マルチメディア通信と分散処理 79－18
（1996. 11. 15）

OTP を利用した遠隔ユーザー認証システムの実験的運用

山田 芳郎　林 秀郎†,　都築 伸二,　田崎 三郎

愛媛大学工学部
松山市文京町 3
hayashi@ee.ehime-u.ac.jp

現在の IP プロトコルは、ネットワーク層で暗号化などの安全対策がなされていない。このため、インターネットを経由した遠隔ログインは、システムセキュリティ上の大きな問題となる。このため、S/KEY などのような使い捨てパスワードを用いたユーザ認証システムが提案されている。本稿では、当研究室における OPT システムの運用結果と、S/KEY による認証のセッションを組み込んだ端末側ユーザインターフェースのプロトタイプについて報告する。

An Experimetal Operation of the OTP based Remote User Authentification System

Yosio YAMADA,　Hideo HAYASHI†,　Shinji TSUZUKI,　and　Saburo TAZAKI,

Ehime University, Faculty of Engineering
3 Bunkyo-cho, Matsuyama
hyashi@ee.ehime-u.ac.jp

Abstract: The current IP protocol does not support the encription of the packets at the network layer. This causes some serious difficulties with respect to the system security when the remote logins are done over the Internet. For this purpose, the One-Time Password (OTP) systems, such as the S/KEY system, are proposed. This paper presents an experimental operation of the OTP system in our laboratry, and also discusses a prototype of the user interface of the Telnet client with a built-in OTP based user authentification session.

1 まえがき

UNIX システムなどのマルチユーザ・システムにおいては, セキュリティ上の安全性を確保するための基本的かつ重要な手段の一つとして, 4 から 8 文字程度のパスワードを用いた単純なユーザー認証システム (以下単純パスワードシステムという) が実装されている. 現在の IP プロトコルにおいては, ネットワーク層において暗号化などのセキュリティ上の安全対策がなされていないため, ユーザー認証のためのパスワードが悪意を持った第三者に傍受される危険性が常に存在する. そして, パスワードを入手した第三者は容易に他人になりすますことができる. さらに, インターネット (Internet) を介して遠隔接続されるような場合には, 遠隔端末 – サーバー間にどのようなシステムが介在しているかは正確に知ることができないため, セキュリティの観点から安全な通信を保証することは困難である. 従って, 単純パスワードシステムを用いる場合には, パスワードを定期的に変更することが一般に奨励されている. この考えをさらに進めて, 一回切りのパスワード (以下使い捨てパスワードと言う) をログインする度ごとに自動的に生成する One-Time Password システム (以下 OTP と言う) が提案されている.

OTP としては, Bellcore 社の S/KEY [1], Navel Research Laboratory で開発された OPIE (one-Time Passwords In Everything) [2] などがよく知られている. OTP は, ユーザー認証の安全性を大幅に改善するという意味では単純パスワードと比べて優れたシステムであるが, 一方では, ログインごとに使い捨てパスワードを計算する手間がかかるなど, ユーザーにとって使い勝手の上で心理的な負担がかかるシステムであるともいえる. 本稿では, 当研究室において OTP を実験的に運用し, ユーザーから見た操作性について検討した結果を報告する.

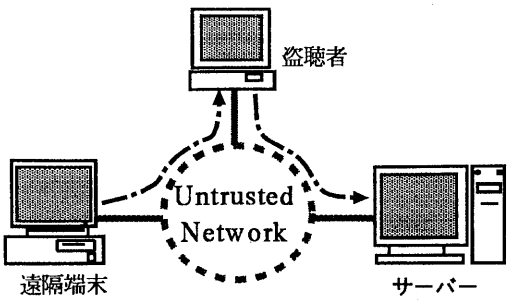

図 1: パスワードの盗聴

2 OTP [6],

最初に実用化された OTP システムは Bellcore 社の S/KEY システムであるが, 同システム名は Bellcore 社の登録商標となっているため, 現在では, 使い捨てパスワードの一般的な名称として OTP が広く使用されている. OTP における使い捨てパスワードは, パスフレーズ, シーケンス番号及び種の 3 つの情報から計算される.

- パスフレーズ

  ユーザーが持つ 10 文字以上 50 文字以内の文字列. (秘密情報)

- シーケンス番号

  一方向関数を適応する回数.

- 種

  種はパスフレーズと連結され, OTP サーバー毎に異なる文字列生成する.

OTP を用いた遠隔端末 – サーバー間のユーザー認証は, 以下のように行なわれる. (図 2 参照)

1. ユーザーは遠隔端末からサーバに接続を要求する.

2. サーバ はユーザー名に対する "種" と" シーケンス番号" を遠隔端末へ送る.

3. ユーザーは遠隔端末側においてサーバーから送られた "種" と "シーケンス番号" を用いて自分の "パスフレーズ" から "使い捨てパスワード" を計算し, サーバ へ送る.

4. サーバ は "使い捨てパスワード" の正当性を検査することによってユーザ認証を行う.

上述の認証プロセスにおいて, 使い捨てパスワードを計算するために, 一方向関数が利用されている. 一方向関数とは, x から y を計算するのは容易であるが, y から x を導くのは非常に困難な関数である. すなわち,

$$y = f(x) \tag{1}$$

は容易に計算できるが,

$$y = f(x\prime) \tag{2}$$

となるような $x\prime$ を計算するための逆関数 f^{-1} は存在せず, $x\prime$ を見つけるには可能性をすべて調べなければならない関数である. S/KEY では, 一方向関数として MD4 [3] が使用される. OPIE [2] では, MD4 [3], MD5 [4], SHA(Secure Hash Algorithm) のいずれかを選択できる. 例えば, "シーケンス番号" N に対する "使い捨てパスワード" f^N から次にログインするための "シーケンス番号" $N-1$ に対する "使い捨てパスワード" f^{N-1} を計算することは, 系列を一回下る必要があり, 計算はほとんど不可能である. 従って, "種" , "シーケンス番号", 及び "使い捨てパスワード" が他人に傍受されたとしても, 次にログインするための"使い捨てパスワード" が解読される可能性は非常に小さくなることが期待できる.

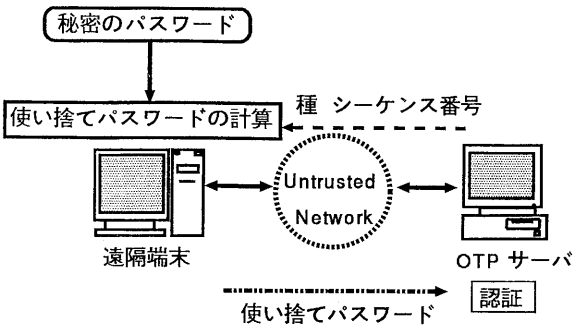

図 2: OTP 認証の概略図

3 遠隔端末におけるユーザーインターフェイス

本研究では, OTP を研究室内 LAN において実験的に運用した. まず, 1 台の UNIX ホストに OTP サーバー (S/KEY) を実装した. そして, その他研究室内の端末 (Windows, Macintosh, UNIX, MS-DOS) に使い捨てパスワードを計算するためのソフト (S/KEY, OPIE) を実装し, どの端末からも OTP サーバーへ使い捨てパスワードによる遠隔ログインが可能なネットワークシステムを構築した.

上述のようにサーバーにログインする際, 遠隔端末側で "種", "シーケンス番号" 及び "パスフレーズ " をもとに "使い捨てパスワード" を計算し, サーバーへ送る必要がある.

3.1 X window 環境

サーバーへ telnet でログインする場合, 図 3 に示すように, ユーザー名に対する "種", "シーケンス番号" が表示される. "使い捨てパスワード" は, 端末側で 別の window (図 4) を開き, そこで計算した "使い捨てパスワード" を図 3 の window にいき copy and paste する.

3.2 Windows 環境, Macintosh 環境

Windows 端末では, 図 5 に示す window で" 使い捨てパスワード" を計算し, 計算結果をマウスで copy and paste してサーバーへログインする. Macintosh 環境も同様.

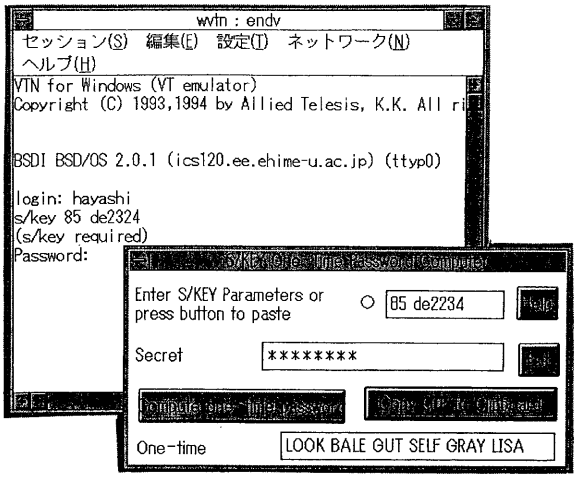

図 5: Windows 端末における "使い捨てパスワード" の計算

3.3 MS-DOS 環境

MS-DOS 端末では, あらかじめ "種", "シーケンス番号" を調べ対応する "使い捨てパスワード" を計算し, 出力された "使い捨てパスワード" をキーボードを用いて入力する.

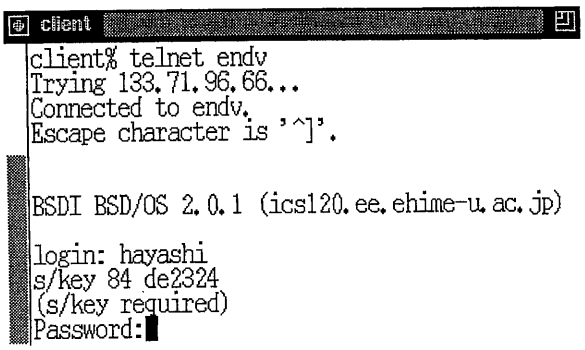

図 3: X 端末からのユーザー認証

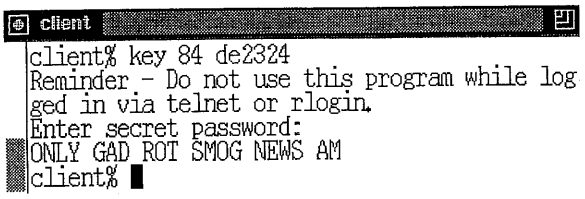

図 4: X 端末における "使い捨てパスワード" の計算

4 OTP の運用実験

日常的に UNIX-WS を使用しているユーザーを対象として, 学部生 11 人, 大学院生 9 人, その他遠隔地のユーザー 4 人の計 24 人に 7 日間の期間を設け "パスフレーズ" を登録してもらった. 登録期間中は従来の単純パスワードシステムと OTP を併用して行い, この間 OTP のガイドライン (印刷物) を配布し操作法の説明を行った.

図 6 に, OTP サーバーを実装した UNIX ホストの 1 日ごとのログイン失敗率及びログイン成功累計を示す. ここでログイン失敗率は,

$$\frac{\text{ログイン失敗回数}}{\text{ログイン成功回数} + \text{ログイン失敗回数}} \times 100$$

のように定義する.

また比較のため同時期に, 単純パスワードシステムの平均ログイン失敗率を測定した.

図 6 から単純パスワードと OTP の併用期間では, 単純パスワードシステムの平均ログイ

ン失敗率 (9.7 %) とほぼ同じ値を示しているが, OTP に完全移行する前日から急激にログイン失敗率が増え, 移行日には 77 % となった. また, OTP のみの運用に移行して一週間経ってもログイン失敗率が 20 % 前後と単純パスワードシステムの平均ログイン失敗率よりも高い値のままであった. これは, 本実験の被験者のように日常的に UNIX- WS を使用しているユーザーであっても OTP に慣れるためには比較的長時間の習熟期間が必要であることを示している.

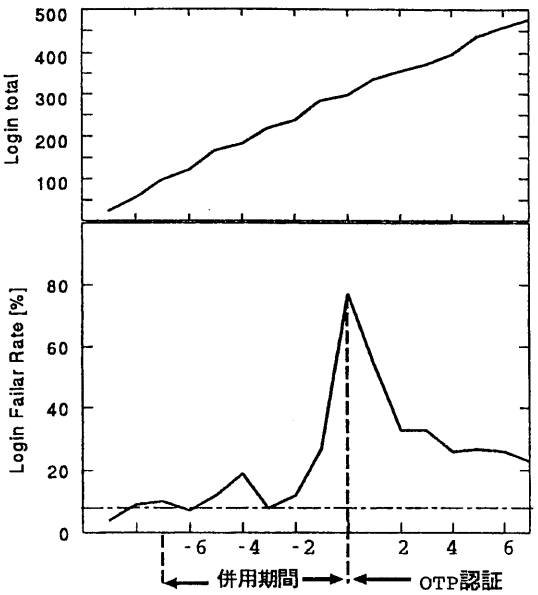

図 6: 一日毎の ログイン失敗率及びログイン成功累計

5 操作性のよい OTP 遠隔端末の検討

OTP を実験的に運用した結果, ユーザーから OTP の操作法及びセキュリティ上の必要性について理解を得ることができた. しかし, 単純パスワードシステムに比べると操作が複雑でログイン失敗率も高い結果となった.

そこで遠隔端末において, "種 "シーケンス番号" の入手, "使い捨てパスワード" の計算, 及び計算した "使い捨てパスワード" の入力を自動化し, ユーザーからの操作としては "パスフレーズ" の入力のみのクライアントシステムを試作した. これは telnet プログラムに "使い捨てパスワード" を計算するプログラムを組み込むことにより実現できる. この場合のログインプロセスは, 以下のようになる.

i. ユーザー名を入力する.

ii. サーバーから送られた"種", "シーケンス番号"を自動的に取り込まれる.

iii. ユーザーにパスフレーズの入力を要求する.

iv. 取り込んだ "種", "シーケンス番号" とユーザーが入力したパスフレーズをもとに使い捨てパスワードが自動的に計算される.

v. サーバーに送信される.

ここでは, Windows 上のパッケージソフトである SmarTerm を同ソフトスクリプト言語で拡張してプロトタイプを作成した. 図 7 にこのスクリプトを示す. このスクリプトは Windows 上でボタンを click することで実行され, 図 8 のように window が開き, パスフレーズを入力することによりログインが完了する. 以上のように従来の単純パスワードシステムとほとんど変わらないユーザーインターフェイスを実現できた.

今後は, telnet プログラムに "使い捨てパスワード" を計算するプログラムを組み込むことによりすべての端末において同様のシステムを実装する予定である.

6 むすび

本稿では, OTP を実験的に運用しユーザーから見た操作性について検討した. また, Windows 上で OTP によるログインに関しては, 従

来の単純パスワードシステムの操作性とほとんど変わらない操作性で OTP を利用できる telnet クライアントのプロトタイプを作成した.

今後は, telnet プログラムに "使い捨てパスワード" を計算するプログラムを組み込むことによりすべての端末において同様のシステムを実装する予定である.

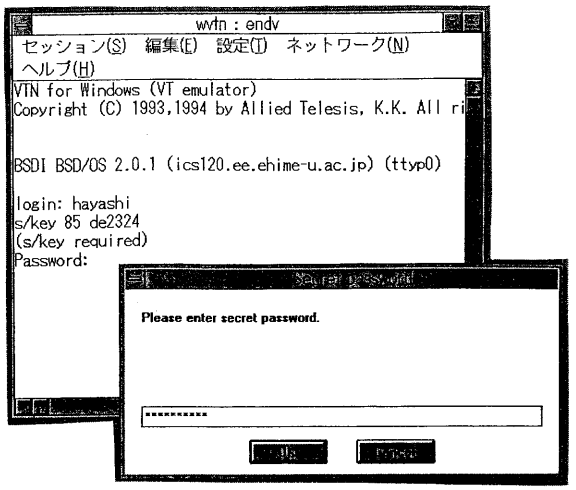

図 8: パスフレーズを入力する window

```
hostname$ = INPUT$("Please enter host name:","host name",1)
                                        ! Enter host name.

CIRCUIT  CONNECT  hostname$        ! Connecting to TCP/IP host.
 SEND  "< CR>"                  ! Send a <RETURN> to the host

SELECTWAIT  TIMEOUT  5     ! Wait for 5 seconds for
      MATCH  "login:":           ! a match with login prompt
          GOTO  login:

      EXPIRED:
          ERRORBOX  "Unable to connect"
                                        ! Error box is displayed
          GOTO  done
ENDSELECTWAIT

login:
    username$  =  INPUT$("Please enter user name :"
                         , "user name", 1)
                                        ! Enter user name.
    SEND  username$, "<CR>"  ! Send the login name
    GOTO  password

password:
     callenge$ =  SCREEN$ (-2 , -3 , 11)
          ! Capture  seed  and seekencens from screen
     spassword$ = INPUT$ ( " Please  enter  secret  password : "
                         , "Secret Password",1)
                                        ! Enter  secret  password.
     EXECUTE  KEY.EXE  & " " & callenge$ & " " & spassword$
                         ! Calculate one-time password and record
                         ! to the file name "c:\otp.txt"
     SEND " c : \otp.txt" , "<CR>"   ! Send  one-time password.

     ECHO CHR $(7)                  ! Ring the bell.

     OKBOX  "welcome, & username$   ! Acknowledge  login.
     GOTO  done

done:
stop                                    ! End the script
```

図 7: SmarTerm スクリプト

参考文献

[1] N.Haller, The S/KEY(TM) One-Time Password System, Proceedings of the Internet Society Symposium on Network and Distributed System Seculity, pp. 154–158, 1994

[2] D.McDonald and R.Atkinson, One-time Passwords In Everything (OPIE), Proceedings of the Proceedings of the Fifth USENIX UNIX seculity Symposium, 1995

[3] R.Rivest, The MD4 Message-Digest Algorithm, RFC1320,1992

[4] R.Rivest, The MD5 Message-Digest Algorithm, RFC1321,1992

[5] 山本 和彦 :" S/KEY ", UNIX MAGAZINE, 株式会社アスキー, pp.114–124 (1995.3)

[6] 山本 和彦 :" OTP (One-Time Password) ", UNIX MAGAZINE, 株式会社アスキー, pp.68–76 (1996.2)