様式 C-19

科学研究費補助金研究成果報告書

平成２１年５月２２日現在

研究種目：基盤研究（C）
研究期間：2007～2008
課題番号：19500051
研究課題名（和文）データ圧縮を用いる場合における TLS の安全性向上のための研究

研究課題名（英文）Development of Secure Improvement Method for TLS Using Data Compression

研究代表者
木村　成伴（KIMURA SHIGETOMO）
筑波大学・大学院システム情報工学研究科・准教授
研究者番号：20272180

研究成果の概要：
暗号化通信プロトコル TLS (Transport Layer Security)は，データ圧縮を併用してデータを伝送すると，暗号化したデータの種別が推測しやすくなるという問題が知られている．この問題を解決するため，本研究では，TLS のデータ圧縮方式を改善した．そして，提案方式のプロトタイプシステムを用いて，3 種類の電子メールの伝送実験を行うことで，提案方式は処理負荷をほとんど増加させずに，安全性が向上することを示した．

交付額

（金額単位：円）

	直接経費	間接経費	合　計
２００７年度	1,000,000	300,000	1,300,000
２００８年度	1,300,000	390,000	1,690,000
年度			
年度			
年度			
総　計	2,300,000	690,000	2,990,000

研究分野：情報通信工学
科研費の分科・細目：情報学・計算機システム・ネットワーク
キーワード：データ圧縮，ＴＬＳ，ＳＳＬ，暗号化通信プロトコル，電子メール伝送プロトコル，セキュア・ネットワーク，ネットワーク

１．研究開始当初の背景

暗号化通信プロトコルTLS (Transport Layer Security)はTCPとアプリケーション層の間で動作し，通信内容を第三者から秘匿する機能，そして，通信相手を認証し，送受信されたデータが通信経路上で改竄されていないことを保証する機能をアプリケーション層に提供する．このため，同プロトコルは古くからHTTP (Hypertext Transfer Protocol)で広く利用されてきたが，その利用はSMTP (Simple Mail Transfer Protocol)やPOP (Post Office Protocol), IMAP (Internet Message Access Protocol)などの電子メール送受信用のプロトコルなどにも拡がっている．このTLSでは，図 1に示すように，アプリケーション層から送られたデータを最大16KBまでに細分化し，次にこれらの各々を圧縮する．そして，最後に圧縮済みデータを暗号化するという処理を行っている．この圧縮機能を用いることで，送信データがより小さくなることから，データの暗号化処理にか

かる時間が短縮される他，低速回線を利用しているユーザにとっては，そのデータ伝送時間を大幅に短縮することが可能となる．TLSの仕様書RFC2246 には，利用可能な圧縮アルゴリズムとして「無圧縮」しか規定されていなかったため，従来はこのデータ圧縮機能を用いることは困難であった．しかし，2004年にTLS用の圧縮アルゴリズムとしてDEFLATEを追加することがRFC3749 で規定されたことにより，今後，この圧縮アルゴリズムを実装したシステムが提供されることが予想される．

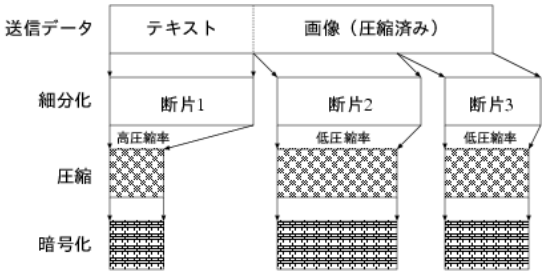

図 1 TLS によるデータ処理例

一方，このRFC3749 でも指摘されているように，データの圧縮を行うと圧縮しなかったときには分からなかった情報が漏洩する可能性があるという問題が知られている．例えば，図 1のように，送信データを 3 つに細分化し，これらの各々を圧縮すると断片 1 の圧縮率は高く，断片 2 と断片 3 の圧縮率は低くなったとする．また，その後の暗号化ではデータサイズが殆ど変わらないものとすると，この通信を傍受した第三者は，断片 1 が圧縮しやすいテキストデータなどであり，断片 2 と断片 3 が圧縮しづらい画像などの圧縮済みデータであることを容易に推測することが出来る．更に，暗号化したデータを伝送する TCP のヘッダ部分は暗号化の対象となっていないため，ここに含まれるポート番号から使用されているアプリケーション層を容易に判別することが可能である．例えば，宛先ポート番号がSMTP over TLSで利用される 25 番であったとすると，図 1の伝送は画像などのファイルを 1 つ添付した電子メールを送信していることが推測される．暗号化によりデータの内容は依然として秘匿されているが，第三者に送信データに関する手掛かりを僅かでも与えることは問題がある．

上述の問題は TLS 特有の問題ではなく，パケット単位でのデータ圧縮を併用した暗号化通信プロトコル全般に該当するが，TLSの場合も含め，これらの問題に対する研究は見られなかった．なお，TLS のように細分化した後に圧縮するのではなく，アプリケーション層で送信データ全体を圧縮し，これを TLS などを用いて(圧縮せずに)暗号化すれば，問題は解決するが，これを実現する為には個々のアプリケーションプロトコルがデータ圧縮機能を持つ必要がある．HTTP では既にデータ圧縮機能が利用可能であったが，他の全てのアプリケーションプロトコルがこの機能を有しているとは限らない．例えば，IMAP にデータ圧縮機能を追加する拡張が提案されているが，IMAP over TLS は広く利用可能なのに対して，提案段階の同拡張機能が普及するにはかなりの時間を要し，更に，この機能に対応する為にはアプリケーションプログラムの変更が求められる．本研究による TLS の拡張は，単にライブラリを変更することで対応可能であり，アプリケーションプログラムの変更を必要としない利点がある．

２．研究の目的

TLS はアプリケーションプロトコルに依存しないという汎用性に着目し，研究代表者は 2001 年度から 2005 年度にかけて,科学研究費の支援の元で，仕組みとしては用意されていたもののこれまで利用されてこなかった TLS のデータ圧縮機能を対象とした研究を行ってきた．特に，電子メールなどのアプリケーションプログラムに特化した圧縮アルゴリズムの提案しており，その成果は公表されている．しかし，前章で述べたデータ内容が推測される問題は，TLS でデータ圧縮を利用するために解決すべき残された課題となっていた．

そこで，本研究では TLS のデータ伝送方法に更なる改良を加え，データ圧縮を併用した場合でも送信データの内容に関する情報これらの情報を第三者が推測することが困難とする手法を開発することを目的とした．

３．研究の方法

情報漏洩防止のための基本的なアイデアは，ネットワーク上に送信するデータサイズを揃えることである．このサイズ調整の処理を，図 1の処理に組み込むことが必要である．しかし，暗号化後のデータを寄せ集めてサイズを規定の長さに調整すると，これを復元する為の境界情報を平文で付加して送る必要があり，これでは情報漏洩を防止したことにはならない．

そこで提案方式では，図 2に示すように，圧縮後のデータにそのバイト数の情報などを含むヘッダを付与し，その後，これらを寄せ集めてサイズを規定の長さに調整してから暗号化することとした．これにより暗号化後のサイズは一様になるため，情報が漏洩することはない．また，暗号化されたデータにはTLSのヘッダを付けて送付されるが，図 1では圧縮によってサイズが小さくなったデータ各々に対してヘッダが付くのに対して，図 2では暗号化の段階で許容されている最

大サイズのデータ毎にヘッダを付けることになる．このため，図 2の方がTLSヘッダを送付する回数が少なくて済む可能性があるという通信効率面での効果ももたらす．

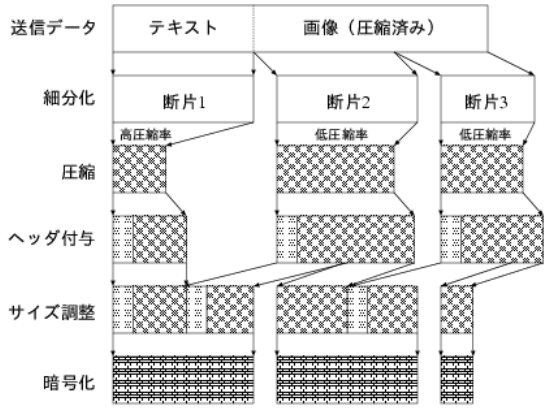

図 2 提案方式のデータ処理

4．研究成果

図 2の処理は図 1のそれよりも段階が多いため，その処理遅延も多くなる．更に，「サイズ調整」段階では固定サイズに達するだけの圧縮済みデータが来るまで送信を遅延させる必要があり，これに伴う待ち時間も生じることになる．そこで，本方式のプロトタイプシステムを実装するとともに，その性能評価実験を行った．

図 3に実験で用いた測定環境を示す．図の環境において，従来のTLSと提案方式を用いた場合について，性能測定を行い，その時のデータの圧縮率と平均データ伝送時間を測定した．ここで，データ回線速度はアナログ電話回線を想定した 28.8kbps，ISDNを想定した 64kbps，ADLSを想定した 3Mbps，LANを想定した 1Gbpsをサーバのファイアウォールにある帯域圧縮機能を用いて用意した．また，転送に用いたアプリケーションプロトコルは，3 種類の電子メール転送プロトコル(SMTP, POP, IMAP)であり，表 1に示す 3 種類のファイルを添付した電子メール(サイズは 40KBと 400KB. テキストファイルは添付のための符号化方式が 2 種類)をクライアント・サーバ間で伝送させた．また，TLSで用いるデータ圧縮方式として，無圧縮，汎用圧縮アルゴリズムDEFLATE，研究代表者らが開発した電子メールの転送に特化した圧縮方式(MAILComp)の 3 つを用いた．

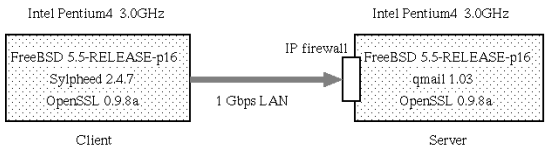

図 3 実験環境

表 1 伝送した電子メールの種別

Type of data	Text encoding	Attached file
Text data	None	Text file
Text data	Base64	Text file
Non-compressed binary data	Base64	MS-Word file
Compressed binary data	Base64	JPEG file

電子メールのサイズが 40KBと 400KBのときの，無圧縮(nc)のデータサイズを 1.00 としたときのデータ圧縮率を表 2と表 3にそれぞれ示す．これらの表から分かる通り，汎用圧縮アルゴリズムDEFLATE (DE)を用いた場合よりも，電子メールに特化したMAILComp (MC)を用いた方が，特に圧縮していないテキストデータや無圧縮バイナリデータ(Wordファイル)の場合に，圧縮率が 1.6 倍程度良くなることが分かった．

表 2 データ圧縮率の比較（40KB）

Kinds of data	nc	DE	MC
Text data without Base64	1.00	0.300	0.299
Text data with Base64	1.00	0.353	0.227
Non-compressed binary data	1.00	0.175	0.114
Compressed binary data	1.00	0.749	0.724

表 3 データ圧縮率の比較（400KB）

Kinds of data	nc	DE	MC
Text data without Base64	1.00	0.239	0.238
Text data with Base64	1.00	0.289	0.180
Non-compressed binary data	1.00	0.200	0.122
Compressed binary data	1.00	0.720	0.691

次に，SMTPを用いて 400KBの電子メールを 10 回転送した時の平均伝送時間と 95%の信頼レベルを表 4～表 7に示す．ここで測定した伝送時間は，TLSのセッションが開始されてから終了するまでの時間を指す．また，TLSの電子署名アルゴリズムはRSAを，暗号化アルゴリズムはAESを，ハッシュアルゴリズムはSHA-1 を使っている．なお，SMTP以外の結果は紙面の都合から省略する．実験で得られたすべての結果は，第 5 章の〔雑誌論文〕に示す文献[1]に掲載している．

表 4 平均伝送時間(秒)

(SMTP, 400KB, 28.8Kbps)

Kinds of data	nc	DE	MC
Text data without Base64	128.5 ±0.01	32.05 ±0.02	31.49 ±0.04
Text data with Base64	169.2 ±0.01	50.20 ±0.02	31.55 ±0.01
Non-compressed binary data	169.2 ±0.01	35.41 ±0.01	21.97 ±0.01
Compressed binary data	168.8 ±0.01	122.1 ±0.02	116.3 ±0.01

表 5 平均伝送時間(秒)

(SMTP, 400KB, 64Kbps)

Kinds of data	nc	DE	MC
Text data without Base64	58.22 ±0.01	14.78 ±0.00	14.71 ±0.01
Text data with Base64	76.58 ±0.01	23.05 ±0.01	14.78 ±0.01
Non-compressed binary data	76.61 ±0.02	16.45 ±0.00	10.43 ±0.01
Compressed binary data	76.44 ±0.01	55.42 ±0.01	53.11 ±0.02

表 6 平均伝送時間 (秒)

(SMTP, 400KB, 3Mbps)

Kinds of data	nc	DE	MC
Text data without Base64	4.48 ±0.01	1.71 ±0.02	1.75 ±0.02
Text data with Base64	5.88 ±0.02	2.47 ±0.02	1.84 ±0.02
Non-compressed binary data	5.86 ±0.03	1.94 ±0.01	1.52 ±0.01
Compressed binary data	5.78 ±0.01	4.44 ±0.02	4.26 ±0.01

表 7 平均伝送時間 (秒)

(SMTP, 400KB, 1Gbps)

Kinds of data	nc	DE	MC
Text data without Base64	0.66 ±0.01	0.68 ±0.01	0.69 ±0.01
Text data with Base64	0.76 ±0.02	0.84 ±0.02	0.84 ±0.02
Non-compressed binary data	0.84 ±0.02	0.83 ±0.02	0.85 ±0.02
Compressed binary data	0.84 ±0.01	0.89 ±0.03	0.86 ±0.01

これらの表より，データ回線速度が遅いほどデータ圧縮の効果は高く，例えば，28.8Kbps の場合，MAILComp は DEFLATE を用いた場合よりも，Base64 で符号化したテキストデータの場合は約 19 秒，無圧縮バイナリデータは約 13 秒も平均伝送時間が短くなっている．しかし，データ回線速度が光速になるにつれて，この差は小さくなり，特に，1Gbps の場合はいずれの圧縮方式も平均伝送時間はほとんど変わらなかった．しかし，この伝送時間には提案方式で追加した処理時間も加わっていることから，これらの処理による遅延はほとんどなく，安全性の問題が解決していることが分かった．

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計　1　件）

[1] Shigetomo Kimura, Yusuke Kumagai, Daigo Manabe, Yoshihiko Ebihara, “MAILComp ― New Compression Method Designed for Electric Mail Transfer Protocols over TLS,” Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems (CISIS2009), pp. 263－271, 2009, 査読有.

〔学会発表〕（計　1　件）

[1] 熊谷祐輔, 木村成伴, 海老原義彦, “TLS における電子メールに特化した圧縮方式の性能評価,” 情報処理学会第 70 回全国大会, 2008 年 3 月 14 日, 筑波大学.

６．研究組織

(1)研究代表者
木村　成伴（KIMURA SHIGETOMO）
筑波大学・大学院システム情報工学研究科・
准教授
研究者番号：20272180

(2)研究分担者
なし

(3)連携研究者
なし