セキュアファイル転送システムの開発

中川和美[†] 岡本隆司[†] 櫻井幸一[††]

インターネットで標準的なファイル転送コマンドであるftpに，慣用暗号によるデータの暗号化技術を適用し，安全なファイル転送システムを開発した．今回，データの暗号化／復号に必要な鍵を，二重暗号化方式により自動配布することで，鍵管理を不要とした．さらに，ファイル転送前に行われるユーザ認証部分には，零知識対話証明技術を適用し，ネットワーク上およびリモートマシン上でのパスワードの盗聴を防いでいる．

Implementation of Secure File Transfer System

KAZUMI NAKAGAWA,[†] TAKASHI OKAMOTO[†] and KOUICHI SAKURAI[††]

This paper reports on an implementation and evaluation of a secure file transfer system over Internet. The proposed system uses a secret-key transferring protocol based on a public-key cryptosystem and zero-knowledge identification scheme. Comparisons of our system to Kerberos is discussed.

1. はじめに

インターネットで標準的なファイル転送システムでは，従来，送受信間で転送されるデータはすべて生のままであり，ネットワーク上の盗聴者が簡単にデータを盗聴することが可能であった．また，ユーザ認証においても，受信側にユーザのパスワードがそのまま転送されるため，盗聴者がネットワーク上で不当にユーザのパスワードを入手したり，何らかの手段で受信側から不当にユーザのパスワードを入手したりして，ユーザに成り済ますことが可能であった．

これらの問題を解決するための一方法として，Kerberos[6]が提案されており，徐々に利用され始めている．しかし，文献1)にも述べられているように，Kerberosを利用するためには安全な認証サーバやチケット交付サーバの設置等の前提条件だけでなく，システムすべてのサービスを一括してKerberos対応化しなければならないという問題点があり，既存の環境にうまく溶け込めなかった．

そこで，今回，二重暗号化方式による鍵配布，零知識対話証明技術に基づくユーザ認証，および，慣用鍵暗号を組み合わせ，

- 転送データの機密性を保つ
- 不正な成り済ましを防止する

と共に，

- 既存システムとの互換性を保つ

ことが可能なセキュアファイル転送システムを開発した．本システムのデータ転送プロトコルは，ファイル転送プロトコルの標準規格であるRFC 959: FILE TRANSFER PROTOCOL[5]に，セキュリティ機能を拡張したInternet Draft: FTP Security Extensions[4]に準拠している．これにより，従来のファイル転送システムとの互換性を保ち，オープンなシステムに対応することが可能である．また，従来のファイル転送システム(7K step)に3K stepの追加プログラムを加えるだけで容易に実現できる．

2. 背景

2.1 従来システムの問題点

従来のファイル転送システムの構成を以下に示す．

従来のファイル転送システムでは，システムを起動すると，ユーザ認証を行い(**図1**の[a])，認証に成功するとファイルが転送される(図1の[b])．

1. ユーザ認証
   送信側が，ユーザ名とユーザを証明するパスワードを受信側に転送する(図1の[c])．

† 三菱電機株式会社情報システム研究所
Computer & Information Systems Laboratory, Mitsubishi Electric Corporation
†† 九州大学工学部情報工学科
Department of Computer Science and Communication Engineering, Faculty of Engineering, Kyushu University

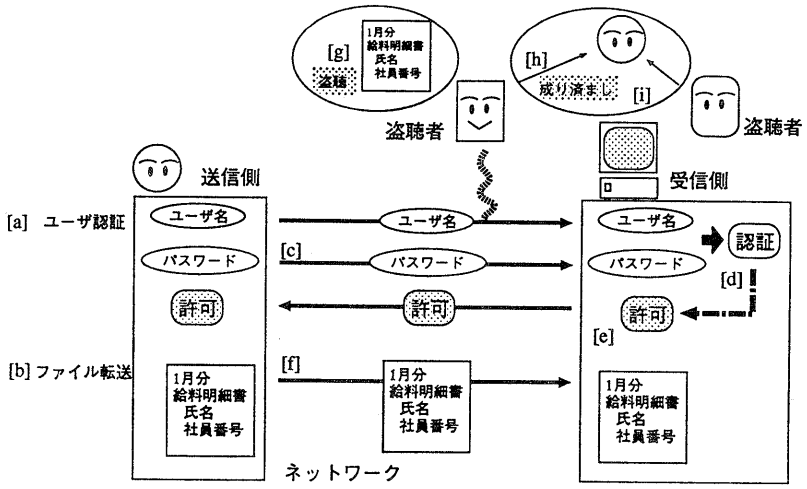


図 1 従来のファイル転送システム
Fig. 1 Usual file transfer system.

受信側は，転送されたユーザ名とパスワードによりユーザが本物であるかどうかを確認する（図1の［d］）.

ユーザ認証が失敗すると，ユーザは受信側にファイルを転送することはできない.

ユーザ認証が成功すると，ユーザにファイルの転送が許可される（図1の［e］）.

2. ファイル転送

ユーザ認証が成功し，ファイル転送の許可が得られたら，ユーザは受信側にファイルを転送する（図1の［f］）.

なお，送受信間のデータはすべて，何の処理もされずにネットワーク上を転送される.

このことから，従来のファイル転送システムには，以下の問題点があげられる.

- 盗聴者がネットワーク上から転送したファイルを盗聴する（図1の［g］）.
- 盗聴者がネットワーク上から転送中のパスワードを不正に入手し，ユーザに成り済ます（図1の［h］）.
- 盗聴者が受信側のシステムを変更することにより，転送されたパスワードを不正入手して，ユーザに成り済ます（図1の［i］）.

2.2 解決へのアプローチ

上記の問題点を解決するための一方法として，Kerberos が提案されている.

Kerberos を利用したファイル転送システムの例をあげると以下のようになる．システムの構成は，サーバとしてクライアントの身元を証明する「チケット」の発行を行う Kerberos サーバと呼ばれる認証サーバ（**図2**の［a］）およびチケット交付サーバ（TGS: Ticket-granting Server（図2の［b］）），クライアントとして送信側（図2の［d］）および受信側（図2の［e］）となる．以下の図を用いて，Kerberos を利用したファイル転送システムの処理の概要を述べる.

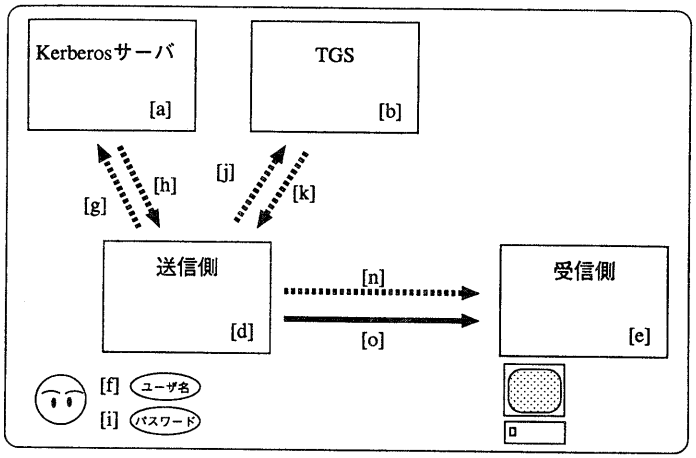


図 2 Kerberos を利用したファイル転送システム その1
Fig. 2 Kerberos system 1.

送信側と受信側が同一のサーバにより管理されている場合，

1. 送信側は，ユーザ名を格納する（図2の［f］）．
2. 送信側は，Kerberos サーバにユーザ名と TGS への身元を保証するためのチケットを要求する（図2の［g］）．
3. Kerberos サーバは，送信側にチケットを返す（図2の［h］）．
4. 送信側はパスワードを格納する（図2の［i］）．
5. 送信側は，TGS に受信側への身元を保証するためのチケットを要求する（図2の［j］）．
6. TGS は，送信側にチケットと送受信間で使用する共有鍵を返す（図2の［k］）．
7. 送信側は，受信側にチケットと共有鍵を転送する（図2の［n］）．
8. 送信側は，受信側に共有鍵を使って暗号化ファイルを転送する（図2の［o］）．
9. 受信側は，共有鍵を使ってファイルを復号する．

また，送信側と受信側が別々のサーバにより管理されている場合は，更に受信側を管理する TGS（**図3**の［c］）へのチケット要求（図3の［l］，［m］）も必要となる．

以上から，Kerberos を導入し，安全なファイル転送システムを構築するには，

- 鍵情報などの盗聴を防ぐことのできる安全な Kerberos サーバおよび TGS の設定
- システム中で認証機能を用いているサービスすべての Kerberos 対応化

が必要である．

このように，Kerberos を導入するには，該当するサービスすべてを同時に Kerberos 対応にする必要があり，既存環境と共存しながら徐々に Kerberos 対応サービスをインストールしていくという手法は取れない．これが既存環境への適合を妨げている要因の一つであろう．

2.3 本システムの目的

今回開発した，セキュアファイル転送システムの目的は以下のとおりである．

- 一般環境での混在使用

ファイル転送システムのみの設定で簡単にシステムを構築でき，大規模なシステム構築を必要としない．また，その他の従来サービスには全く影響を及ぼさない．

- オープン環境に対応

標準規格のプロトコルを採用し，従来のファイル転送システムとの互換性を保つ．

- アルゴリズムの実証

零知識対話証明を用いたユーザ認証と，二重暗号化方式を用いた暗号化ファイル転送を組み合わせた方式を実装することによって，その有効性を示す．

3. セキュアファイル転送システム

3.1 特　　徴

本システムの特徴は以下のとおりである．

- パスワード転送なしのユーザ認証

零知識対話証明[3]を用いることにより，パスワードを転送することなく，ユーザ認証を行う．これにより，盗聴者および第三者が，ユーザのパスワードをネットワークや受信側から不正に入手し，ユーザに成り済ますことを防止する．

- 転送データの暗号化／復号

送信側と受信側間で転送するデータを，慣用鍵暗号を用いてすべて暗号化した．これにより，ネットワーク上の盗聴を防止する．

- データの暗号化／復号のための鍵管理不要

送受信間でのデータを暗号化／復号するための鍵を共有鍵とし，二重暗号化方式[7]を用いて暗号化し，送信側から受信側に配布する．共有鍵は毎回ランダ

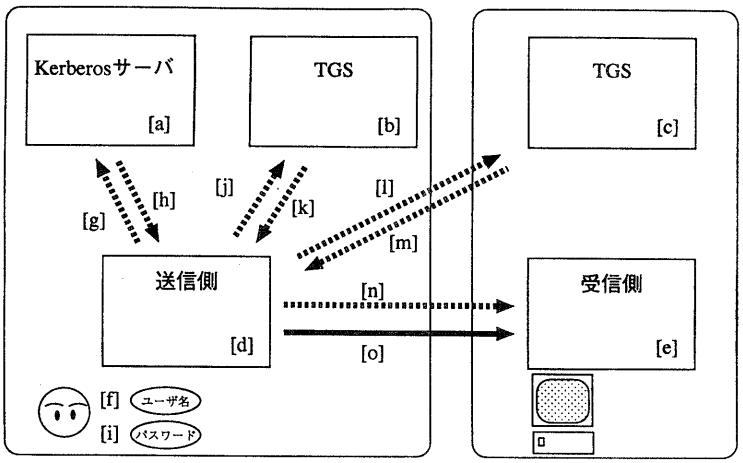


図 3 Kerberos を利用したファイル転送システム　その2
Fig. 3 Kerberos system 2.

ムに作成する．これにより，鍵管理を不要とする．

- Internet Draft に準拠した暗号化データ転送プロトコル

暗号化データ転送プロトコルは，ファイル転送プロトコルの標準規格である RFC 959 : FILE TRANSFER PROTOCOL[5] に，セキュリティ機能を拡張した Internet Draft : FTP Security Extensions[4] に準拠している．これにより，従来のファイル転送システムとの互換性を保ち，オープンなシステムに対応する．すなわち送信側あるいは受信側が従来のファイル転送システムを用いている場合，鍵配布や転送データの暗号化／復号を行わず，また，ユーザ認証には従来の方式を用いてファイル転送を行う．これにより，既存システムとの互換性を保つことができる．

3.2 システムの構成

以下の図を用いて，本システムの構成を述べる．

本システムを起動すると，まずデータの暗号化／復号のための共有鍵の作成および配布が行われ（**図 4** の［a］），次にユーザ認証が行われる（図 4 の［b］）．ユーザ認証に成功すると，ファイルの転送が行われる（図 4 の［c］）．ユーザ認証後に転送されるデータは，すべて共有鍵を用いて暗号化される．

1. 共有鍵配布

   まず送信側がデータの暗号化のための共有鍵をランダムに作成する．

   送信側は，二重暗号化方式により共有鍵を暗号化して受信側に配布する（図 4 の［d］）．

2. ユーザ認証

   送信側が，ユーザ名を暗号化して受信側に転送すると，受信側は，これを復号して受けとる（図 4 の［e］）．その後，送受信間で相互に零知識対話証明によるデータを暗号化して転送する（図 4 の［f］）．

   受信側は，零知識対話証明によるデータを復号し，ユーザ名とデータによりユーザが本物かどうか確認する（図 4 の［g］）．

   ユーザ認証が失敗すると，ユーザは受信側にファイルを転送することはできない．ユーザ認証が成功すると，ユーザにファイルの転送が許可される（図 4 の［h］）．

3. ファイル転送

   送信側は，ファイルを共有鍵で暗号化して受信側に転送する．

   受信側は，転送された暗号化ファイルを共有鍵を用いて復号する（図 4 の［i］）．

本システムは，従来のファイル転送システムに，共有鍵配布処理，零知識対話証明によるユーザ認証処理，ファイルの暗号化／復号処理を追加したものである．処理構造の図を**図 5** に示す．

3.3 鍵 配 布

本システムを起動すると，まず，送信側は転送データの暗号化／復号に必要な共有鍵を作成し，受信側に配布する．

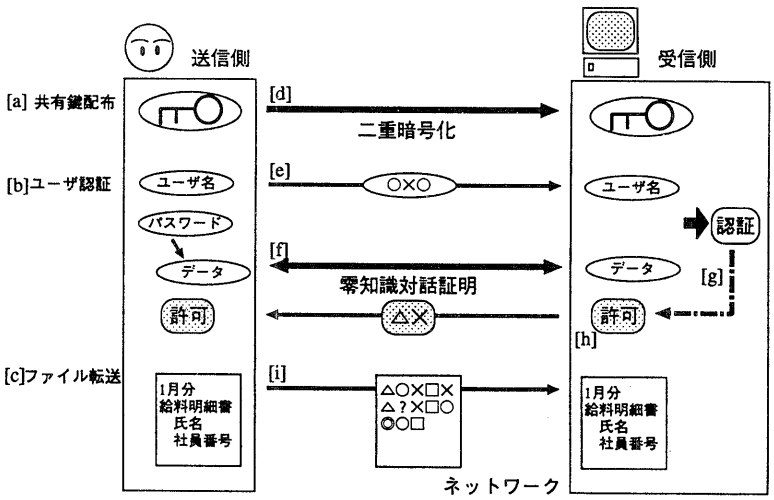


図 4 セキュアファイル転送システム
Fig. 4 Secure file transfer system.

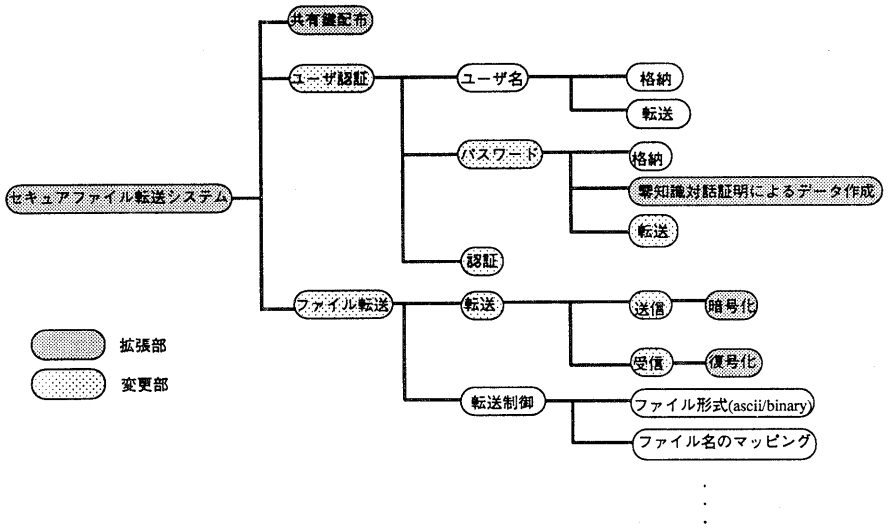


図 5 セキュアファイル転送システムの処理構造
Fig. 5 Process structure of secure file transfer system.

鍵配布には，Shamir, Rivest, Adleman による二重暗号化方式を用いている．以下の図を用いて，配布手順を述べる．

1. まず，送信側は，共有鍵 K を乱数により作成する（**図 6** の［a］）．
2. 次に，素数 p を用いて，
$$C_{rnd}C_{rec}\equiv 1 \pmod{p-1}.$$
となるような乱数 C_{rnd} およびその逆元 C_{rec} を作成する．
3. 次に，
$$K_c\equiv K^{C_{rnd}} \pmod{p}.$$
となる K_c を受信側に転送する（図 6 の［b］）．
4. 受信側は，
$$S_{rnd}S_{rec}\equiv 1 \pmod{p-1}.$$
となるような乱数 S_{rnd} およびその逆元 S_{rec} を作成する．
5. 次に，
$$K_{cs}\equiv K_c^{S_{rnd}}[=(K^{C_{rnd}})^{S_{rnd}}] \pmod{p}.$$
となる K_{cs} を送信側に転送する（図 6 の［c］）．
6. 送信側は，
$$K_s\equiv K_{cs}^{C_{rec}}[=((K^{C_{rnd}})^{S_{rnd}})^{C_{rec}}=K^{S_{rnd}}] \pmod{p}.$$
となる K_s を受信側に転送する（図 6 の［d］）．
7. 受信側は，
$$K\equiv K_s^{S_{rec}}[=(K^{S_{rnd}})^{S_{rec}}] \pmod{p}.$$
より，共有鍵 K を受けとる（図 6 の［e］）．

3.4 ユーザ認証

共用鍵の配布を終了した後，ユーザ認証を開始する．ユーザ認証には零知識対話証明を用いている．零知識対話証明とは，証明者が秘密の情報を明かすことなく，検証者と対話を行いながらその秘密を知っていることを検証者に対して証明する方法である．

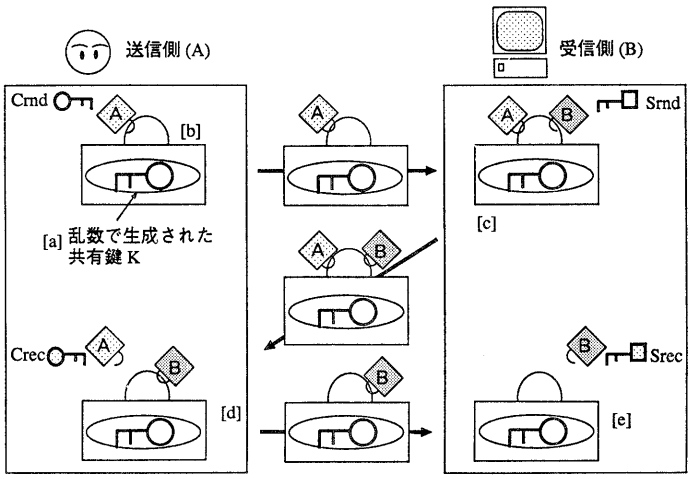


図 6 二重暗号方式による鍵配布
Fig. 6 Secret-key transferring based on a public-key cryptosystem.

本システムでは，そのうち Fiat-Shamir 法[8]を適用している．以下の図を用いて，その手順を述べる．

送信側は，事前にパスワード s を任意に決定しておく（**図7**の［a］）．そして，素数 p と $p-1$ 以下の素数 q から得られた $N=pq$ なる整数 N を用いて，

$$I \equiv s^2 \pmod{N}.$$

となる I を算出し，I と N をあらかじめ公開情報として，受信側に通知しておく．

本システムでは，ユーザがユーザ名とパスワード s を入力することによって，ユーザ認証を開始する．

1. 送信側はユーザ名を受信側に転送する（図7の［b］）．
2. 送信側は，乱数 r を選び，
   $$R \equiv r^2 \pmod{N}.$$
   となる R を受信側に転送する（図7の［c］）．
3. 受信側は，2進数 $e=0$ or 1 をランダムに選び，送信側に転送する（図7の［d］）．
4. 送信側は，e を受けとり，もし $e=0$ ならば，
   $$V \equiv r \pmod{N}.$$
   を，$e=1$ ならば，
   $$V \equiv rs \pmod{N}.$$
   を受信側に転送する（図7の［e］）．
5. 受信側は，$e=0$ の場合には，
   $$V^2 \pmod{N} \equiv R.$$
   を検査し，$e=1$ の場合には，
   $$V^2 \pmod{N} \equiv RI \pmod{N}.$$
   を検査する（図7の［f］）．
   検査式が成立しない時は，ユーザが正しくないことが確認されたので終了する．
   検査式が成立する時は，ユーザが正しいことが確認されたので，受信側はファイル転送を許可する（図7の［g］）．

なお，認証の信頼性を増すために，上記の情報交換を 50 回行っている．情報は，パラレルで転送することで，認証に要する時間を短縮した．もちろん，認証においては，パラレル転送での安全性は，シリアル転送の安全性と同等である．

また，送信側と受信側での各種情報の交換は，すべて共有鍵により暗号化されて転送される．転送された情報は，受信側により共有鍵を用いて復号される．

3.5 ファイルの転送

ユーザ認証が成功すると，ユーザはファイル転送を許可される．

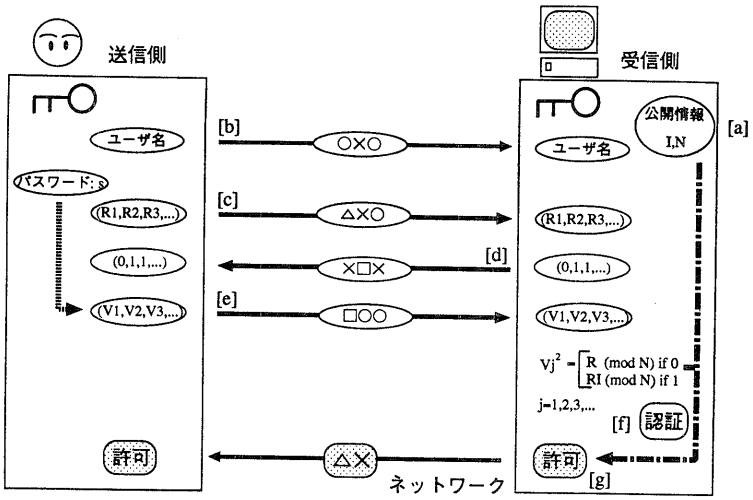


図 7 ユーザ認証
Fig. 7 User authentication based on a zero-knowledge identification scheme.

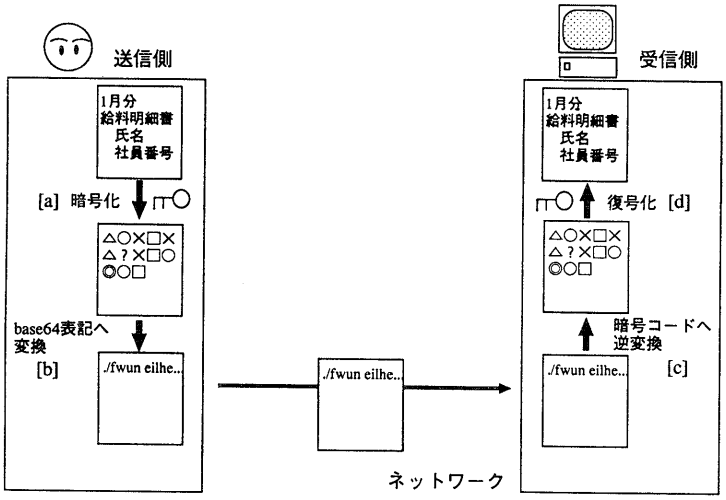


図 8 ファイル転送
Fig. 8 Secure file transferring.

図8を用いて，ファイルの転送手順を述べる．

1. 送信側は，あらかじめ受信側に配布しておいた共有鍵を用いて，受信側に転送するファイルを暗号化する（図8の［a］）．
2. 次に，暗号化したファイルを base 64 表記に変換し，受信側に転送する（面8の［b］）．
3. 受信側は，受けとったファイルを base 64 表記から暗号化ファイルへと逆変換する（図8の［c］）．
4. そして，共有鍵を用いて，暗号化ファイルを復号して，元のファイルを得る（図8の［d］）．

base 64 表記とは，ロング整数を6文字（各文字は，基数64表記の「数字」を表す．）までで表すことが可能な表記法である[9]．

また，今回採用した暗号方式は，独自に作成した64［bit］単位のブロック暗号方式である．

従って，暗号化する前のファイルサイズを a[Byte] とすると，暗号化したファイルサイズは，

$$a+(8-a \bmod 8)e \text{ [Byte]}$$

(a：8の倍数の時 $e=0$，a：8の倍数でない時 $e=1$)

さらに，暗号化ファイルを base 64 表記に変換するので，

$$\{a+(8-a \bmod 8)e\}(6/4)$$
$$=\{a+(8-a \bmod 8)e\}(3/2)$$
$$=(3/2)a+12e-(3/2)(a \bmod 8)e \text{ [Byte]}.$$

ファイルサイズに比べると $12e-(3/2)(a \bmod 8)e$ はほとんど考えなくてよいので，転送するファイルサイズは，

$(3/2)a$ [Byte].

となる．

3.6 システムの性能

本システムを当社のワークステーション ME/R 7200 (CPU：PA-RISC［25 MHz］) に実装し，性能測定を行った．結果は以下のとおりである．

3.6.1 処理時間

本システムを起動し，共有鍵配布時間，ユーザ認証時間，ファイル転送時間およびファイル転送時の暗号化／復号時間を実測した．

測定には実時間を用いた．また，誤差を除くために同じ測定を50回行って，その平均をとった．

結果は以下のとおりである．

- 共有鍵配布時間（素数 p：348 ビットとした場合）（**表1**参照）
- ユーザ認証時間（整数 N：256 ビットとした場合）（**表2**参照）
- ファイル転送時間（**表3**参照）

ただし，本システムのファイル転送時間には，通信経路の接続，データの暗号化，暗号文から base 64 表記への変換，base 64 表記から暗号文への逆変換，データの復号およびその他の処理時間も含まれている．また，暗号化／復号時間は，上記のファイル転送時間のうちデータの暗号化／復号にかかる時間を抜き出したものである．

上記から，鍵配布およびユーザ認証では，ほとんどユーザが意識しないで処理を行うことが可能である．また，ファイル転送においては，従来のシステムと比較すると，ファイルサイズが大きくなるにつれて多少速度的な問題はあるが，数百KB程度のファイルならば，実用可能な範囲であると思われる．

3.6.2 プログラムサイズ

本システムは，従来のファイル転送システムに共有鍵配布処理，零知識対話証明によるユーザ認証処理およびデータの暗号化／復号処理を追加し，拡張したものである．

使用した言語は，アセンブラおよびC言語である．アセンブラは，各処理の多倍長整数演算部分に使用し，それ以外の部分にC言語を使用した．

プログラムサイズは，送信側・受信側を合わせて，

従来のシステム……7 K step

追加した分……3 K step

となり，本システムのプログラムサイズは，10 K step である．

表1 共有鍵配布時間
Table 1 Secret-key transferring time.

従　来 (sec)	—
本システム (sec)	4.89

表2 ユーザ認証時間
Table 2 User authentication time.

従　来 (sec)	0.07
本システム (sec)	0.36

表3 ファイル転送時間
Table 3 File transferring time.

ファイルサイズ (KB)	32	64	128	256	512
従　来 (sec)	0.20	0.20	0.21	0.31	0.55
本システム (sec)	0.45	0.81	1.57	2.79	5.35
暗号化 (sec)	0.06	0.13	0.24	0.49	0.97
復　号 (sec)	0.06	0.12	0.24	0.48	0.96

4. おわりに

本稿では，従来のファイル転送システムの問題点および問題を解決するためのアプローチ例とその問題点を述べた上で，今回開発を行ってきたセキュアファイル転送システムの特徴，構成，各種原理および性能について述べた．今後は，ファイル転送の高速化やパスワードの初期設定および変更その他に対する処理の安全性について検討を加え，更にセキュリティ機能を向上したシステムとしたいと考えている．また，今回開発した技術を応用して，コンピュータネットワークサービス全般についての開発も進めていきたい所存である．

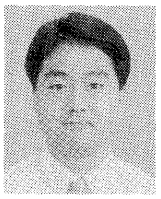
謝辞 最後に，本研究の機会を与えて頂いた三菱電機(株)情報システム研究所 曽我正和前所長，渡辺治前部長，中島邦男部長に深く感謝いたします．

参考文献

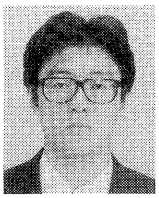
1) Bellovin, S. M. and Merritt, M.: Limitations of the Kerberos Authentication System, *ACM SIGCOMM Computer Communication Review*, Vol. 20, No. 5, pp. 119-132 (1990).
2) Davies, D. W. and Price, W. L.: *Security for Computer Networks*, John Wiley & Sons, Ltd. (1984). (上園忠弘 (監訳): ネットワーク・セキュリティ, p. 318, 日経 BP 社 (1990).)
3) 小林信博，岡本隆司，桜井幸一：零知識対話証明技術のコンピュータ間認証への適用，第 44 回情報処理学会全国大会論文集 (4), pp. 265-266 (1992).
4) Lunt, S. J.: *FTP Security Extensions*, p. 16, Internet Draft (1993).
5) Postel, J. and Reynolds, J.: *FILE TRANSFER PROTOCOL (FTP)*, p. 69, RFC-959 (1985).
6) Steiner, J. G., Neuman, C. and Schiller, J. I.: Kerberos: An Authentication Service for Open Network Systems, *Proceedings of the USEIXN 1988 Winter Conference*, pp. 191-202 (1988).
7) Shamir, A., Rivest, R. L. and Adleman, L.: Mental Poker, MIT Laboratory for Computer Science, Report TM-125 (1979).
8) Fiat, A. and Shamir, A.: How to Prove Yourself: Practical Solution to Identification and Signature Problems, *Proc. of CRYPTO '86*, pp. 186-194, Springer-Verlag, Berlin (1987).
9) AT & T UNIX ソフトウエアオペレーション パシフィック株式会社：UNIX System V プログラマ・リファレンス・マニュアル, p. 387, AT & T UNIX ソフトウエアオペレーションパシフィック株式会社 (1989).
10) 山口 英：ネットワーク環境のユーザ認証システム Kerberos, *Super ASCII*, Vol. 3, No. 2, pp. 73-88 (1992).

(平成 6 年 4 月 7 日受付)
(平成 6 年 7 月 14 日採録)

中川 和美 (正会員)
昭和 43 年生．平成 3 年 3 月東京女子大学文理学部数理学科卒業．平成 3 年 4 月三菱電機株式会社入社．現在，情報システム研究所に勤務．コンピュータセキュリティ，特にネットワークセキュリティに興味を持つ．



岡本 隆司 (正会員)
昭和 35 年生．昭和 58 年北海道大学工学部精密工学科卒業．昭和 60 年 3 月北海道大学大学院工学研究科精密工学専攻修士課程修了．昭和 60 年 4 月三菱電機株式会社入社．産業システム研究所で数値制御装置に関する研究開発を経て，現在，情報システム研究所にて，コンピュータセキュリティ，ネットワークアプリケーション，衛星通信ネットワークの研究開発に従事．精密工学会会員．



櫻井 幸一 (正会員)
昭和 38 年生．昭和 61 年九州大学理学部数学科卒業．昭和 63 年 3 月九州大学工学研究科応用物理学専攻修士課程修了．昭和 63 年 4 月三菱電機株式会社入社．情報システム研究所にて，暗号と情報セキュリティの研究開発に従事．平成 5 年 6 月博士(工学)の学位授与(九州大学)．平成 6 年 3 月より九州大学工学部情報工学科助教授，現在に至る．計算量理論，複雑性理論，分散ネットワークと情報セキュリティに興味を持つ．電子情報通信学会，日本数学会各会員．