Web サービスにおける不正ユーザを排除可能な匿名認証システムの実装

堀地 恭輔 †　　中西 透 †　　舩曳 信生 †

† 岡山大学大学院自然科学研究科
700-8530 岡山県岡山市北区津島中 3-1-1

horichi@sec.cne.okayama-u.ac.jp, {nakanisi, funabiki}@cne.okayama-u.ac.jp

あらまし　グループ署名を用いた匿名認証では，不正ユーザの排除はグループ管理者を介して行われるため，グループ管理者には不正ユーザの特定が可能であり，ユーザは完全な匿名性を享受できない．それに対して，ブラックリストを用いることによりグループ管理者なしで不正ユーザを排除可能な匿名認証方式が提案されている．しかし，この方式は Web システムにおいて実装されていない．そこで本研究では，Web サービスにおける不正ユーザを排除可能な匿名認証システムの実装を行った．この匿名認証システムでは，既存のシステム変更を最小限に抑えるため，プロキシを用いている．そして，評価実験により，実装した匿名認証システムの処理時間を測定し，小規模なグループなら実用的な時間で利用が可能なことを確認した．

An Implementation of an Anonymous Authentication System with Exclusion of Illegal Users for Web Services

Kyosuke Horichi†　　Toru Nakanishi†　　Nobuo Funabiki†

†Graduate School of Natural Science and Technology, Okayama Univercity
3–1–1 Tsushima-naka, Kita-ku, Okayama, Okayama 700-8530, Japan

horichi@sec.cne.okayama-u.ac.jp, {nakanisi, funabiki}@cne.okayama-u.ac.jp

Abstract　In an anonymous authentication using the group signature, the group manager can identify the illegal user, and thus, the user cannot enjoy the complete anonymity. On the other hand, an anonymous authentication has been proposed, where an illegal user can be excluded without the group manager using a blacklist. Unfortunately, this scheme has not been implemented as a Web system. In this study, we implemented the anonymous authentication system using the blacklist for Web services, where proxies were used to avoid modifications of existing Web servers and browsers. In addition, we measured the processing time of the system, and confirmed the practical time for small groups.

1　はじめに

現在，提供されている多くの Web サービスでは，不正アクセスを防ぐためにユーザ認証を行っているが，通常のユーザ認証ではサービス提供者側にユーザの利用履歴が蓄積してしまい，プライバシー問題が発生しうる．この問題の解決策として，グループ署名を用いた匿名認証が提案されている．この匿名認証では，ユーザがあるサービスを受けたい場合，そのサービスが属しているグループにユーザ登録を行う．そして，グループ署名を用いることによって，サービス提供者への認証を匿名で行うことができる．

その一方で，匿名のユーザが不正を行った場合に，グループ管理者がその不正ユーザを特定し，ユーザ権限を失効し排除する．このことは，グループ管理者は常にユーザの状態把握が可能であり，ユーザは完全な匿名性を享受できないことを意味する．

それに対して文献 [2] では，ブラックリストを利用することにより，グループ管理者を利用することなく不正ユーザを排除可能な匿名認証方式が提案されている．この方式では，サービス提供者が不正ユーザのグループ署名に含まれるタグと呼ばれる情報からブラックリストを作成する．ユーザは認証時にブラックリストに自身のタグが含まれていないことを匿名で示すことで，匿名性を維持したまま不正ユーザでないことの確認を行う．しかしこの方式は Web システムにおいて実装されていない．

本研究では，文献 [1] で提案している匿名認証システムを修正することで，文献 [2] の匿名認証方式を Web システムとして実装する．実装した匿名認証システムではブラウザ，Web サーバとは別にプロキシを導入し，ユーザ･サーバのプロキシ間で認証を行う．ブラックリストを管理するためにアプリケーション側でタグを管理する必要があるため，プロキシ中の認証データ検証処理中にタグを抜き出し Web サーバへ HTTP により転送する処理を組み込む．また，ブラックリスト管理のプロトコルはアプリケーションに依存するため，今回は匿名掲示板を Web サービスとして実装した．

本研究では，実装した匿名認証プロトコル，ブラックリスト管理プロトコルの動作確認を行うとともに，その処理時間を測定することにより評価を行った．認証プロトコルは，ブラックリストの人数が 200 人以下の場合は，約 3.7 秒で処理が完了することを確認した．しかし，ブラックリストサイズに比例して認証時間が増加し，800 人では認証に約 13 秒要する．但し、ブラックリスト管理プロトコルは，ブラックリストサイズに依存せず，約 0.13 秒で処理が完了することを確認しており、十分な実用性を有していると考えられる．

2 準備

2.1 グループ署名を用いた匿名認証とその問題点

グループ署名とは匿名性を持つディジタル署名技術である．グループ署名を用いることにより，ユーザはサービス提供者に個人情報を秘匿したまま，正当なユーザであることを示せる．グループ署名は以下の性質を満たす．

- 偽造不能性：ある特定のグループのメンバのみが署名を作成できる．
- 匿名性：通常の検証者は，どのメンバが署名を作成したのか判断できない．
- リンク不能性：通常の検証者は，任意の 2 つの署名が同じ者により作成されたかどうかを判断できない．
- 追跡可能性：何か不正が発生した際に，グループ管理者のみは，署名の作成者を特定できる．

グループ署名においてグループ管理者 (GM) と呼ばれる機関が必要である．グループ管理者はユーザがグループのメンバに加入することを許可する権限を有する．

もし，何かの不正が発生した場合には，グループ署名の追跡可能性により，グループ管理者はその不正を行ったユーザを特定し，ユーザ権限を失効し排除することができる．しかし，このことはグループ管理者は常にユーザの状態把握が可能であり，ユーザは完全な匿名性を享受できないことを意味する．また，不正者のユーザ権限の失効確認をグループ管理者により作成された失効リストを用いて行っているため，ユーザ認証の際に，常に認証サーバとグループ管理者との通信が必要となる．この通信は，匿名認証の処理時間にオーバーヘッドをもたらすため，実用に際しての問題となる．

2.2 グループ管理者を利用することなく不正ユーザを排除可能な匿名認証方式

ブラックリストを利用することにより，グループ管理者を利用することなく不正ユーザを排除可能な匿名認証方式が [2] で提案されている．ま

ず、本方式で利用される知識の署名を説明する．知識の署名を SPK(Signature based on proof of knowledge) と記述する．SPK は対話型零知識証明をハッシュ関数を用い，非対話型に変更した零知識証明のことである．例として，離散対数 x を知ることを証明する，メッセージ $M \in \{0,1\}^*$ に対する SPK は以下のように記述する．

$SPK\{(x) : y = g^x\}(M)$

ブラックリストはサービスごとにサービス提供者が作成する．各認証においてユーザが送付するデータにはタグと呼ばれるデータが付加されている．ブラックリストには，不正行為が行われた認証中のタグが含まれている．このタグは匿名性を満たす必要がある．そして，ユーザは認証において，ブラックリスト中のタグが自身のものでないことを SPK により匿名で証明する．不正ユーザはこの証明を行えないため，排除される．

準備

双線形写像 e を持つ素数位数 p の群 $\mathbb{G}_1$,$\mathbb{G}_2$ を選択する．また，DDH 仮定を満たす素数位数 p の群 $\mathbb{G}$ も同時に選択する．さらに，$\mathbb{H}_0$ を $\mathbb{G}$，$\mathbb{H}$ を $\mathbb{Z}_p$ 上の値を出力するハッシュ関数とする．ランダムに $g_0, g_1, g_2 \in \mathbb{G}_1$ と $h_0 \in \mathbb{G}_2$ を選択し，$params = (p, \mathbb{G}_1, \mathbb{G}_2, e, \mathbb{H}_0, \mathbb{H}, g_0, g_1, g_2, h_0)$ として出力する．

鍵生成

GM はランダムに $\gamma \in_R \mathbb{Z}_p$ を選び，$\omega = h_0^\gamma$ を計算する．γ が GM の秘密鍵となり，ω が公開鍵となる．

メンバ登録

1. ユーザは GM に以下の SPK を計算し送る．ここで $(x, y') \in_R \mathbb{Z}_p^2$，$\epsilon$ は空のメッセージである．
$$SPK\left\{(x, y') : C = g_1^x g_2^{y'}\right\}(\epsilon)$$
2. GM は SPK が間違っていれば $invalid$ を返し終了し，正しければユーザに (A, e, y'') を送る．ここで $A = (g_0 C g_2^{y''})^{\frac{1}{e+\gamma}} \in \mathbb{G}_1$ である．
3. ユーザは $y = y' + y''$ を計算する．$\hat{e}(A, \omega h_0^e) = \hat{e}(g_0 g_1^x g_2^y, h_0)$ ならば (A, e, x, y) を出力し完了する．

ブラックリスト管理

検証者であるサービス提供者 (SP) は，不正ユーザの認証データのタグからブラックリスト BL を作成する．

匿名認証

1. SP は (BL, m) をユーザに送る．$m \in_R \{0,1\}^\ell$ はランダムチャレンジ，$BL =< \tau_1, ..., \tau_n >$ は SP の最新のブラックリストであり，$\tau_i = (s_i, t_i) \in \{0,1\}^\ell \times G$ はブラックリストの中の i 番目のタグである．
2. ユーザは $b_i = H_0(s_i || SP)$ を $i = 0$〜n に対し計算する．もし，ある i について $t_i = b_i^x$ のとき，停止する．ここで SP は SP の ID とする．
3. ユーザは SP に (τ, Π_2) を返す．タグ $\tau = (s, t) \in \{0,1\}^\ell \times G$ はランダムに $s \in_R \{0,1\}^\ell$ を選び，$b = H_0(s||SP)$ とし，$t = b^x$ として計算される．Π_2 は次に示す SPK である：
$$SPK\left\{(A, e, x, y) : A^{e+\gamma} = g_0 g_1^x g_2^y \wedge \left(\bigwedge_{i=1}^{n} t_i \neq b_i^x\right) \wedge t = b^x\right\}(m)$$
4. もし Π_2 が無効なら，SP は"failure"とし，そうでなければ"success"とする．

2.3 プロキシを用いた匿名認証システム

本研究グループでは，プロキシを用いた匿名認証システム [1] を提案・実装している．このシステムの概要を以下に示す．グループ署名を用いた匿名認証システムでは，ユーザ側，サーバ側ともに匿名認証を行う際に独自の暗号処理が必要となる．本システムでは，この暗号処理部分をプロキシを用いて実装することにより，ブラウザや Web サーバは既存のシステムを変更すること無く匿名認証システムを利用することができる．本システムでは，ユーザ，サービス提供者それぞれにプロキシをおいている．この 2 つのプロキシ間では匿名認証の際，第三者からの不正傍受や，サーバのなりすましを防ぐため，TLS によるサーバ認証と暗号化を利用する．この暗号通信路を使って，独自の匿名認証プロトコルによるクライアント認証を行う．

今回実装した匿名認証システムでは，文献 [2] に基づいた匿名認証プロトコルをプロキシベースの匿名認証システムへ追加実装をしている．

3 不正ユーザを排除可能な匿名認証プロトコルの実装

この章では，実装した匿名認証プロトコルを示す．

3.1 実装システムのプロトコル

本研究の認証プロトコルは，文献 [2] の認証方式を基に作成した．以下に認証プロトコルの詳細を示す．

1. CllientHello メッセージにより，ユーザはサーバに対して認証要求を行う．
2. SendBlackList メッセージにより，サーバは乱数チャレンジとブラックリストをユーザに送信する．
3. ユーザは秘密鍵により，自分のタグがブラックリストにないことを証明し，認証データを生成する．
4. SendSPK メッセージにより，認証データをサーバに送信する．
5. サーバは受け取った認証データを検証する．
6. SendSuccess メッセージにより，認証が正しく終了したことを知らせる．

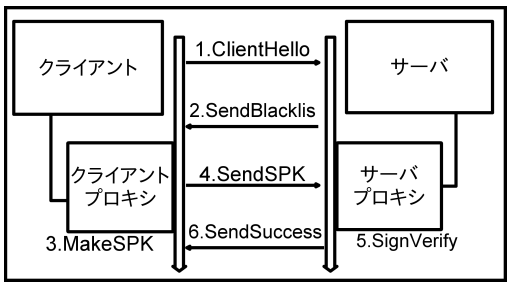

図 1: 認証プロトコル

3.2 実装の要件

本プロトコルを実装するにあたり，以下の 2 つの要件が必要となる．

1. グループ署名の暗号処理は，処理の高速化と既存のグループ署名で用いる楕円曲線暗号ライブラリが C 言語であることから，C 言語で行う．
2. 第三者に傍受されるとユーザの秘密情報等の匿名性が損なわれる情報を扱うため，サーバ認証に基づいた暗号化通信を行う必要がある．

3.3 実装

ユーザ側，グループ管理者側ともに Java を用いてプロキシとしてブラウザ･サーバの外側へ実装した．Java により実装を行ったのは Java が豊富なネットワーク API を持つためである．このとき，要件 1 を満たすために JNI(Java Native Interface) を用いて実装した．JNI を経由することで C 言語のライブラリを Java で利用することができる．また，要件 2 を満たすために暗号通信路 SSL/TLS を用いて実装を行った．

4 ブラックリスト管理プロトコルの実装

本章では，不正ユーザを排除可能な匿名認証システムに対する，ブラックリスト管理プロトコルの実装について示す．ここで，ブラックリスト管理とは不正ユーザのブラックリスト登録，解除を行い，不正ユーザをデータベースで管理することである．

4.1 ブラックリスト管理プロトコルの実装の要件

サービス提供者によるブラックリストの管理はアプリケーションに依存するため，Web アプリケーションとして匿名掲示板を採用し，それに基づいて実装した．匿名掲示板を採用した理由としては以下が挙げられる．

- 評価対象として作成が容易である．
- グループ署名を用いて匿名化することにより，自由な発言が期待できる．

このとき，以下の 2 つが必要となる．

1. プロキシ間での認証時にサーバ側でのタグの抜き出し
   今回実装した匿名認証システムでは，ブラックリストにタグを記載し，アクセスしてきたユーザのタグと比較を行うことで認証を行っている．サーバ側で不正ユーザのタグをブラックリストに登録するため，認証時にタグを抜き出し保持しておく必要がある．実装したシステムでは，グループ署名の暗号処理をプロキシを用いて行っているため，プロキシでタグを抜き出す．

2. プロキシから Web サーバへのタグの転送
   ブラックリストの管理は，Web サーバ上の掲示板アプリケーションで行う．このため掲示板アプリケーションはプロキシからタグを受け取りデータベースで管理する必要がある．ユーザが記事投稿を行った際の HTTP リクエストにタグを埋め込んで，Web サーバに渡すようにする．

4.2 ブラックリストの管理

掲示板は，データベース上の掲示板テーブルで全記事を管理している．そこで，ブラックリストも同一データベースのブラックリストテーブルで管理を行う．このブラックリストテーブルには，ブラックリストとして選択された記事が保存される．また，ブラックリストテーブルからタグのみをファイル出力し，このファイルを認証時にブラックリストとして使用する．

ブラックリスト登録機能では，サービス提供者は Web サーバにアクセスし，ブラックリスト登録フォームを取得する．このフォームでは，投稿されている全記事がラジオボタンとともに表示される．これらの中から不正な記事を選択することによって，ブラックリストに，選択された記事のタグを追加更新する．この時，ブラックリスト登録期限（無期限，１週間，１ヶ月）を選択し，データベースに追加、保存する．以降の認証ではこの更新されたブラックリストが使用される．

ブラックリスト解除機能では，同様にブラックリスト解除フォームを取得する．このフォームでは，ブラックリストに登録されている全記事がラジオボタンとともに表示される．これらの中から解除する記事を選択することによって，ブラックリストから，選択されたタグを解除する．また，サーバ側で一定時間毎にブラックリストをチェックし，ブラックリスト登録の期限が切れたユーザの解除を自動的に行う．

図 2: ブラックリスト管理用フォーム（登録）

4.3 実装

匿名認証で用いている暗号処理が C 言語で実装されているため，タグの抜き出し処理は C 言語により実装している．認証データを検証する処理において，タグをサーバ側プロキシで抜き出す．ユーザが記事投稿を行った場合，HTTP リクエストがプロキシを経由して掲示板アプリケーションが動作する Web サーバに送られる．サーバ側プロキシにおいて，この HTTP リクエスト内に署名データと分離してタグも埋め込み，Web サーバに渡すようにした．

匿名掲示板はブラウザに依存すること無く閲覧を可能にするために JSP，Java サーブレットを用いて実装している．投稿された記事は MySQL を用いてデータベースに保存される．そこで，ユーザが記事を投稿した場合，認証時にプロキシから渡されるタグを，記事と一緒に掲示板テーブルに保存するように実装した．

次に，ブラックリスト管理機能について述べる．ブラックリスト登録機能に関しては、サーバでは掲示板テーブルから，選択された記事とタグを抽出し，ブラックリストテーブルにコピーする．そして，ブラックリストテーブルから，タグのみを抽出しブラックリストファイルへ出力を行っている．ブラックリスト解除機能に関

しては，ブラックリストテーブルから，選択された記事のタグを抽出し，ブラックリストファイルから，一致したユーザのタグを除去し，ブラックリストテーブルからも削除する．また，サーバ側で一定時間毎にブラックリストに登録されたユーザのチェックを行い，期限が切れているユーザがいる場合，ブラックリストから解除する．

5　実験・評価

5.1　実験環境

実験環境として，クライアント PC とサービス提供者用 Web サーバ PC を準備し，学内 LAN 環境に接続した．図 3 に各 PC の性能を示す．

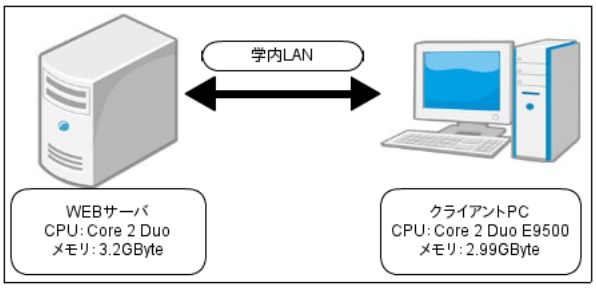

図 3: 実験環境

5.2　測定結果

匿名認証システムの評価として，認証に要する処理時間の測定を行った．図 4 に認証時間を示す．図 4 で示すように，ブラックリストサイズの増加に伴い，認証時間も線形的に増加する．これはブラックリストに記載されている全てのタグと比較し，ゼロ知識証明を行っているためである．ブラックリストの人数が 200 人程度なら数秒で認証できていることから，小規模なグループでは実用的であることが確認できる．

次に，ブラックリスト登録処理と解除処理に要する処理時間の測定を行った．その結果，ともに約 0.13 秒で処理が完了することを確認した．このことより，ブラックリスト管理プロトコルは実用的な時間で動作することを確認できた．

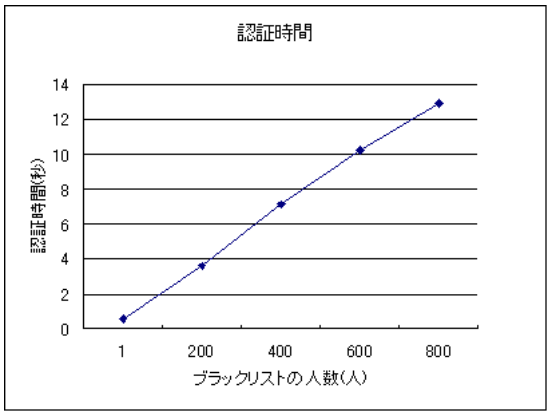

図 4: 認証時間

6　おわりに

本研究では，グループ管理者を利用することなく不正ユーザを排除可能な匿名認証システムの実装を行った．そして動作確認を行うとともに，その処理時間を測定することにより評価を行った．認証時間はブラックリストサイズが 200 人程度の少人数であれば，実用的な時間で動作可能である．ブラックリスト登録・解除処理は約 0.13 秒で処理が完了することから，ブラックリスト管理プロトコルは十分な実用性を有しているといえる．

今後の課題としては，多人数での認証時間の削減や本匿名認証システムの Wikipedia などの他のアプリケーションへの応用が考えられる．

謝辞　本研究は科研費 (22560378) の助成を受けて行われた．

参考文献

[1] 大林弘樹，中西透，舩曳信生，”Web サービスにおけるプロキシを用いた匿名認証システムの実装”，コンピュータセキュリティシンポジウム 2008(CSS2008)，pp.801-806，2008-10.

[2] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, ”Blacklistable Anonymous Credentials : Blocking Misbehaving Users without TTPs”, CCS 2007, pp.72-81, 2007.