

Android における NTLM 認証を可能とする Web プロキシアプリの実装

柿 祐輔† 後藤 浩行‡ 村上智祐‡ 齋藤 孝道†
 明治大学† 明治大学大学院‡

1. はじめに

近年、スマートフォンの普及に伴い、スマートフォンを用いて Web サイトを閲覧する機会が増大した。

その際、個人情報を扱う Web サイトなどでは個人を特定するために ID とパスワードを用いて認証をする場合がある。

しかし、スマートフォンの OS の一つである Android に標準でインストールされているブラウザ（以降、Android Browser と呼ぶ）では ID とパスワードを用いた NTLM 認証 [1] を使用することができない。

すなわち、Android が NTLM 認証を行う Web サーバ（以降、NTLM Authentication Server と呼ぶ）、例えば、Exchange サーバや Sharepoint サーバなどと通信できない。

本論文では、Android において NTLM 認証を行うために Android のプロキシアプリケーションを実装し、Android Browser からプロキシアプリケーションにアクセスすることにより、NTLM 認証を行う方法を示す。

2. NTLM 認証

2.1. 概要

NTLM 認証とは Windows 系 OS におけるネットワーク上での認証方式の 1 つであり、現在でも多くの Windows 系の Web サーバで用いられる認証方式である。

NTLM 認証はチャレンジ/レスポンス方式を採用している。チャレンジ/レスポンス方式とは認証を受けるブラウザが認証要求をサーバに送り、サーバがチャレンジというランダムの数値列を返信する。ブラウザはパスワードとチャレンジから指定のアルゴリズムを用いてレスポンスを作成し、レスポンスを送信する。これを用いることで平文のパスワードがネットワーク上に流れなくなる。

「An implementation of Web proxy application enables Android browser to use NTLM authentication」

† Yusuke Kaki

‡ Hiroyuki Goto, Tomosuke Murakami

† Takamichi Saito

Meiji University(†)Graduate School of Meiji University(‡)

以下に NTLM 認証時に使用されるアルゴリズムについて説明する。

ユーザが入力したパスワードを Unicode に変換する。変換したパスワードをハッシュ関数を用いて、ハッシュ化する。そして、ハッシュ値を DES の暗号鍵として利用し、チャレンジを暗号化し、レスポンスを作成する。

2.2. フロー図

ここでは標準的なフローの説明をするために PC のブラウザを用いて行う HTTP 上での NTLM 認証を例として説明する。（図 1 参照）

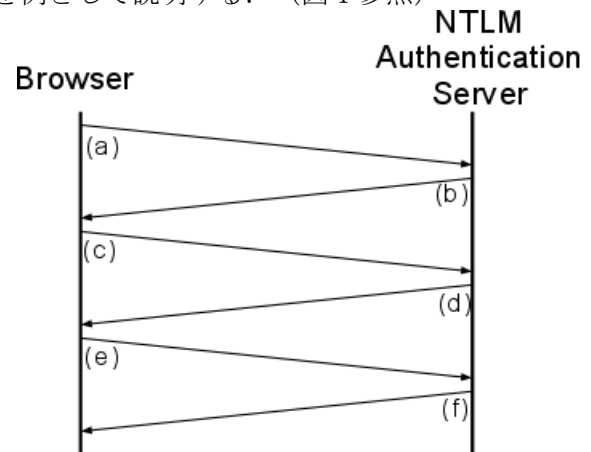


図 1 NTLM 認証フロー

- Browser から NTLM Authentication Server にアクセスする。
- NTLM Authentication Server にアクセスする際に NTLM 認証が必要なので、NTLM 認証が必要であることを Browser に通知する。
- Browser が NTLM Authentication Server に、NTLM 認証が必要であることを了承し、チャレンジコードを要求する。
- NTLM Authentication Server は NTLM 認証を受けると、チャレンジを生成し、それを Browser に送信する。
- Browser 上で入力したパスワードとチャレンジからレスポンスを作成し、NTLM Authentication Server に送信する。
- NTLM Authentication Server は保存してあ

るパスワードとチャレンジから同様にレスポンスを作成し、レスポンスと比較し、認証結果を Browser に通知する。

3. 提案システム

3.1. 概要

図1でPCのブラウザを用いてNTLM認証のフローを説明した。しかし、Androidでは以下の問題点が挙げられる。

Android BrowserではNTLM Authentication Serverとの間で通信をする際に図1の(b)を受信するが、(c)を送信せずに(a)を送信しているため、チャレンジコードを受信できず認証ができない[2]。

本論文では、Androidアプリケーションとして、NTLM認証のプロキシを実装し、Android BrowserからプロキシアプリケーションにアクセスすることでAndroidにおいてNTLM認証を可能にする仕組みを実装した。

3.2. 構成

Android Browser

- Android 2.2
- WebKit 533

Android BrowserはAndroidのバージョンにより、WebKitのバージョンも異なっており、本論文ではAndroid 2.2を用いた。

Android Proxy App

- Android 2.2
- JRE 6
- JCIFS 1.3.15

AndroidアプリはJavaで実装されている。

JCIFS[3]はWindows関連の通信をJavaで利用可能にしたライブラリである。本論文では、JCIFSを用いて、Android Browserの代理にNTLM Authentication Serverとの認証を行い、認証結果をAndroid Browserに通知するWebプロキシアプリケーションを実装した。

NTLM Authentication Server

- Windows Server 2008 R2
- IIS 7.5

Android Browserからアクセスする対象として、NTLM認証を必要とするWebページを、当該サーバ上に用意した。

3.3. フロー図

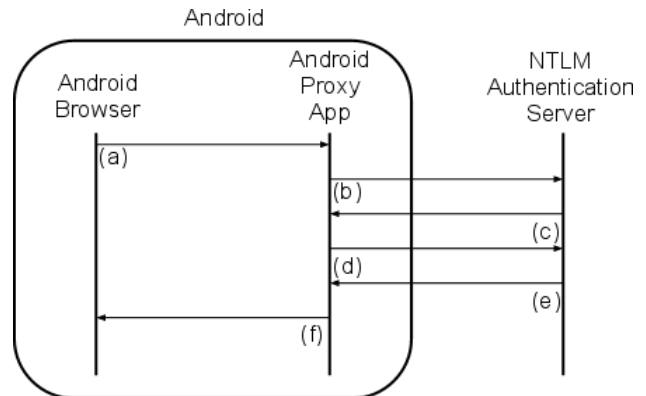


図2 提案システムのフロー

- Android Browserから認証先のURLをパラメータにつけて、Android Proxy Appにアクセスする。
- Android Proxy Appがパラメータで指定されたNTLM Authentication Serverとの通信でNTLM認証が必要なことを了承し、チャレンジコードを要求する。
- NTLM Authentication ServerがAndroid Proxy Appにチャレンジを返す。
- Android Proxy Appがパスワードとチャレンジからレスポンスを作成し、NTLM Authentication Serverにレスポンスを送信する。
- NTLM Authentication ServerがAndroid Proxy Appから送られてきたレスポンスを図1の(f)と同様に比較し、Android Proxy Appに認証結果を通知する。
- Android Proxy AppがAndroid Browserに認証結果を通知する。

4. まとめ

NTLM認証が必要なWebサーバと通信する際にプロキシアプリを利用することでAndroid BrowserからもNTLM認証を可能にする方法を提案し実装した。Sharepointサーバへの接続も確認した。

5. 参考文献

- NTLM認証プロトコルとセキュリティサポートプロバイダ
<http://www.monyo.com/technical/samba/translation/ntlm.html>
- NTLM authentication in Browser
<http://code.google.com/p/android/issues/detail?id=4962>
- JCIFS NTLM HTTP Authentication
<http://jcifs.samba.org/src/docs/ntlmhttpauth.html>