように証明書の拡張領域にアクセス権限情報を入れる方式の方が，同じ問題をかかえるがより簡潔であるといった利点がある．

8.2 プロトコルの効率化

4.6 節で述べたように HTTP 上のチケット保持証明用のプロトコルには，1 往復分のオーバヘッドが存在する．画像などが多く貼られたページへのアクセスを考慮すれば，このプロトコルの効率化は今後の課題として解決しなければならない．また XML の処理の時間も無視できず，実装上の工夫が必要であろう．

8.3 個人情報登録サーバの運用問題

個人情報登録サーバには，いくつかの運用上の問題

をかかえている．

- ユーザの個人情報集中の問題
  個人情報登録サーバにすべての個人情報を登録することについては，プライバシ保護という点でユーザの敬遠が予想される．
- 負荷集中
  チケット発行のたびに，RS/TGS にアクセスする必要があるため，負荷が集中する．
- WWW サーバの信用調査
  WWW サーバの公開鍵証明書の有効性検証や信頼性の検証を実施する場合，個人情報登録サーバにかかる負荷が大きく，運用上問題がある．

最初の個人情報の集中の問題については，1 人のユーザが複数の個人情報登録サーバに分散して個人情報を登録するのが望ましいし，現実的である．具体的には，ISP ごとや組織ごとあるいはユーザの加入するサービスごとのような管理区分ごとに，情報登録サーバが設置されるのがよい．こういった運用には提案のシステムでも対応できるが，チケットによる認証とチケットの発行を統合したプロトコルを定めれば利便性が高まるだろう．

チケット発行の負荷集中に関しては，1 つの個人情報登録サーバの分散化が必要である．同様の分散化は Kerberos でも実装されてはいるが，提案システムは Internet というグローバルな環境での利用を前提としているため，特別な対応が要求される．

信用調査の負荷の問題への対策としては，キャッシュ利用による負荷の軽減などのシステムによる効率化のほか，専用の信用調査機関の設置による運用の軽減などが考えられる．

チケット発行と信用調査の負荷の問題については，今後の課題として検討していく必要がある．

8.4 鍵配布とスケーラビリティ

提案システムは，Kerberos の公開鍵拡張を参照し設計されたものであり，Kerberos にはすでに 8.1.1 項で触れたように鍵配送とスケーラビリティの問題がある．ところで，提案システムは，確かに共通鍵を利用するが，クライアントと WWW サーバ間は共通鍵は公開鍵で暗号化されたチケットとして配布され，また RS/TGS とクライアント間は SSL によって保護された通信路上で共通鍵およびチケットが配布される．したがって，提案システムにおいては，鍵配布やスケーラビリティの問題は，公開鍵証明書の配布と有効性確認の問題に帰着される．

具体的に考察を進める．まずユーザ数の増加に対しては，複数の情報登録サーバを導入することで対応が可能である．ユーザと情報登録サーバの間は SSL による相互認証を利用するので，共通鍵特有の問題を懸念する必要はない．また，情報登録サーバの数の増加や WWW サーバの数の増加については，それぞれの証明書がクライアントを介して交換されるので，鍵配布については問題はない．一方，有効性確認と信頼調査に関するスケーラビリティについては，情報登録サーバや WWW サーバの増加による負荷増大への対策の問題に帰着されるが，これについてはすでに 8.3 節で述べたように今後の課題としたい．

9. ま と め

本論文では，プライバシに配慮しながらユーザの個人情報によるアクセス制御システムを WWW 上に実現する方法について説明した．ユーザは，P3P の枠組みで個人情報の開示制御を行うことが可能であり，また開示レベルの導入により情報の開示をしなくても，情報の品質によりアクセス制御をすることが可能となる．また，本システムでは WWW サーバごとの事前登録が不要であり，ユーザの Single Sign On が実現した．今後は，課題で述べたようにプロトコルの効率化や個人情報登録サーバの問題の解決，および公開鍵を利用した提案システムなどについて検討していく．

謝辞 本研究は通信・放送機構の委託研究「情報通信不適正利用対策技術の研究開発」によるものである．ここに記して謝意を表す．また本研究を進めるにあたって，ご助言と協力をいただいた重本泰史氏，五十嵐信夫氏に感謝する．

参 考 文 献

1) ITU-T Recommendation X.509 (2000) — ISO/IEC 9594-8:2001, *Information technology — Open Systems Interconnection — The Directory: Authentication framework.*
2) World Wide Web Consortium (W3C). (http://www.w3.org)
3) Platform for Privacy Preferences (P3P) Project. (http://www.w3.org/P3P/)
4) Ellison, C.: SPKI Requirements, RFC2692 (1999).
5) Ellison, C., et al.: SPKI Certificate Theory, RFC2693 (1999).
6) Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms, *Comm. ACM*, Vol.24, No.2, pp.84–88 (1981).
7) Reed, M., Syversion, P. and Goldschag, D.: Anonymous connections and Onion Routing, *IEEE Selected Areas in Communications*,

Vol.16, No.4, pp.482–494 (1998).
8) Reiter, M. and Rubin, A.: Crowds: anonymous for web transactions, *ACM Trans. Information System Security*, Vol.1, No.1, pp.66–92 (1998).
9) Chaum, D.: Security without identification: Transaction systems to make big brother obsolte, *Comm. ACM*, Vol.28, No.10, pp.1030–1044 (1985).
10) Chaum, D. and Evertse, J.: A secure and privacy-protecting protocol for transmitting personal information between organizations, *CRYPTO'86*, Lecture Notes in Computer Science 263, pp.118–167, Springer-Verlag (1987).
11) Camenisch, J. and Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation, *EUROCRYPT 2001*, Lecture Notes in Computer Science 2045, pp.93–118, Springer-Verlag (2001).
12) Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M.: Private infromation retrieval, *36th Annual Symposium on Foundations of Computer Science*, pp.41–50 (1995).
13) Bao, F. and Deng, R.: Privacy Protection for Transactions of Digital Goods, *ICIC'01*, Lecture Notes in Computer Science 2229, pp.203–213, Springer-Verlag (2001).
14) Saito, T., Umesawa, K. and Okuno, H.: Privacy Enhanced Access Control by SPKI, *7th International Conference on Parallel and Distributed Systems: NGITA3*, Iwate, pp.301–306, IEEE (2000).
15) 梅澤健太郎，齋藤孝道，奥乃 博：プライバシーを重視したアクセス制御機構の提案，情報処理学会論文誌，Vol.42, No.8, pp.2067–2076 (2001).
16) Canovas, O. and Gomez, A.F.: AMBAR Protocol: Access Management Based on Authorization Reduction, *ICIC'01*, Lecture Notes in Computer Science 2229, pp.376–380, Springer-Verlag (2001).
17) Steiner, J.G., Neuman, B.C. and Schiller, J.I.: Kerberos: An Authentication Service for Open Network Systems, *Proc. Usenix Conference*, Dallas, Texas, pp.191–202 (1988).
18) Kohl, J. and Nueman, C.: The Kerberos Network Authentication Service (V5), RFC 1510 (1993).
19) Davis, D.: Kerberos Plus RSA for World Wide Web Security, *Proc. USENIX Workshop on Electonic Commerce* (1995).
20) IETF, Kerberos Working Group. (http://www.ietf.org/html.charters/krb-wg-charter.html)
21) Tung, B., et al.: Public Key Cryptography for Initial Authentication in Kerberos, Internet Draft (2001). (http://www.ietf.org/internet-drafts/draft-ietf-cat-kerberos-pk-init-15.txt)
22) Sirbu, M. and Chaung, J.: Distributed Authentication in Kerberos Using Public Key Cryptography, *Symposium on Network and Distributed System Security* (1997).
23) Medvinsky, A., Hur, M., Madvinsky, S. and Neuman, C.: Public Key Utilizing Tickets for Application Servers (PKTAPP), Internet Draft (2001). (http://www.ietf.org/internet-drafts/draft-ietf-cat-kerberos-pk-tapp-04.txt)
24) Rescorla, E.: HTTP over TLS, RFC 2818, Section 3, Endpoint Identification (2000).
25) Myers, M., Ankney, R., Malpani, A., Galperin, S. and Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol — OCSP, RFC 2560 (1999).
26) ISO/IEC 9798-2: 1999, *Information technology — Security techniques — Entity authentication — Part2: Mechanisms using symmetric encipherment algorithms.*
27) ISO/IEC 9798-4:1999, *Information technology — Security techniques — Entity authentication — Part4: Mechanisms using a cryptograhic check function.*
28) Menezes, A., Van Oorschot, P. and Vanstone, S.: *Handbook of Applied Cryptography*, chapter 10, pp.400–403, CRC Press (1997)
29) Krawczyk, H., Bellare, M. and Canetti, R.: HMAC: Keyed-Hashing for Message Authentication, RFC 2104 (1997).

(平成 13 年 12 月 4 日受付)
(平成 14 年 6 月 4 日採録)

本城 信輔（正会員）

1995 年東京大学大学院理学系研究科物理学専攻修士課程修了．1999 年（株）日立製作所入社．以来，システム開発研究所にてセキュリティ技術などの研究・開発に従事．

洲崎　誠一（正会員）

1991 年 3 月横浜国立大学電子情報工学科卒業．同年 4 月（株）日立製作所システム開発研究所に入所．以来，情報セキュリティ技術の研究開発に従事．2001 年 4 月横浜国立大学大学院環境情報学府入学．現在，同大学院環境情報学府博士課程後期に在学するとともに，システム開発研究所第 7 部（セキュリティシステム研究部）研究員．1996 年情報処理学会第 52 回全国大会優秀賞，平成 12 年度山下記念研究賞受賞．

齋藤　　司

1984 年早稲田大学教育学部数学専修卒業．1984 年（株）日立製作所入社．以来，公共システム事業部にて，官庁のシステム構築に従事．

三浦　信治

1998 年奈良先端科学技術大学院大学情報科学研究科情報処理学専攻博士前期課程修了．同年，日立公共システムエンジニアリング（株）入社．以来，同社ソリューション第 1 事業部にて，官公庁のシステム構築に従事．