マルチメディア通信と分散処理　106－18
コンピュータセキュリティ　16－18
（２００２．２．１４）

SAS 暗号通信方式を利用した HTTP ベースの安全、簡便な認証方式

真島 大介　羽田 知良　田鍋 潤一郎　清水 明宏
NTT アドバンステクノロジ㈱　高知工科大学

あらまし

インターネット上の各種認証方式には、ネットワーク上のデータが暗号化されない、サーバ上のデータが暗号化されない、クライアント側に認証に必要な情報を記憶しておく必要がある等の問題がある。我々は、次回認証情報予告型のワンタイムパスワード方式 SAS（Simple And Secure）認証方式を HTTP プロトコルに実装することにより、このような課題を解決した。具体的には、Web クライアントと Web サーバの間に HTTP ベースの SAS Proxy および認証サーバを挿入し、そこに SAS の技術を適用することにより、簡便、安全かつ低コストな認証方式を実現した。

評価の結果、本方式は SSL に比較して次のような優位性をもっていることを確認した。

① 通信パケット量は約 70%と少ない。

② 同時アクセスがあるサーバの負荷は約 50%で済む。

HTTP based secure and simple authentication method using SAS

Daisuke Mashima　Tomoyoshi Hada　Junichiro Tanabe　Akihiro Shimizu
NTT Advanced Technology Corporation　Kochi University of Technology

Abstract

Authentication methods on the Internet have some problems. For example, data is not encrypted on the network or server, and private data must be kept in the client.

We solved these problems by applying SAS(Simple And Secure) , a kind of one time password technology, on the HTTP protocol.

Our idea is to insert HTTP based SAS Proxy and authentication server between web client and web server. This idea makes authentication easy secure and low cost.

We implemented this authentication system and appeared our system is more advantaged than SSL because of following result.

1) Total communication packets are 70% of SSL,

2) Load of server which has simultaneous access is 50% of SSL.

1 はじめに

インターネットの Web を利用したエレクトロニックコマース（EC）やイントラネットでの利用が増加し、これに伴いセキュリティ確保が重要な課題になっている。

我々は、TAO（通信・放送機構）から受託した研究開発「学校インターネット I」の一環として、小中学校へのインターネット導入に取り組んで

いる。学校のように、複数の生徒や教師が１台のPCを共有するような環境では、簡便に各自のセキュリティを確保する必要があり、その対策として、SAS（Simple And Secure）認証方式を利用したWeb（HTTP）ベースの認証方式を考案した。

実際にシステムを構築して評価したところ、SSL（Secure Sockets Layer）と比較して、通信パケット量は約70%、同時アクセスがあるサーバの負荷は約50%で済むことがわかった。

本論文では、考案した方式およびそのシステムの評価結果について述べる。

2 認証方式の現状と課題

既存の認証・暗号通信方式には次のような問題がある。

① ワンタイムパスワード： パスワード生成用の専用のハードウェア（カード）を用いて毎回パスワードを生成し、さらにそのパスワードをシステムに入力する方式である。セキュリティは確保できるが、操作が面倒であるという問題がある。

② SSL（Secure Sockets Layer）：電子証明書による公開鍵暗号方式を用いて、認証及び暗号通信を行う方式である。ただし、なりすましを防止するために PKI（Public Key Infrastructure、公開鍵暗号基盤）を導入する必要があり、認証処理に時間がかかる、導入、運用に費用がかかるという問題がある。

③ Basic 認証、Digest 認証：Basic 認証は、HTTP プロトコルヘッダに、認証用パスワードを設定して送る方法である。この方式は、通信のたびに毎回認証処理が行われるという利点があるが、HTTP プロトコルヘッダの内容は暗号化されないため、盗聴されやすいという問題がある。この問題に対処したのがDigest認証であるが、サーバにパスワードは平文のまま保存されるため、サーバアタックに弱いという問題がある。

3 SAS 認証方式

SAS 認証方式[1]は、次回認証情報予告型のワンタイムパスワード方式であり、次の特長をもつ。

① 入力されたパスワードから毎回異なった認証情報を生成して安全に認証可能

② 次回の認証情報を予告することによって、極めて少ない計算量で認証を実現可能

具体的には、次のように利用する。

① ワンタイムの認証情報の作成

・利用者はクライアント端末からユーザ ID、パスワードを入力する。

・クライアント側で、これらと乱数を用いて認証キーの元となる認証ベースおよびワンタイムの認証キーを作成する。この際、今回分と次回分の認証キーを作成する。

② サーバへの認証情報の送信

・今回分の認証情報を作成するための情報と次回分の認証情報を、ワンタイムの暗号鍵で隠蔽し、サーバに送信する。

③ サーバにおける認証

・サーバでは、前回認証を行った際、今回分の認証情報を予約、保存しておく（予約情報）。今回受信した情報と予約情報から、今回の認証情報を計算し、予約情報と照合し一致していれば認証ＯＫとする。その後、次回用の認証情報を計算し、次回認証用の予約情報として保存する。

4 考案した方式

4.1 概要

HTTP 環境に安全な通信環境を提供するにあたっては以下の点が必要となる。

① 安全かつ簡便なユーザ認証の提供

② 通信データの保護

①については**2章**で述べたような課題の解決が必要となる。②については HTTP リクエストおよびレスポンスのメッセージボディ部分の暗号化

だけでは不十分である。なぜなら、URL の一部、あるいはヘッダ部分に重要な情報が記述されて通信されることがあるためである。したがって、HTTP のヘッダ部分まで含めての保護が必要となる。

我々は、これらの課題を解決するため、HTTP の通信環境に、HTTP ベースの SASProxy サーバおよび認証サーバを挿入し、そこに SAS 認証の技術を適用することにより、簡便、安全かつ低コストなセキュリティプラットフォームを考案した（図 1）[2]。

なお、SASProxy サーバは Web クライアントと同じローカルネットワーク上に、そして認証サーバは Web サーバと同じマシン上、または Web サーバと同じローカルネットワーク上に配置され、ローカルネットワーク上でのセキュリティは保証されていると想定している。

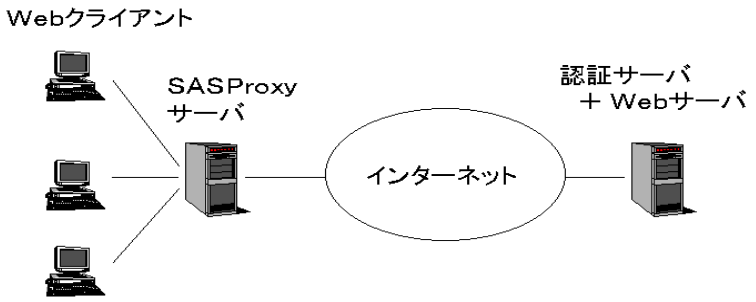

図 1 考案した方式

4.2 処理手順

考案した方式の暗号の処理手順は次の通りである（図 2）。

① Web クライアント（ブラウザ）からのリクエストを SASProxy サーバが受け取る。

②SASProxy サーバは認証サーバと HTTP を用いた通信を行い、ユーザの認証回数、セッション ID などの取得、および、それらの情報とユーザのパスワードなどを用いて暗号化鍵の計算を行う。

③ SASProxy サーバは②で計算した ワンタイムの暗号化鍵を用いて Web クライアントから受け取ったリクエストを暗号化し、認証サーバに送信する。

④ 認証サーバは登録されているユーザ情報か

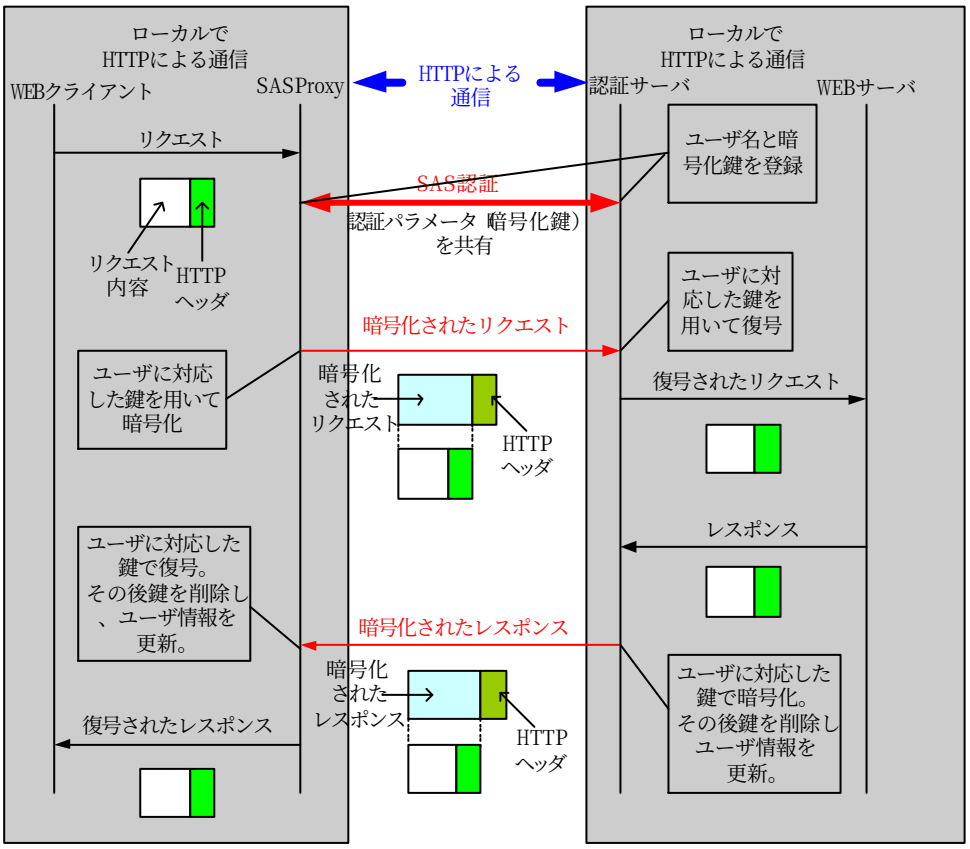

図 2 処理方式

ら復号のための鍵を計算し、受信したリクエストデータを復号する。ここで同時に正しく復号できたかどうかの検証を行う。パスワードが不正である場合は正しく復号できないため、この時点でエラー（認証失敗）となる。

⑤ 正常にリクエストデータを復号できた場合、認証サーバは復号したリクエストデータを Web サーバに中継してそのレスポンスを受け取り、そのレスポンスを復号の際に用いた鍵で暗号化して、レスポンスとして SASProxy サーバに送信する。

⑥ SASProxy サーバは、受け取ったレスポンスデータを②で計算した鍵を用いて復号する。この際、レスポンスデータが正しく復号されたかどうかの検証を行う。正しく復号されなかった場合は、ネットワーク上でデータの改ざんが行われた可能性が考えられるため、処理をやり直す。

⑦ 正常にレスポンスデータを復号できた場合、SASProxy サーバは、復号されたレスポンスデータを Web クライアントに対して送信する。

⑧ SASProxy サーバ、認証サーバとも、処理終了時には、鍵およびユーザ情報を削除し、新しい情報に更新する。

5 評価

考案した方式を実現し、SSL と比較しながら性能評価を行い、本方式の有効性を確認した。ここでは、その結果と考察を述べる。

5.1 評価環境

本システムを以下のような LAN 環境に構築した（図3）。

・ネットワーク：10BASE-T

・認証サーバ：Sun Enterprise 450、メモリ 512MB

・SASProxy サーバ：Sun Ultra10、メモリ 512MB

・Web クライアント：Pentium266MHz、メモリ 128MB

5.2 評価方法

以下のデータを本方式と SSL とを比較しながら測定した。

・上り電文および下り電文のパケット数

・上り電文および下り電文のデータ量

・サーバの CPU 使用率

測定にあたっては、送信するファイルサイズを 10kB、100kB、500kB、1MB と変化させ、それぞれ10回測定し、結果の平均値を求めた。

あわせて、被験者（11名）の実際の体感速度を5段階のアンケート形式で収集した。

5.3 測定結果および考察

①伝送量の評価

図4および図5に伝送パケット量の測定結果を、図6に通信データ量の測定結果を示す。

上りおよび下り電文の総パケット数の総和を比較すると、SAS 方式は SSL 方式の 67%のパケット量で通信が完了している。図4および図5か

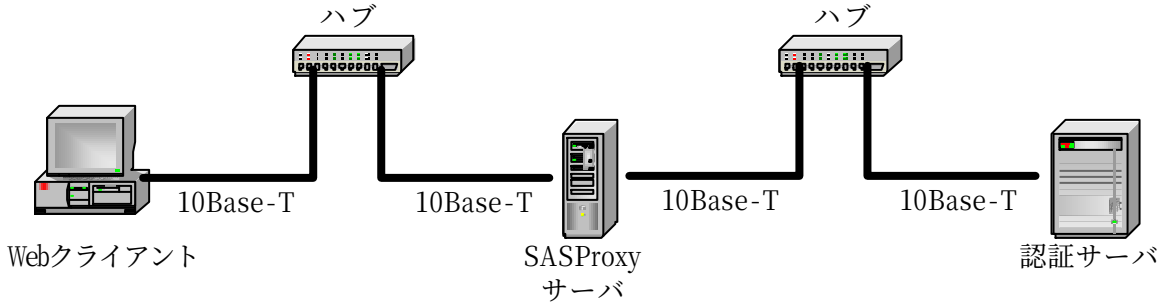

図3 評価環境

ら、ファイルサイズが大きくなれば、さらにこの差は大きくなることが予測される。

一方通信データ量は、SAS 方式は SSL 方式の 96%となっている。図6から、通信データ量についても、ファイルサイズが大きくなるほど、SAS 方式の方が有利になっていることがわかる。パケット数に比較してデータ量の差が小さいのは、SSL 方式の方が、データを含まないパケットが多いことが原因であると考えられる。

(2) サーバの負荷

図７に、単一アクセス時のサーバの CPU 使用率を示す。これから、SAS 方式の方が、サーバの CPU 使用率は高く、ファイルサイズが大きくなるにつれてこの差は大きくなっている（Java で記述したプログラムコードを改良すれば、この差は縮まると思われる）。

しかし、サーバは多くのクライアントから同時にアクセスされるのが一般的である。そこで、通常やり取りすることが多いと予測される 10kB のファイルを送信する場合の複数アクセス時の CPU 使用率も測定した結果が、図8である。これから、SAS 方式の方が、同時アクセス時は SSL 方式よりも優れていることがわかる。この原因は、SSL 方式は負荷の大きい公開鍵暗号方式を使用しているためであると考えられる。

図８の場合は、SAS方式が SSL 方式の 52%の CPU 使用率となっているが、図7の結果を考慮すると、ファイルサイズが大きくなればこの差は縮まると思われる。

図 7および図８から、サーバの負荷については、ファイルサイズおよび同時アクセス数を考慮して、SAS 方式とSSL方式の優劣を判断すればよいことになる。ただし、商用のシステムでは、同時アクセス数がはるかに

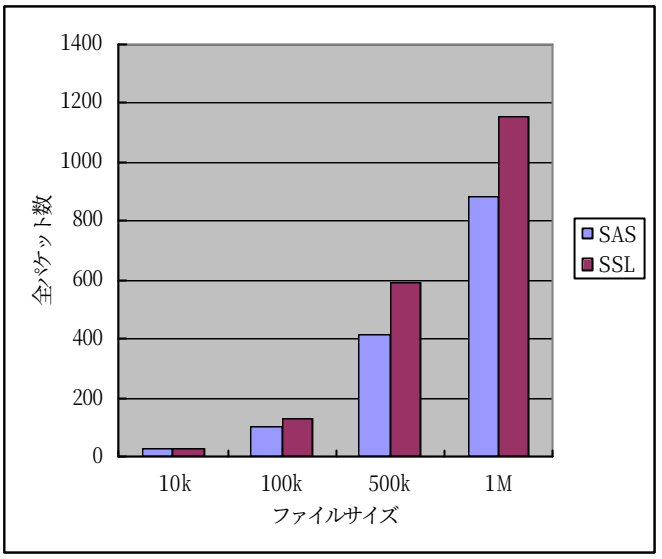

図４ 上り電文のパケット数

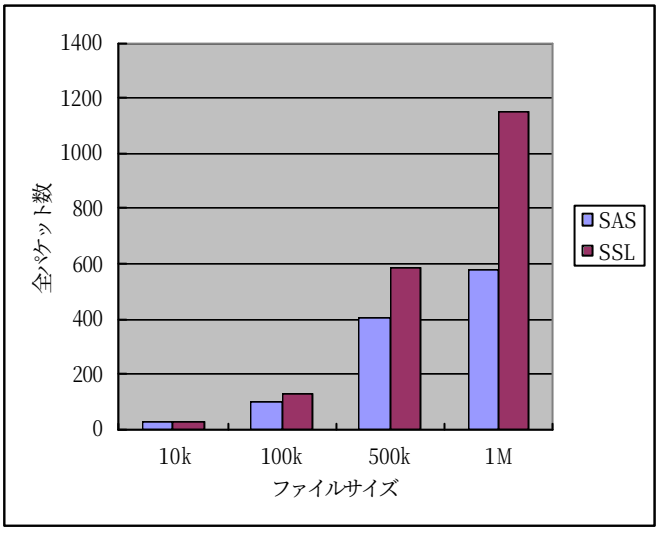

図５ 下り電文のパケット数

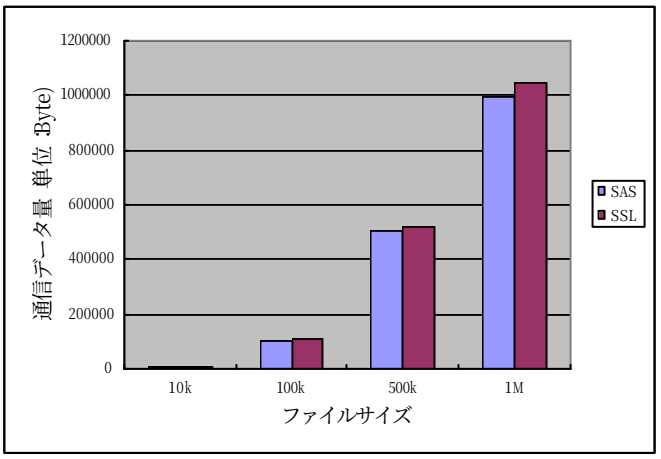

図６ 通信データ量

多くなるため、SAS 方式の方が有利になるであろう。

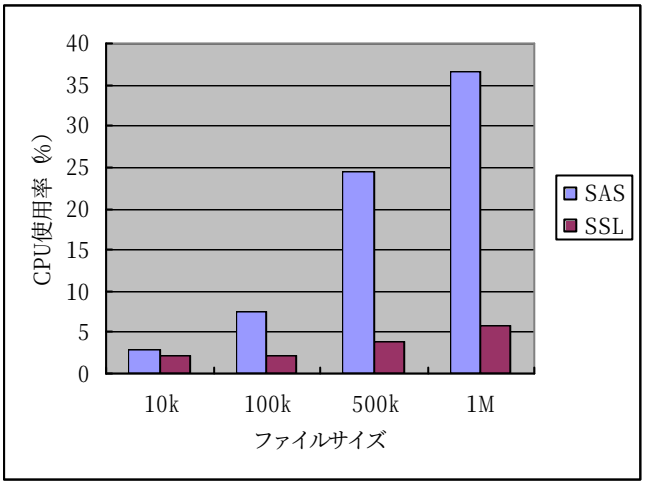

図７ 単一アクセス時のサーバのＣＰＵ使用率

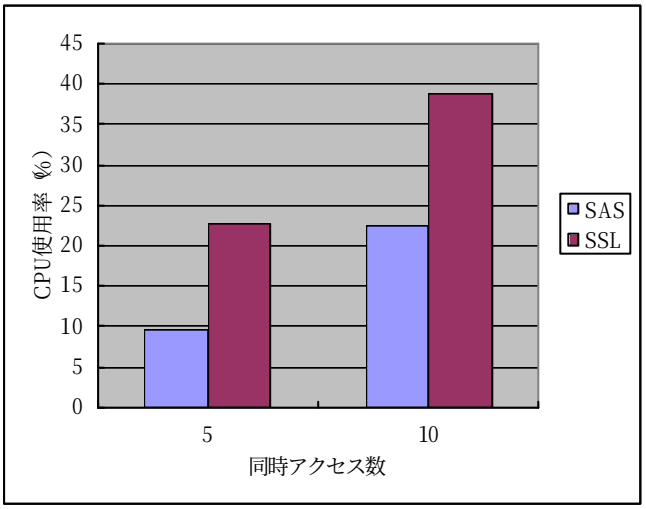

図８ 複数アクセス時のサーバのＣＰＵ使用率

⑶体感速度

被験者 11名の本方式の SSL に対する体感速度の評価の平均値は 3.6 となった。5段階評価なので、2.5 ならば同等、2.5 以上ならば本システムの方が優れていることになる。3.6 という値は、体感速度すなわち応答待ち時間においてもこれまでに述べた性能評価の結果同様本方式が有利であることを示している。

6 おわりに

SAS 方式を利用した HTTP ベースの認証方式を考案し、システムを実現した。SSL と比較して評価を行った結果、本方式は、安全、簡便かつ高性能なシステムであることを確認できた。

今回実装した方式は、ローカルネットワーク上に SASProxy サーバを配置し、ローカルネットワーク上に接続されたクライアントのリクエストを集中的に処理する方式であったが、この SASProxy の機能を各 Web クライアントマシン上に実装することで、Web クライアントWeb サーバ間の通信を End-to-End で保護することも可能となる。また、ローカルプロキシ化することによって、SASProxy モジュールの構造がシンプルになるだけでなく、負荷も小さくなることが予想されるため、安全性、体感（処理）速度の点においての性能向上が期待される。

今回考案した方式は、管理（入力）すべきパスワードが１個（回）であるため、IC カードあるいは小中学校であればフロッピーディスクに保存して各自に持たせることで、非常に安価にパスワードレスシステムを実現できる。今後は、本システムを学校だけでなく企業システムにも適用していく予定である。

［参考文献 ］

⑴Mandula SANDIRIGAMA、Akihiro SHIMIZU、”Simple and Secure Password Authentication Protpcol (SAS)”、pp.1363-1365、IEICE TRANS.COMMUN.、VOL E83-B NO6. June 2000。

⑵真島、羽田、田鍋、清水、”SAS 暗号通信方式を利用した HTTP ベースの安全、簡便な認証方式 ”、情報処理学会第 63 回全国大会、2001 年 10 月。