匿名属性認証システムの Android 携帯端末への実装

三嶋 徹 †　　中西 透 ‡　　渡邊 寛 ‡　　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
†mishima@sec.cne.okayama-u.ac.jp, ‡{nakanisi, can, funabiki}@cne.okayama-u.ac.jp

あらまし Web サービスにおける ID とパスワードを用いたユーザ認証では，サービス提供者が ID を通じてユーザの利用履歴の蓄積ができてしまう問題がある．一方，サービス提供者はユーザが持つ属性のみを用いて認証を行いたい場合も多い．この問題の解決策として，匿名属性認証システムが提案されている．本研究では，PC 向けに実装されている匿名属性認証システムを Android に移植し，Web アプリケーションと連携させた．実機での評価実験の結果，ユーザ認証に要する時間は約 1 秒と，属性数に依存せず一定であり，十分な実用性を持つことを確認できた．

An Implementation of Anonymous Attributes Authentication System on Android Smart Phone

Toru Mishima †　　Toru Nakanishi‡　　Kan Watanabe‡　　Nobuo Funabiki‡

†‡Graduate School of Natural Science and Technology, Okayama University
3-1-1 Tsushima-naka, Kita-ku, Okayama 700-8530 JAPAN
†mishima@sec.cne.okayama-u.ac.jp, ‡{nakanisi, can, funabiki}@cne.okayama-u.ac.jp

Abstract In the current Web services using ID-based Authentications, service providers can create user's profile via user's ID. This may cause serious privacy problem. On the other hand, lots of service providers want to obtain user's attributes instead of user ID. To solve this problem, an anonymous attribute authentication system was proposed. In this paper, we implemented that system on Android devices, which is work in cooperation with Web services. From the experiments on a real tablet device, the authentication time is 1 second, and does not depend on the number of attributes. Thus, we confirm that our system is sufficiently practical.

1　まえがき

Web サービスで多く利用されているユーザ認証では，正規ユーザであることを確認するために，サービス提供者に対してユーザ ID とパスワードを用いた認証を行った後，サービスを開始する．このようなユーザ認証では，ユーザ ID を通じてサーバがユーザの利用履歴を蓄積可能である．また，サービスを提供する際に収集した性別や，生年月日，職業といったユーザの属性をもとに，ユーザのプロファイルを生成できてしまい，重大なプライバシ問題が発生しうる．

一方，サービス提供者はユーザの持つ属性をユーザ認証のために取得したい場合も多い．そのため，本研究グループでは，ユーザの属性をもとに匿名で認証を行う，匿名属性認証システム [1] を提案している．匿名属性認証とは，個人を特定できる情報を用いずに，ユーザがサービス提供者の要求する属性を所持していることを証明する認証方式である．この証明では，指

定された属性を持つことをサービス提供者は検証可能であるが，どのユーザが作成したかを特定できない．このため，匿名属性認証システムを用いることにより，プライバシを保護しながら正確な属性の取得が行える．

本研究グループでは，先行研究として，[1] の匿名属性認証システムを，プロキシを用いた匿名認証システム [2] に組み込んだ，匿名属性認証システムのアンケートサービスへの応用 [3] を提案している．[2] のシステムでは，認証機能を Web サービスに組み込むために，クライアントとサーバの双方にプロキシサーバを配置し，プロキシ同士で認証処理を行う．したがって，Web サーバおよび Web ブラウザに必要な変更が少なく，導入が容易である．また，これらの [1]，[2]，[3] のシステムは，Windows または Linux を搭載した PC を対象に実装されている．

近年，スマートフォンに代表される携帯型多機能端末が普及しつつある．スマートフォンは，PC の環境に近い機能を備えており，PC とスマートフォン・タブレットの両方に対応している Web サービスが増加している．

そこで本研究では，[1] のシステムを，スマートフォンおよびタブレットの代表的な OS の一つである Android に実装し，プロキシを用いた匿名認証システム [2] に対応させることにより，[1] のシステムで提供される認証方式をもつ Web サービスと連携させる．Android ではプロキシが使用できないことから，実装の方針として，[1] の匿名属性認証システムの認証アルゴリズムを Android に実装した上で，ユーザが利用するプロキシを認証用アプリケーションに置き換える．実装したシステムでは，認証に必要な手続きはすべて，認証用アプリケーションとサーバプロキシ間で行い，現行のブラウザをそのまま利用することができる．

実装システムの評価として，Android2.3.3 搭載のスマートフォンである HTC Desire HD および Android4.0.3 搭載のタブレット端末である Sony Tablet S を用いて，属性数を変えながら，認証用のアプリケーションが起動してから認証が完了するまでの時間と，認証データの生成時間を計測した．その結果，どちらも属性数に依存せず一定処理時間となった．認証時間は約 1 秒，認証データ生成時間は約 0.3 秒であり，実用レベルの処理時間であると考えられる．

以降，2 章では先行研究と実装に利用する技術を示す．3 章では Android の実装の詳細について述べ，4 章で実験および評価結果を示す．

2 先行研究と利用する技術

2.1 匿名属性認証

匿名属性認証とは，個人を特定できる情報を用いずに，ユーザはサービス提供者が要求する属性を所持していることを証明する認証方式である．匿名属性認証おける属性とは，ユーザが共通して持つ性質や特徴の情報であり，国籍，性別，所属といったものが挙げられる．

[1] のシステムにおける匿名属性認証では，以下の 2 つの関係の証明を行える．

- AND relation
  指定された複数の属性すべてをユーザが所持すること
- OR relation
  指定された複数の属性のうち 1 つをユーザが所持すること

例えば，AND relation では，ユーザが学生かつ特定の学部に所属しており，在学中であることを学部棟に入るために証明できる．OR relation では，コピー機を使用する際に，スタッフか教員のどちらかであることを証明できる．AND・OR relation のどちらも，証明コストは属性数に依存せず一定である．

2.2 匿名属性認証システムのプロトコル

図 1 に [1] の匿名属性認証システムの概要を示す．

このシステムにおける認証プロトコルについて述べる．まず，証明書発行機関が認証データ作成および検証用の公開鍵を配布する．次に，ユーザは証明書発行機関に対してユーザ登録を行い，属性証明書を受け取る．ユーザ認証に必

要なデータとして，属性カテゴリと属性値がある．属性カテゴリとは，属性の種類を表す．属性カテゴリの例として，国籍，性別，所属などがある．また，属性値とは，ユーザが持つ具体的な値のことを示す．例えば，属性カテゴリ「国籍」に対して，属性値は「日本」「中国」「アメリカ」などの値を持ちうる．ユーザ認証では，ユーザと検証者の間で事前に共有した乱数メッセージ，属性証明書，証明書発行機関の公開鍵，属性カテゴリ，属性値を用いて，ユーザが認証データを作成し，検証者に送信する．検証者は，乱数メッセージ，証明書発行機関の公開鍵，属性カテゴリ，属性値を基に認証データの検証を行う．

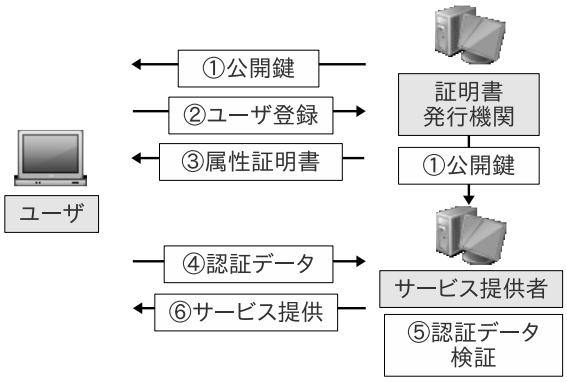

図 1: 匿名属性認証システムの概要

2.3 プロキシを用いた匿名認証システム

図 2 に，本研究グループで提案している，プロキシを用いた匿名認証システムの概要を示す．

このシステムは，クライアント PC とサーバ PC 双方に，匿名認証システムを実装したプロキシを配置することによって，匿名認証システムと Web サービスを連携させる．特徴として，認証に必要なすべての手続きをプロキシ間のみで行うため，既存の Web システムを変更することなく匿名認証を導入することができる．つまり，ブラウザや Web サーバにペアリングなどの特殊な暗号処理機能を組み込む必要がないため，現行のブラウザや Web サーバを変更することなく利用することが可能である．本システムではクライアント (ユーザ) とサーバ (サービス提供者) の両方にプロキシを配置しており，この 2 つのプロキシ間の通信では，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い，その後，HTTP のリクエストとレスポンスの転送を行う．

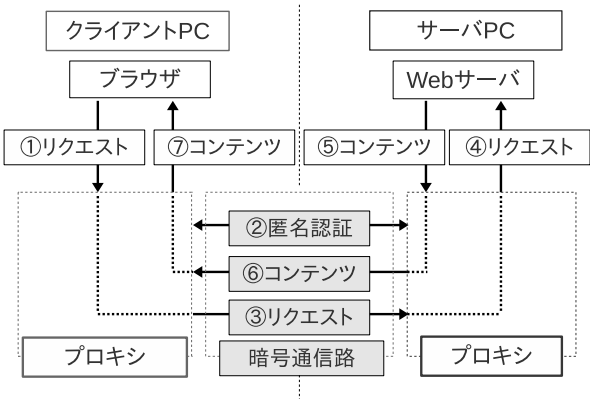

図 2: プロキシを用いた匿名認証システムの概要

プロキシを用いた匿名認証システムの実装として，[3] では，[1] の匿名属性認証システムをプロキシを用いたシステムに組み込み，アンケートサービスに応用している．[3] のアンケートサービスでは，ユーザに送信してもよい属性カテゴリを選択させ，その属性を用いて匿名認証を行い，Web でのアンケートを行う．その際，匿名ではあるがユーザは自身の属性を偽ることができないため，サービス提供者にとって，より信頼性の高い属性情報に基づいたアンケート結果を得られる．

2.4 利用する技術

本研究においてシステムの実装に利用した技術および関連事項について述べる．

2.4.1 Android

Android とは，2007 年 11 月に Google 社が発表した，携帯型端末向けのプラットフォームである．アプリケーションの開発言語は Java 言語であり，Java SE 準拠の API が用意されているが，使用できない API 群もある．また，C 言語や C++言語で作成したネイティブコードも

JNI(Java Native Interface) を通じて実行することができる．ネイティブコードを用いることにより，プログラムがCPUアーキテクチャに依存するというデメリットがあるものの，高速化や既存のCライブラリの再利用が可能である．

2.4.2 droid-wrapper[4]

C言語及びC++言語で記述されたネイティブライブラリをビルドするための，Androidのツールチェイン用ラッパである．コンパイラのラッパとしてdroid-gcc，droid-g++，リンカのラッパとしてdroid-ldを利用できる．ラッパは，複雑な仕様を隠蔽して，操作を単純にする役割を果たす．

Android用にネイティブプログラムをコンパイルするためには多くのオプションを指定する必要がある．しかし，configureおよびlibtoolのようなビルド支援ツールを使用しているプログラムの場合，Android標準のネイティブ開発ツールキットであるAndroid NDK[5]ではコンパイラとリンカにオプションを渡すことができない．そこで，droid-wrapperを用いることによりコンパイラとリンカに適切なオプションを渡すことができ，ビルドが成功する．droid-wrapperを使用するためにはAndroidのソースコードをダウンロードし，開発向けにビルドしたものが必要となる．

3 Androidにおける匿名属性認証システムの実装

3.1 実装方針

Androidにおける匿名属性認証システムの概要を図3に示す．本研究では匿名認証システムとしてAND relationを用いた匿名属性認証システムを利用する．また，匿名属性認証システムの認証アルゴリズムはC言語で実装されているため，JNI(Java Native Interface) を用いて認証機能を実装する．

[2]のPCに対する匿名認証システムではWebサービスとの連携のためにプロキシを用いていたが，Androidではセキュリティ上の問題でユーザ権限ではプロキシを動作させることができない．そこで，実装方針として，[2]におけるクライアント側のプロキシの機能を代替する認証用アプリケーションを作成することとした．認証用アプリケーションは，ブラウザから認証要求を受け取り，認証を行った後，サーバ側からレスポンスを受け取ってブラウザに渡す．プロキシとの違いは，プロキシは常時起動してブラウザからの要求を待ち続けているが，認証用アプリケーションは認証要求が発生してから起動され，認証が終了してレスポンスをブラウザに渡すと認証用アプリケーションを終了させる必要があることである．

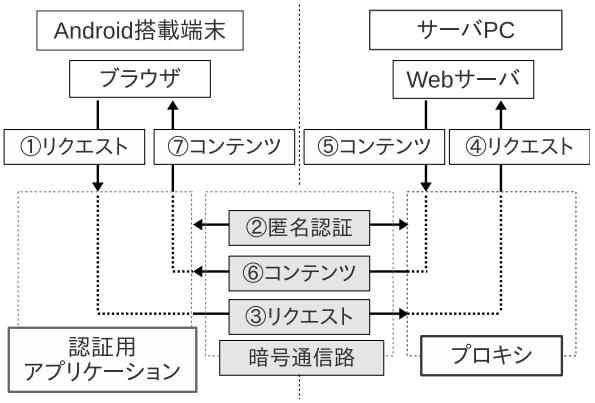

図 3: Android用匿名属性認証システムの概要

3.2 Webブラウザとの連携

Webブラウザと認証用アプリケーションの連携を図4に示す．認証用アプリケーションは，Webブラウザからユーザが選択した属性カテゴリを受け取り，匿名属性認証を行い，Webブラウザに遷移先のURLへアクセスさせる．

3.2.1 Webブラウザからの認証用アプリケーション起動方法

まず，Webブラウザから認証用アプリケーションを起動させる方法を示す．Webブラウザから認証用アプリケーションを起動させ，ユーザが選択した属性カテゴリを受け渡すために，Androidのインテントという仕組みを利用する．

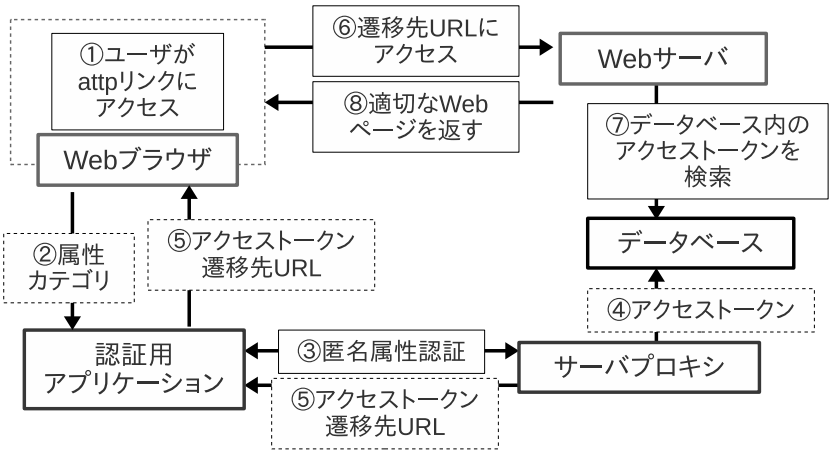

図 4: Web ブラウザと認証用アプリケーションの連携

インテントとは，アクションとパラメータを含んだメッセージの受け渡しを行うことでアプリケーションを連携させる仕組みである．本研究では暗黙的インテントを利用する．暗黙的インテントのメッセージがアプリケーションから発行されると，OS がメッセージ中のアクションとパラメータの組を処理可能なアプリケーションを探し出して起動する．これは，各アプリケーションが処理可能なアクションとパラメータを記述するマニフェストファイルを備えており，それを OS が参照することによって実現している．

実装したシステムでは，Web ブラウザが発行するメッセージの内容が URL の文字列のみであるため，GET パラメータを利用し，URL に属性カテゴリをパラメータとして付与し，認証用アプリケーションに受け渡す．このとき，attp というスキームを新しく定義し，認証用アプリケーションのマニフェストファイルに記述した．このスキームは，http リンクのように Web ページ中に記述することができ，`"attp://"` で始まるリンクに Web ブラウザでアクセスすると，Web ブラウザから，アクションが attp でパラメータに属性カテゴリを含んだ URL を文字列を含んだ暗黙的インテントのメッセージを，認証用アプリケーションに受け渡すことができる．このとき，Web ブラウザから認証用アプリケーションに受け渡す文字列は，`"attp://ホスト名/パス?属性カテゴリ"` の形式で表現する．例えば，文字列として `"attp://example.jp/test.jsp?att1=4&att2=3"` が与えられたとき，GET パラメータ部分の `"att1=4&att2=3"` がユーザが選択した属性カテゴリを示す．

3.2.2 Web ブラウザへの遷移先 URL 受け渡し方法

次に，認証用アプリケーションから Web ブラウザにレスポンスを受け渡す方法を示す．Android では，認証用アプリケーションからブラウザに Cookie やセッションなどの情報を受け渡すことができないため，アクセストークンを用いてアクセス制御を行う．まず，サーバプロキシは匿名属性認証が成功した際に乱数としてアクセストークンを生成し，データベースに保存する．そして，サーバプロキシから認証用アプリケーションに遷移先の URL とアクセストークンを送信する．認証用アプリケーションは受け取った URL にアクセストークンをパラメータとして付与し，暗黙的インテントのメッセージを発行し，Web ブラウザに遷移先 URL にアクセスさせる．Web サーバ側では，アクセスがあった際にパラメータからアクセストークンを取得し，データベース内にアクセストークンが存在している場合，認証が成功した旨を表示させ，データベースからアクセストークンを削除する．これにより，同じトークンを用いた 2 回以上のアクセスや不正なトークン，トークンなしでのアクセスを防ぐことができる．

3.3 C 言語ライブラリおよび匿名属性認証ライブラリのビルド

[2] のシステムは x86 プロセッサを搭載した PC を対象として実装されている．一方，Android 端末の多くは ARM プロセッサを搭載しているため，C 言語で作成されたライブラリに互換性がない．そこで，C 言語を用いて作成されたライブラリは，Android NDK を用いて，多くの Android 端末が搭載している CPU である ARM プロセッサ向けにビルドする必要がある．

匿名属性認証システムは，ライブラリとして C 言語を用いて作成された多倍長演算ライブラリ GMP(The GNU Multiple Precision Arithmetic Library)[6] およびペアリング楕円曲線演算ライブラリ ELiPS(Efficient Library for Pairing - based System) を利用しており，さらに匿名属性認証ライブラリが ELiPS の関数を使用している．ライブラリのビルドの順番は，まず GMP のビルドを行い，次に GMP のビルド成果物を利用して ELiPS をビルド，最後に GMP と ELiPS のビルド成果物を利用して匿名属性認証ライブラリをビルドする．

しかしながら，GMP および ELiPS はビルドを容易にするために libtool のようなビルド支援ツールを用いており，Android NDK のビルドツールはビルド時に libtool へビルドオプションを渡すことができないため，そのままではビルド不可能であるという問題点がある．そこで，libtool にビルドオプションを透過的に渡すためのラッパーである droid-wrapper を用いる．droid-wrapper を用いてライブラリをビルドするためには，まず Android のソースコードを開発向けにビルドする必要がある．そして，ビルドした Android ソースツリーの場所を droid-wrapper に渡した上で，コンパイラを droid-gcc，リンカを droid-ld と指定してライブラリをビルドする．

ビルド成果物である共有ライブラリまたは静的ライブラリは，Android でネイティブコードを扱う際に使用する Makefile である Android.mk に記述することで使用できる．

3.4 通信機能の実装

従来のプロキシを用いたシステム [2] では匿名属性認証に必要なデータをやり取りする際に，SSL のソケットを作成した上で，ObjectInputStream および ObjectOutputStream というクラスを用いてオブジェクト単位で通信を行なっている．オブジェクト単位で通信を行うためには，やり取りするオブジェクトを直列化する必要がある．直列化とは，オブジェクトを 1 バイトずつ読み書きできるバイト配列状のデータ構造であるストリームの状態に変換することを指す．

しかし，PC と Android とでは，Java の実装の都合上，オブジェクトの直列化の仕様に互換性がないため，各々が受け取ったストリームを元のオブジェクトに戻すことができない．したがって，オブジェクト単位での通信は不可能である．そこで，実装システムでは，PC および Android の双方で，ObjectInputStream の親クラスである InputStream クラス，ObjectOutputStream の親クラスである OutputStream を用いて，バイト配列でデータを送受信するような仕様に変更した．

4 実験と評価

4.1 実験の概要

実装したシステムの動作確認及び評価のために，C 言語で実装されている匿名属性認証ライブラリの認証データ生成時間，認証アプリケーションがサーバプロキシと接続を確立してから認証を完了するまでに要した時間，認証用アプリケーションが起動してからブラウザに遷移先ページを表示させるまでの全処理時間を，それぞれ属性数を変化させながら計測し平均をとった．

4.2 実験環境

クライアントの実験環境を表 1，サーバの実験環境を表 2 に示す．実機とサーバ PC はルータを介して同じ LAN 上に存在する．

表 1: クライアントの実験環境

	スマートフォン	タブレット
OS	Android 2.3.3	Android 4.0.3
CPU	Qualcomm MSM8255 Snapdragon 1GHz	NVIDIA Tegra2 1GHz
RAM	768MB	1GB
ネットワーク	IEEE 802.11g	
多倍長演算ライブラリ	GMP 5.0.0	

表 2: サーバの実験環境

OS	Ubuntu 11.04
CPU	Intel Core i5 650@3.2GHz
RAM	3.8GB

4.3 実験結果

実験結果を表 3 に示す.

表 3: 実験結果 (平均時間)

	スマートフォン	タブレット
認証データ生成時間 [s]	0.34	0.287
認証時間 [s]	0.59	0.62
全処理時間 [s]	1.11	0.988

表 3 の実験結果より，ネイティブコードの実行時間である認証データ生成時間は，スマートフォンとタブレット共に 0.3 秒程度である．認証時間は 0.6 秒であり，匿名属性認証システム単体としての動作時間は実用的である．また，認証時間のうち約 50%が認証データ生成時間となった. そして，ブラウザの動作時間も含んだ全処理時間は約 1 秒であり，モバイル環境での Web サービスとしては問題ないと考える．

5 むすび

本研究では, 匿名属性認証システムを Android に実装し，従来のプロキシを用いた匿名認証システムを用いた Web サーバと連携させ，処理時間を測定することにより評価を行った．測定結果より，実機での署名生成時間は 0.3 秒，認証時間は 0.6 秒，ブラウザに遷移先ページが表示されるまでの時間は約 1 秒であった．これらの結果から，実装したシステムは実用的であることを確認できた.

今後の課題として，匿名属性認証以外の認証方式を容易に組み込めるよう API を整備することや，無線 LAN 認証など Web サービス以外での匿名認証の Android 実装が考えられる．

参考文献

[1] A. Sudarsono, T. Nakanishi, and N. Funabiki, "Efficient Proofs of Attributes in Pairing-Based Anonymous Credential System", Proceeding of The 11th Privacy Enhancing Technologies Symposium (PETS2011), pp.246-263, 2011.

[2] 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 (CSS)，pp.163-168，2008-10.

[3] 濱田雄治，中西透，舩曳信生，"Web サービスにおける匿名属性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 (CSS)，pp.60-65，2010-10.

[4] droid-wrapper, `https://github.com/tmurakam/droid-wrapper`.

[5] AndroidNDK, `http://developer.android.com/tools/sdk/ndk/index.html`.

[6] GMP, `http://gmplib.org`.