2005－CSEC－31（4）
2005／12／9

ウェブログのためのアクセス制御機構の提案

花舘 蔵之　　長尾 伸二　　濱田 貴広　　永井 啓喜　　塩野入 理

NTT 情報流通プラットフォーム研究所
239-0847 神奈川県横須賀市光の丘 1-1

あらまし　本論文では，複数のウェブログプロバイダ (WP) がサービス提供する状況下で，任意の WP が管理するウェブログを用いて利用者同士が互いに記事を公開，閲覧するコミュニティ・アクセス制御方式を提案する．本方式では，(1) 各コミュニティメンバのウェブログ内に記事の閲覧権を表すトークンを格納し，(2) 公開者が閲覧を許可するトークンを記事に割り当て，(3) 記事閲覧時に閲覧者と公開者のウェブログが互いに同一トークンを保持していることを検証する．本方式は，認証機関のボトルネック化の回避，コミュニティメンバ更新時のスケイラビリティ確保，コミュニティメンバリスト共有時の安全性確保を満たす．また，本方式の実装を行った結果，500ms 以内で実行可能であることを確認した．

A proposal of the access control machanism for the Weblog

Masayuki HANADATE　　Shinji NAGAO　　Takahiro HAMADA
Hiroki NAGAI　　Osamu SHIONOIRI

NTT Information and Sharing Platform Laboratories
1-1 Hikarinooka Yokosuka-Shi Kanagawa, 239-0847, JAPAN

Abstract　In this paper, we propose an access control mechanism for Weblog groups, in which the Weblogs provided by multiple Weblog providers allow only same readers, those accepted by the members of the Weblog group, to download articles. This access control is realized as follows: (1) a token that represents the access right to download articles is stored in the Weblogs of every accepted reader, (2) each author associates one or more articles on his/her Weblog with the token, (3) the reader is allowed to download the article only if 2 Weblogs (i.e. the author's Weblog and the reader's Weblog) hold the same token. This technology removes the bottleneck of a separate authentication facility. It also provides excellent scalability in terms of updating the community member list among Weblogs, and enhanced security in sharing the community member list among Weblogs. We implement this technology and confirm that authentication is completed within 500ms.

1　はじめに

近年，ウェブ上で個人が様々な情報 (記事) を日記形式で公開するウェブログの利用者が増加している [1]．ウェブログとは，ウェブ上での情報公開で必要となる様々な維持管理を支援するソフトウェアを用いて，個人が様々な情報を公開するウェブサービスである．そのサービス提供形態は，利用者自身がサーバ上にウェブログを構築するユーザ管理型と，第三者 (WP:Weblog Provider) がサーバ上に複数のウェブログを構築し，利用者に統括的に提供するプロバイダ管理型がある．ここで WP は利用者を識別する ID(利用者 ID) も管理する．

ウェブログの情報配信方式はプル型情報配信方式であるため，記事を閲覧する利用者 (閲覧者) は，無関心なウェブログの記事閲覧を回避可能である．逆に，多くの記事が，記事を公開する利用者 (公開者) と無関係な不特定多数の閲覧者によって閲覧されて

いる．この事実は，不特定多数の閲覧者から情報を閲覧されたくない等 [2]，公開者のプライバシ保護上の理由からウェブログの導入障壁の 1 つとなっている．

公開者と無関係な不特定多数の閲覧者による閲覧を防止するためには，ウェブログのアクセス制御が必要である．プロバイダ管理型ウェブログのアクセス制御は，WP が管理する利用者 ID と記事の閲覧を許可する利用者 ID のリスト (閲覧者リスト) を用いて，(a) 記事と閲覧者リストを関連付ける認可設定，(b) 閲覧者が利用者 ID を保持する本人かどうかを検証する個人認証，(c) 閲覧者リストに利用者 ID が含まれることを検証する認可判定を行うことにより実現可能である．さらに，閲覧者リストに含まれる利用者のウェブログ間で閲覧者リストを共有する場合，閲覧者リストに含まれる利用者同士が記事の公開や閲覧を互いに行う情報共有を実現可能である．

本論文では，この閲覧者リストをウェブログ間で共有するアクセス制御をコミュニティ・アクセス制御と呼ぶ．その適用例として，ソーシャルネットワークサービス (SNS)[9, 10] が挙げられる，SNS では，全利用者がウェブログを保持し，利用者間の社会的関係等に基づく 1 つの閲覧者リストを，その閲覧者リストに含まれるウェブログ間で共有する．そして，閲覧者リストに登録されたウェブログを用いてコミュニティ内で情報共有を行う．

今，複数の WP が管理する様々なウェブログを閲覧する場合，1 人の利用者が複数の利用者 ID を管理すること，または，公開者と閲覧者が同一 WP に所属することが必要となり，利用者の利便性が低下する．そこで，本論文では複数の WP が管理する任意のウェブログを用いたコミュニティ型アクセス制御方式を検討する．ここで，(1) 異なる WP 間で利用者 ID を共有すること，(2) 異なる WP が管理するウェブログ間で閲覧者リストを共有することが求められる．利用者 ID の共有方法は，1 つの認証機関が異なる WP 間で共通的に利用できる利用者 ID を管理するシングルサインオン (SSO) 方式 [4, 5, 6, 7, 8] が提案されている．これらの方法では認証時に常に用いられる 1 つの認証機関がボトルネックとなる．また，閲覧者リストの更新時に閲覧者リストを共有するウェブログ数に比例して更新時間が増加する．さらに，閲覧者リストの共有時に閲覧者リストの偽造や改ざんを防止した安全な共有手段が必要となる．

本論文では，これらの要件を満たすために汎用的な権利流通基盤である FlexToken[3] を用いたコミュニティ・アクセス制御方式を提案する．提案方式ではコミュニティを構成する全てのウェブログに，情報共有される記事の閲覧権を表象する情報 (トークン) を格納する．また，ウェブログは記事とトークンの対応関係を管理する．記事の閲覧時に，閲覧者は最初に自分のウェブログにアクセスし本人認証を行う．この状態で，閲覧者が公開者のウェブログにアクセスすると，公開者と閲覧者のウェブログ間で互いに同一トークンを保持していることが検証される．本論文では，この同一トークンの検証で用いられるプロトコルをトークン照合プロトコルと呼ぶ．

提案方式では，複数の WP によって分散管理されたウェブログ同士が認証を行うため，認証機関のボトルネックは回避される．閲覧者リストの更新は，常に 1 つのウェブログにトークンを発行するのみであり一定時間で行われる．また，閲覧者リストの共有は，トークンの複製や改ざんを防止した安全なプロトコルを用いて行われる．

第 2 章では，ウェブログの概要とシステムの前提条件を示し，前提とするシステム構成でのアクセス制御の問題点を述べる．第 3 章では，FlexToken を用いたコミュニティ・アクセス制御方式を説明する．我々はトークン照合プロトコルを実装し処理時間を測定した．その測定結果を第 4 章で報告する．第 5 章では本方式の適用効果について考察し，本論文を第 6 章でまとめる．

2 ウェブログ

本章では，ウェブログとそのアクセス制御方式の概要を述べる．そして，ウェブログを用いた情報共有を行うコミュニティ・アクセス制御方式の問題点を述べる．

2.1 ウェブログの概要

ウェブログは，ウェブ上での情報公開で必要となる様々な維持管理 (e.g. HTML を用いたページ作成，ページ間のリンク階層管理，時系列に従った整列表示等) を自動的または効率的に行うウェブログ・アプリケーション (ブログウェア) を用いて，個人が様々な情報 (記事) をウェブ上で公開するウェブサービスである．ブログウェアは，利用者が記事を入力

する編集機能と，利用者が記事を閲覧する閲覧機能を有する．ブログウェアが搭載されるサーバと，ブログウェアに指示情報を送信し応答情報を表示するブラウザが搭載される端末は，同期通信路で接続されている．利用者は端末を用いてブログウェアに接続し，ブログウェアの各機能を利用する．本論文では編集機能を用いる利用者を公開者，閲覧機能を用いる利用者を閲覧者と呼ぶ．

利用者が編集機能を用いる時，ブログウェアは常に利用者の本人認証を行い，編集可能な 1 人の利用者であるかどうかを検証する．本論文では，利用者 U がブログウェア B の編集機能を利用可能であるとき,「U が B を保持 (所有) している」と呼ぶ．

ウェブログのサービス提供形態は，利用者自身が自分でブログウェアをサーバ上に構築するユーザ管理型と，公開者や閲覧者とは異なる第三者 (WP:ウェブログプロバイダ) が様々な利用者のブログウェアを 1 つのサーバ上で統括的に構築するプロバイダ管理型に分類される．プロバイダ管理型では，WP が様々な利用者を識別する ID(利用者 ID) を管理する．また，利用者によるプログラム改変やプログラム内の機密情報参照は不可能である．一方，ユーザ管理型では，利用者が自分のブログウェアに対してプログラム改変や機密情報参照を行える．本論文では，ブログウェアのアクセス制御を検討するにあたって安全性を考慮し，プロバイダ管理型ウェブログをサービス提供形態の前提条件とする．また，WP は複数存在することを前提条件とする．次章ではこの前提条件を用いたウェブログのアクセス制御について述べる．

2.2 ウェブログのアクセス制御

一般に，ウェブ上で公開される情報 (公開情報) のアクセス制御は，(1) 公開者が閲覧者リストと公開情報の組を設定する認可設定，(2) 閲覧者が利用者 ID を保持する本人自身であることを検証する個人認証，(3) 閲覧者リストに利用者 ID が含まれることを検証する認可判定から構成される．ここで，公開情報の閲覧を許可する利用者 ID のリストを閲覧者リストと呼ぶ．

このアクセス制御をプロバイダ管理型ウェブログに適用する．ある WP が利用者 ID$U1$ と $U2$ を管理している．ここで，$U1$ を公開者，$U2$ を閲覧者とする．公開者は，記事 V と $U2$ を含む閲覧者リスト π を作成し，公開者のブログウェア B は (V, π) を保持する．閲覧者が B にアクセスした際，B は $U2$ を用いて本人認証を行い，$U2 \in \pi$ を検証し，V を公開する．

ここで，閲覧者リストを複数のブログウェア間で共有することによって，閲覧者リストに含まれる利用者全員が相互に記事を公開，閲覧する情報共有を実現できる．本論文では，このアクセス制御をコミュニティ・アクセス制御と呼ぶ．この適用例としてソーシャルネットワークサービス (SNS) が挙げられる．SNS では，閲覧者リストを知人のウェブログ間で共有し，閲覧者リストに含まれる知人同士が互いにウェブログを用いた情報共有をする．例として，プロバイダ管理型ウェブログを用いる場合 [9] や，ユーザ管理型ウェブログを用いる場合 [10] が提案されている．

2.3 問題点

現在，複数の WS によるプロバイダ管理型ウェブログを用いた SNS は存在していない．複数の WP が存在する場合，利用者が複数の利用者 ID を管理，または，複数の利用者が同一 WP に加入しなければならないため，利便性が低下する．ここで，複数の WP が管理する任意のブログウェアを用いて情報共有可能なコミュニティ・アクセス制御を実現するためには，(1) 異なる WP 間で利用者 ID の共有，(2) 異なる WP に管理されたブログウェア間で閲覧者リストの共有が必要となる．

(1) 利用者 ID の共有はシングルサインオン方式 (SSO) 方式 [4, 6, 7, 8] を用いて実現可能である．しかし SSO 方式では常に利用者 ID を管理する 1 つの認証機関を用いて認証が行われるため，認証機関がボトルネックとなる．例えば，認証機関が障害によって停止すると全アクセス制御機能が停止する．

(2) 閲覧者リストの共有では，閲覧リストの共有時に，閲覧者リストの改ざん等の不正行為を防止した安全な共有方式が求められる．例えば，更新過程のリスト改ざんによる不正アクセスの結果，情報漏洩が発生する．また，閲覧者リストの更新時に，共有するブログウェア数に比例して更新時間が増加する．例えば，閲覧者リストが 100 個のブログウェアで管理されている時，閲覧者 1 名の追加で 100 個の閲覧者リストを同時に更新する必要がある．

本論文では，認証機関のボトルネック化の回避，閲覧者リスト更新のスケイラビリティ確保，閲覧者リスト共有時の安全性確保を行い，複数の WP に管

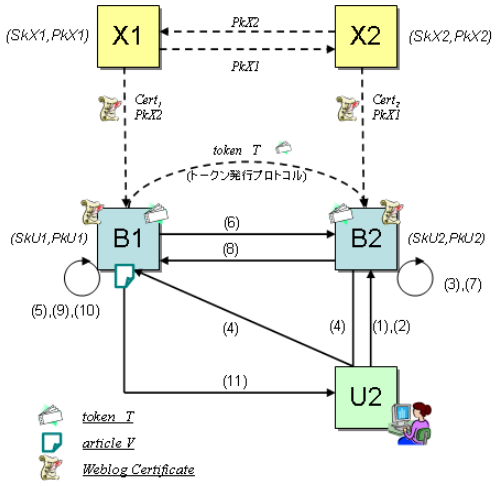

図 1: システム構成とトークン照合プロトコル手順

理される任意のウェブログを用いたコミュニティ・アクセス制御方式を提案する．

3 トークン照合プロトコル

本章では，FlexToken[3] を用いたコミュニティ・アクセス制御を提案する．まず，本章で用いる記法を定義する．次にシステム構成を説明する．そのシステム構成を用いてコミュニティ・アクセス制御とトークン照合プロトコルを説明する．

3.1 記法

本章の説明で用られる記法を定義する．公開鍵暗号方式における秘密鍵 Sk と公開鍵 Pk から構成される鍵ペアを (Sk, Pk) と記述する．公開鍵 Pk に対応する秘密鍵 Sk を用いてメッセージ msg の電子署名 sig を生成する電子署名生成関数 S を $sig = S_{Pk}(msg)$ と記述する．また，署名検証関数 V は，メッセージ msg の電子署名 sig が msg に対応する電子署名 $S_{Pk}(msg)$ である場合，真を返却し，$boolean = V_{Pk}(msg, sig)$ と記述される．ここで，$boolean$ は真偽値である．一方向ハッシュ関数 (e.g. SHA-1) を用いて生成されるメッセージ msg に対するハッシュ値を $H(msg)$ と記述する．

3.2 システム構成

図 1 に FlexToken[3] を用いたコミュニティ・アクセス制御のシステム構成図を示す．利用者を U_i と記述する．ここで，公開者を U_1，閲覧者を U_2 とする．U_i が保持するブログウェアを B_i と記述する．B_i を管理する WP を X_i と記述する．B_i は X_i が管理するサーバ上で動作する．U_i は端末を保持する．端末とサーバは通信路で接続されており，U_i は端末を用いてブログウェアにアクセスする．

提案方式では，記事の閲覧権を表象する一意性のある情報であるトークン [3] が用いられる．トークンの内容は任意の利用者により権利定義として自由に記述される．トークンは権利定義 ID，発行者 ID，トークンの枚数から構成される．権利定義 ID は，一方向ハッシュ関数 ($H(\cdot)$) 等を用いて権利定義 (R) に一意に対応つけられる情報 ($H(R)$) である，発行者 ID は権利定義を記述した利用者 ID である．

トークンはブログウェア上に格納され，通信路を介してコミュニティを構成するブログウェア間で移送または提示される．この過程でトークンは複製可能である．従って，複製トークンを格納したブログウェアを用いた不正な記事閲覧を防止しなければならない．そこで，提案方式では「正当な WP は複製トークンを格納不可能な正当なブログウェアを提供する」という前提条件を設ける (プロバイダ条件)．この条件を満たす WP が提供するブログウェアであることを証明するために，公開鍵証明書 (ブログ証明書) が用いられる．

次にブログ証明書の利用手順を説明する．まず，WP とブログウェアは公開鍵暗号方式の鍵ペアを保持する．WP は自分が管理するブログウェアにブログ証明書を発行する．また，WP は互いに信頼する WP 同士で公開鍵を交換し，自分が管理するブログウェアに交換した公開鍵を通知する．トークンの移送や提示の際，ブログウェアは，WP から通知された公開鍵を用いて相手のブログ証明書を検証する．これにより，相手のブログウェアのプログラムの正当性が検証される．

次章では，このシステム構成を用いて FlexToken を用いてコミュニティ・アクセス制御を説明する．

3.3 コミュニティ・アクセス制御

FlexToken 用いたコミュニティ・アクセス制御の概要を述べる．ブログウェアは，編集機能や閲覧機

(1)	U_2 は B_2 に本人認証を行い，B_2 が本人認証済状態となる．
(2)	U_2 は B_2 に (B_1, T) を送信する．
(3)	B_2 は $r1$ を生成し，$m1$ を保持する．
(4)	B_2 は U_2 に $m1$ を送信し，U_2 は B_1 に $m1$ を転送する．
(5)	B_1 は保持するトークン $(= T)$ を検証し，$m2$ を生成，保持する．
(6)	B_1 は B_2 に $m3$ を送信する．
(7)	B_2 は本人認証済状態の場合，$Cert_1$ と S_1 の署名検証と，$m1 \subset m2$ の検証を行う．
(8)	B_2 は B_1 に $m4$ を送信する．
(9)	B_1 は $Cert_2$ と S_2 の署名検証と，$m2$ を保持していることを検証する．
(10)	B_1 は T に対応する記事 $(= V)$ を特定する．
(11)	B_1 は U_2 に V を送信する．

表 1: トークン照合プロトコル処理手順

サーバ環境	(B_1)
CPU/メモリ	Pentium4 2.0GHz/1.0GB
OS	Linux(Turbo Linux 7)
httpd	Apache 1.3.29
言語	Perl 5.6.1
サーバ環境	(B_2)
CPU/メモリ	Pentium4 3.2GHz/2.0GB
OS	Linux (Fedra Core 4)
httpd	Apache 1.3.33
言語	Perl 5.8.6
端末環境	(U_2)
CPU/メモリ	Pentium4 3.4GHz/2.0GB
OS	Windows XP SP2
VM	JRE 1.4.2_09

表 2: 動作環境

能に加えて，トークン管理機能，トークン発行機能，トークン照合機能を有する．トークン管理機能はトークンを保持する機能である．トークン発行機能は，あるウェブログが生成したトークンを他のウェブログに複製されることなく安全に移送する機能である．トークン照合機能は，2 つのウェブログ間で同一トークンを保持していることを検証する機能である．トークン発行機能はトークン発行プロトコル [3] を用いて実現される．トークン照合機能は次章で述べるトークン照合プロトコルを用いて実現される．

次に処理手順を説明する．ここではトークンを T，記事を V と記述する．まず，U_1 は T を生成し，トークン管理機能を用いて B_1 は T を保持する．トークン発行機能を用いて U_1 は B_1 から B_2 に T を発行する．編集機能を用いて U_1 は V を生成し，B_1 は T と V の対応関係 (V, T) を保持する．U_2 は B_2 で本人認証を行い，トークン照合機能と閲覧機能を用いて B_2 上で B_1 を指定し V を閲覧する．次章ではトークン照合機能で用いられるトークン照合プロトコルの処理手順を説明する．

3.4 トークン照合プロトコル

各登場人物の初期状態を定義する．X_i は鍵ペア (SkX_i, PkX_i) を保持する．B_i はトークン T，鍵ペア (SkU_i, PkU_i)，ブログ証明書 $Cert_i = S_{PkX_i}(PkU_i)$，$X_i$ が信頼する WP の公開鍵 $PkX_i (i \in \{1, 2\})$ を保持する．B_1 は U_1 の記事 V を保持している．

U_2 が B_1 を閲覧する場合の処理手順を表 1 と図 1 に示す．ここで，各電文は $m1 = (B_1, B_2, T, r2)$，$m2 = (B_1, B_2, T, r1, r2)$，$m3 = (m2, s_1, Cert_1)$，$m4 = (m2, s_2, Cert_2)$ である．ただし，$r1$ と $r2$ を乱数，$s_1 = S_{PkU_1}(m2)$，$s_2 = S_{PkU_2}(m2)$ とする．

本論文ではトークン照合プロトコルの実現可能性を検証するために，実装を行い処理時間を測定した．次章では，その実装結果と処理時間の測定結果を述べる．

4 実装結果

本章では，トークン照合プロトコルの実装結果と性能測定結果を述べる．表 4 にシステム構成と動作環境を示す．システムは 2 台のサーバと 1 台の端末から構成される．各システムは LAN で接続され，HTTP をもちいた通信が行われる．各サーバ上では，トークン照合プロトコルを実行する CGI プログラムが動作する．CGI プログラムは，SHA-1(ハッシュ関数) と DSA(2048bit, 署名生成関数) を用いる．端末上は Java 言語を用いた処理時間測定プログラムが動作する．処理時間測定プログラムは端末上で処理 (2) から (11) までのターンアラウンド時間を測定する．平均処理時間は試行回数 50 回の平均値とする．測定の結果，平均処理時間は 492ms であることを確認した．

5 考察

本章では，トークン照合プロトコルのコミュニティ・アクセス制御への適用効果を考察する．トークン照合プロトコルをコミュニティ・アクセス制御に適用

することにより次の効果が得られる．

(1) 本方式では，複数の WP がブログウェアを分散管理する．仮に 1 つの WP が機能停止した場合，残りの WP を用いて情報共有を行うことが可能である．この点において，認証機関のボトルネックは解消されている．

(2) 本方式では，閲覧者の追加は 1 つのトークンを発行するだけであり，トークンを保持するブログウェア数に依存せず一定時間で行うことが可能である．従って，閲覧者リスト更新時のスケイラビリティが確保されている．

(3) 本方式における (a) ブログウェアの安全性と (b) 第三者攻撃からの安全性を考察する．

(a) ブログウェアは (i) プログラム自身が正しい動作をすること (プログラム動作の正当性)，(ii) プログラム改変や機密情報閲覧が困難であること (動作環境の正当性) を満たす必要がある．これらは，プロバイダ条件を満たす WP が提供するプロバイダ管理型ウェブログを用いて実現可能である．また，ブログウェアはブログ証明書を用いて通信相手の正当性を検証可能である．

(b) 第三者による攻撃として (i) 電文改ざん，(ii) リプレイ攻撃，(iii) 通信状態の盗聴が考えられる．電文改ざんは電子署名検証によって検出可能である．リプレイ攻撃について，ブログウエアはプロトコルの実行過程で一時的に乱数 (i.e. $r1$，$r2$) を保持する．ある処理の終了後，その処理で利用された電文を再利用しても，ブログウェアは乱数を破棄しているため電文の再利用は困難である．従ってリプレイ攻撃は防止されている．通信状態の盗聴について，HTTP を用いる場合，HTTP がステートレスのプロトコルであるため端末に通信状態 (e.g. 本人認証済み，トークン照合済み) を格納する必要がある (e.g. cookie)．この結果，端末上の通信状態を盗聴し再利用することによって不正アクセスを行える．この不正アクセスを防止するために，通信状態は乱数や有効期限の設定による一時的な値の使用が求められる．

6 まとめ

本論文では，様々な WP が管理する複数のウェブログを用いたコミュニティ・アクセス制御方式を実現するために，トークン照合プロトコルを用いたアクセス制御方式を提案した．提案方式では，閲覧権を表象する 1 つのトークンを共有した複数のウェブログ間で，互いに記事を公開，閲覧することが可能となる．提案方式により，(1) 認証機関のボトルネック回避，(2) 閲覧者リスト更新時のスケイラビリティ確保，(3) 閲覧者リスト共有時の安全性確保を行うことが可能である．また，提案方式は 500ms 以下という実用的な処理時間内で実行可能であることを確認した．

今後は，トークン発行の実装やリンク先参照時のトークン照合を行い，安全かつ利便性の高いウェブログを用いたグローバルな P2P ソーシャルネットワークサービスの構築を目指す．

参考文献

[1] 総務省, “ブログ・SNS の現状分析及び未来予測”, http://www.soumu.go.jp/s-news/2005/050517_3.html

[2] IRI コマース&テクノロジー/サーベイリサーチセンター編著, “ブログビジネス白書 2005”,2005.

[3] Terada, M., Iguchi, M., Hanadate, M. and Fujimura, K., “Copy prevention scheme for rights trading infrastructure”, *4th Smart Card Research and Advanced Application IFIP Conference(Cardis2000)*,2000.

[4] Liberty Alliance Project, http://www.projectliberty.org/

[5] TypeKey,http://www.sixapart.com/typekey/

[6] OpenID,http://www.openid.net

[7] LID,http://lid.netmesh.org

[8] The YADIS Project, “YADIS-Yet Another Decentralized Identity Interoperability System”,http://www.yadis.org,2005.

[9] mixi,http://mixi.jp/

[10] afflio,http://affelio.jp/