

校内における情報セキュリティとWebサーバ

1 セキュリティ

セキュリティについて論じるのは中々難しいことです。何故ならば、非常に多岐に渡っている問題であり、同時に単なる技術のみに終らない話であるからです。例え、技術的な問題にのみ話を絞ってみても、ネットワークやその機器のセキュリティや、提供しているサービスのセキュリティ、あるいは個人個人のプライバシーの観点からのセキュリティなど非常に広範囲に渡ります。そこで、ここでは単にファイルをどうやって安全に保管し、共有するかということに話を絞りたいと思います。実際、学校などの現場には多くの情報がファイルという形で保管されています。例えば、住所や連絡先などの個人情報や、あるいは個人の成績、またはテスト問題や入試問題のような秘密性の高いものまであります。

こうしたファイルをどのように保持するのは、今や非常に重要な問題となりつつあります。それは単に多くの情報が電子化されたからというだけではなく、ウィルスやスパイウェア、ボットなどの情報を盗み出したり、誤って漏洩したりすることが以前よりも簡単に行われるようになってきており、ネットワークによって全てが繋がっている状態が日常的になって来ているからです。多くの人はこうした状況を認識しつつもそれがもたらす危険性への自覚に欠けているように思われます。Winnyなどのファイル共有ソフトを使わないことは (P2P 一般の問題というよりも、バグのあるソフトが未だに使われていることに対する無自覚と言った方が良いでしょう)、既に常識となりつつありますが、実際にはそうしたソフトを使わなければ、あるいはきちんとウィルスチェックやスパイウェアチェックをかけていれば安全かという、実はそうではないという点が知られざる今の危険性なのです。実際、スパイウェアやボットは年々巧妙化しつつあり、ボットの大半は未確認のものであると言われていています (ハニーボットなどで捕獲してみるとボットらしき動作をするが何のためのものか確認できない)。ウィルスがある種の自己顕示的性質を持つのに対して、スパイウェアやボットは徹底して隠密を常とします。こうしたものの作者はまず既成のウィルスチェック力などにかけ、駆除されないことを確認した上でネットに流していると言われており、何の対策も施さずにインターネット上に Windowsなどを曝すと、数分で感染するとされています。つまり、こうした状況下では最善の努力を行っても、ファイルは流出する可能性が常にあるということを前提にしなければならないということなのです。勿論、こうしたスパイウェアなどの標的はもっと秘匿性の高い、しかもありふれたものを攻撃対象とします。例えば、個人の銀行口座のパスワードなどが最も狙われる訳で、所有する情報が無限定にネット上に流出するということを必要以上に危惧するのもバランスを欠いているでしょう。しかし、一方では、危険性に対して何の対策も施さない状態を続けるならば、万一の場合に社会的に指弾されることもまた必定であると言えます。従って、本論では絶対に安全な方法を提起する訳でも、安全性を保証する訳でもありませんが、通常情報の世界で常識的な程度の安全を施すことを提起するものです。

1.1 ファイル共有

セキュリティの問題を考える前に、さまざまな情報をどのように共有するかという点について考えてみましょう。ここでは情報はファイルの形で保存されていると考えますから、一般にこれはファイル共有の手法として捉えられます。多くの学校現場ではファイル共有と言うと、Windowsのファイル共有 (CIFS: Common Internet File System) を利用しているでしょう。しかし、実は CIFS を利用した共有にはいくつかの問題があります。まず第一にそれは Microsoft の規格だという点です。従って、他の OS でそれを利用するのは不可能ではないのですが、それなりの技術レベルを必要とします。しかし、それ以上に問題なのは Windows には様々なバージョンがあり、そのバージョンによって共有に提供されるセキュリティモデルが異なっているという点です。例えば、XP にはホームバージョンとプロバージョンがあり、認証やファイ

ルアクセスへのセキュリティがホームバージョンでは制限されています。その結果、様々な Windows バージョンが現場で使われている場合には、セキュリティのない誰でもアクセスできるファイル共有を使わざるを得なくなります。勿論、そのためにネットワークセグメントを分離したり、より高度な分離手法を駆使する場合もあるでしょうが、本質的に誰でもアクセスできるファイルである時点でスパイウェアには全くの無防備ですし、誤った無線 LAN の運用による侵入の問題なども発生します。従って、こうしたファイル共有を利用して共有するファイルはもともとセキュリティレベルが高くないものであると思われませんが、それでもやはり一定のレベルが必要であると言えます。そして、そのような対象として CIFS を用いるのは現実には困難なのです。では、どうすれば良いかと言うと、標準的な手法を用いるべきだというのがここでの論点です。また、単なるファイル共有では整理できないという点も付随して指摘しておきましょう。

1.2 ウェブ

さて、では CIFS に代わって何を使えば良いかと言うと、ウェブを使おうというのがここでの提案です。実際、もともとウェブは情報を整理して、共有するために開発されたものなのですから、ウェブで校内の様々な情報を共有するのは自然なことなのです。ウェブというと常に外部に公開するように思っている方々もおられるかも知れませんが、そんな必要はありません。実際、きちんとウェブに外部公開のための対策を施すのは面倒なので、むしろ簡単に内部用のウェブを考えてやれば良いでしょう。同時に、無制限にアクセスできないように、最低限のセキュリティレベルが保つ必要がありますが、ウェブにはそれらをパスワードで実現することができます。その場合、グループで同じユーザ名とパスワードを共有しても構いませんし、ユーザ毎に異なるパスワードで、複数のユーザが一つのグループをなし、そのグループにアクセス可能なディレクトリ (フォルダ) を実現することもできます (ここで言うユーザは OS とは関係なく、ウェブ上のユーザの意味)。では何が問題なのかと言うと、ウェブのそうした管理が実は意外に面倒であるということなのです。ウェブ自体はサービスなので、様々な実現方法がありますし、実際ウェブを提供するシステムも色々ありますが、事実上 Apache が世界標準であると言っても過言ではありません。実際、Apache は日々改良され、セキュリティホールが発見されたならば即座にパッチが提供されます。Microsoft の IIS では、まだ潰されていないセキュリティホールや OS などに問題がある場合が少なくありません。結局、Unix 上で Apache を走らせるのが標準形式になっているのですが、残念ながら Apache は OS のファイルシステムを利用して作られているために、上に述べたようなアクセス制限をしようとする Unix を操作することが必要となります (近年は Webmin のようなウェブ上でシステムを管理するためのツールが進歩してきていますが、まだウェブのユーザアクセス制限は手動での作業が必要なようです)。

つまり、ウェブ自体はその中で閉じた体系であるにも係わらず、そのサーバソフトのために閉じていないのが問題なのです。

1.3 CMS: Contents Management System

ウェブが普及するにつれて、ウェブを管理する必要性は非常に拡大しました。その一方で、Unix をきちんと管理する人は少なく、またウェブの管理のために Unix 管理を学ばなければならないのは管理コストの上昇をもたらしますし、何よりも面倒です。そこで、ウェブをウェブの中で閉じた管理を可能にするソフトウェアが登場しました。それが、CMS (Contents Management System) です。CMS というと、blog や Wiki のようなシステムを思い浮かべる人も多いでしょうが、実際には CMS の方がより広い概念だと言えます。多くの企業のサイトは今や CMS によって管理されています。そうした巨大システムでなくとも、小さな管理の分散や委譲を行う上で CMS は有用です。実際、ウェブを管理するためにシステムの管理権限を与える必要がないのですから、ウェブの管理者とシステムの管理者は分離されます。更に、システムによっては、ウェブの管理者はその権限をページやフォルダの階層に従って分散委譲することもできます。

これは例えば、学校内のあるクラブのページは学生に任せると言うような利用が可能にもなります。

CMSとしては多くのシステムがありますが、ここでの目的は情報の比較的安全な共有ですから、blogやWikiのために結果的にCMS機能を持っているようなサーバシステムではなく、最初からCMSがメインで、その中の幾つかの機能の一つとしてblogやWikiがあるような、そして必要でない機能は導入しなくても良いようなシステムが望ましいでしょう。とは言え、EnterpriseクラスのCMSだと逆に複雑すぎて使いこなすのが難しいかも知れませんが、程良い程度が重要です。筆者も全てのCMSを知っている訳ではないので、ここでは次の2つのソフトを紹介するに止めます。

1. Zope

ZopeはCMSの初期に登場しましたが、非常に洗練されていながら、十分な機能を持っているためにここ数年日本でも急速に人気が出ており、解説書も数冊発行されています。当然、ページも全てウェブインターフェースを使って、作成できますし、管理もZopeの中で閉じています。多くのモジュールが開発されており、ほとんど何でもできると言っても過言ではありませんが、徐々にシステムが巨大化しつつあるのが、唯一の弱点でしょうか。

2. moodle

moodleは正確に言うと一般的なCMSではなく、LMS(Learning Management System)と呼ばれる分野に属するソフトです。LMSは言わばウェブ上で学習をするための支援システムで、学習者が相互に、あるいは教師とそのシステム上で接触を保ちつつ、学ぶために必要な機能が搭載されています。ニュースやフォーラム、カレンダー、教材の提示、アンケート、テスト、学習者管理、コース管理などを有していますが、CMSとしてもある程度の機能を持っているために、学校現場では非常に使いやすいシステムとなっています。実際、本学の東京サテライトでは社会人学生の自習支援のためのシステムとして使っていますが、一方ではシステムに関する管理者間の情報共有としても利用していますし、筆者のゼミではゼミ内部の情報管理にも利用をしています。

1.4 暗号と共有

ここまで共有される情報はそれほど高いセキュリティを必要とするものではなく、従ってその情報へのアクセスに制限をかければ十分であろうと考えて来た訳ですが、それ以上のセキュリティを必要とする場合にはどうすれば良いかという事を考えたいと思います。つまり、ここで考えるべきことは低度のセキュリティでは不足、より高いセキュリティをファイルに対して施すことであり、最悪の想定としては情報流出を考えるということの意味しています。当然、こうした場合には暗号が必要となるのですが、ここで再びWindowsでの状況について考えてみましょう。Windowsのファイルシステムの上ではEFS(Encrypt File System)が提供されています。EFSは公開暗号鍵方式(後述)を利用していますので、現時点ではその暗号が解読されることはほぼないだろうと考えられています。しかし、EFSには重大な問題点があります。それは、先にも触れたように全てのWindowsで利用できないと言う点(プロダクトによって異なる)、秘密鍵がユーザのマイドキュメントフォルダー内部に置かれ、他のユーザからはアクセスはできなくなっていますが、そのユーザのプログラムからは常に利用可能な点です。特に後者については、通常の公開鍵暗号方式における秘密鍵の秘匿方法では秘密鍵はパスワードで暗号化され、ユーザプログラムであってもパスワードにアクセスできなければ復号化できないようになっているのに対して、大きな欠点を持っています。つまり、そのユーザのプロセスとして、スパイウェアなどが感染すれば、そのスパイウェアは自動的にEFS上のファイルにアクセス可能であるということになります。つまり、ここでも暗号化は必要であるが、標準的な手法を採ることが、アクセシビリティの上でもセキュリティの上でも上策であるという訳です。では、暗号化において標準的な方法は何かと言うと、OpenPGP(Open Pretty Good Privacy)が上げられます。以下では、OpenPGPを紹介する前に、その基礎となる暗号について簡単に触れること

にします。

1.5 暗号

1.5.1 暗号システム

暗号システムとは簡単には以下のようなものです。

$$K(\text{平文}) \rightarrow \text{暗号文}$$

$$K^{-1}(\text{暗号文}) \rightarrow \text{平文}$$

ここで K は暗号鍵と言い、 K^{-1} は復号鍵と言います。例えば、パスワードを暗号鍵を使って暗号化し、それを逆に復号鍵 K^{-1} を使えば元のパスワードを取り出せるようなシステムになっています。

なお、分かりやすいように K を暗号鍵と言いましたが、数学的には K を復号鍵に使うことも出来ます。この場合には K^{-1} が暗号鍵になります。

$$K^{-1}(\text{平文}) \rightarrow \text{暗号文}$$

$$K(\text{暗号文}) \rightarrow \text{平文}$$

一般に、暗号システムでは、一つの暗号文が解読されただけでは、システムが破られたとは言わず、暗号鍵 K と復号鍵 K^{-1} が解読された時点で暗号が解読されたと言います (つまり、どんな暗号文も全部解読出来る訳ですから)。

1.6 対称暗号方式

暗号システムの内、先の暗号鍵と復号鍵に同じ鍵を用いる方式を対称暗号方式、あるいは共通鍵暗号方式 (秘密鍵暗号方式という場合もありますが、次の公開鍵暗号における秘密鍵と混同しやすい) と言います。

$$\text{対称暗号方式} \quad K = K^{-1}$$

つまり、このシステムでは鍵は一つだけを用いるのでこの鍵を暗号文に適用すると元の平文が分かるという仕組みになっています。

$$K(\text{平文}) \rightarrow \text{暗号文}$$

$$K(\text{暗号文}) \rightarrow \text{平文}$$

そのために、鍵を盗まれたら一巻のおしまいです。また、通信などには向かないのもこれが理由になっています。

例えば、Unix で長らく使われてきた DES (Data Encryption Standard) は、この対称鍵暗号方式の一つです。その後採用された AES (Advanced Encryption Standard) も対称鍵の一種です。また、高速でありながら、Free に利用可能なものとしては Blowfish などがあります。(ちなみに、AES に採用された Rijndael (J. Daeman and V. Rijmen) もロイヤリティフリーで、アルゴリズムも公開されています。)

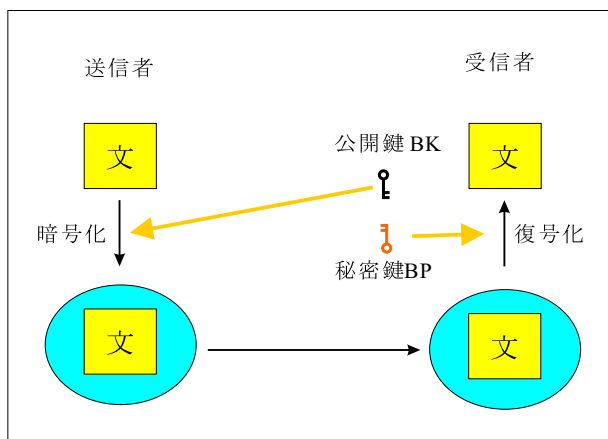
1.7 非対称暗号方式

非対称暗号方式は別名 **公開暗号鍵方式**と呼ばれています。この方式では、暗号鍵と復号鍵が異なります。

非対称暗号方式 $K \neq K^{-1}$

このために、暗号鍵を知られても、復号鍵が分からない限り暗号は解けません。また、最初に述べたように、復号鍵を使って暗号化したメッセージは逆に暗号鍵でないと解けないという構造になっています。この性質がネットワーク上でこの暗号システムを利用するのに非常に適している原因です。

一般には、暗号鍵、復号鍵という言葉を使うと混乱するので、2つのペアの鍵の内片方を **公開鍵** と呼び、他方を **秘密鍵** と呼びます。そして、公開鍵を誰もが読めるように公開しておき、2者の間で秘密の通信をしたい場合には以下のようにします。



まず、送信者 A は、B の公開鍵 BK を使ってメッセージ文を暗号化します。すると、この暗号化文は B の秘密鍵 BP でしか解読出来ないものですので、この暗号化文は誰に見られても構いません。同じように B が A にメッセージを送る場合には、逆に A の公開鍵 AK を取得し、それを使ってメッセージを暗号化して送れば良いわけです。

また、公開鍵方式で2つの鍵は対称な関係にあることを利用して、逆方向に利用することで電子認証などに利用することも可能です。

公開鍵暗号方式の代表は RSA です。RSA は開発者の名前を取って名付けられたものです (Ronald Rivest, Adi Shamir, Len Adleman)。RSA は非常に大きな数を素因数分解するのが大変であるという事実に基づいています。勿論、素因数分解のうまい方法が見つければ、この限りではありませんが、多くの数学者が長年かかっても、今のところ良い方法は見つかっていませんので、当面は安心して良いでしょう。

なお、RSA は非常に強力ですが、そのために計算コストも大きく全てを RSA で行うのは大変であると言われています。このために、多くの場合最初のやり取りのみ公開鍵方式を使い、そこで秘密鍵を交換して、その後は秘密鍵を使うのが一般的です。その意味で、秘密鍵方式も重要なので、先の AES にも意味はあるのです。(ちなみに、VPN などでは DES あるいは DES を 3 回用いる 3DES などが使われますが、全てのパケットにこうした暗号を用いるのは DES であっても、かなりの計算コストが伴います。つまり、VPN では結構この計算に時間がかかるために、スピードが出ない傾向にあります。)

1.8 OpenPGP

OpenPGP は、インターネットにおける標準規格である RFC2440 で定められています。元々は、PGP (Pretty Good Privacy) に源を発しますが、PGP が暗号化に IDEA という特許のあるアルゴリズムを利用したため

に PGP がフリーで商用利用できず、その部分に特許上問題のないアルゴリズムを利用した GnuPG が登場したために、標準規格が定められました。OpenPGP は通常メールの暗号化、署名などに良く使われるために、メール用と思われている方々も多いようですが、ファイルの暗号化、署名一般に使えます。OpenPGP は、電子署名、かいざん防止、暗号化などを提供し、それらは先に簡単に紹介したハッシュ値、暗号化は共通鍵暗号によって、共通鍵の秘匿に公開鍵暗号を、署名に公開鍵暗号の秘密鍵を用いた暗号化(つまり公開鍵で復号できる)によって実現します。メールで利用する場合には、例えば OpenPGP で暗号化したファイルを BASE64 など符号化し、添付します。

OpenPGP の実装の一つである GnuPG は全てロイヤリティーフリーのアルゴリズムを利用しているためにフリーで利用でき、多くの OS で稼働しています。また、Windows 上ではアドインソフトが存在するので、現在では比較的簡単に利用することができますし、多くのメールソフトでも対応がされつつあります。

1.9 PKS

公開鍵暗号における問題は、どうやって公開鍵を公開するのかという問題です。ウェブで用いられる SSL などでは電子証明を第三者機関 CA(Certified A) がそれを保管し、参照をするユーザーに(例えばブラウザ)に提供します。従って、この方式では第三者機関の正当性や、そこに蓄えられた電子証明の正当性などが問題となります。一方、OpenPGP など採っている方法は、これとは異なり、“信頼の輪”と呼ばれる方法で、ある人の公開鍵は世界中の鍵サーバ(PKS: Public Key Server)に置きますが、鍵サーバはその公開鍵の正当性を証明する訳ではなく、その公開鍵を他の人が電子証明により証明します。つまり、A さんの証明が正しいことを B さんが証明し、B さんの証明をまた別の C さんが証明します。このように“信頼の輪”によって示された正当性を信頼するか否かは、その証明を利用する人に委ねられています。こう書くと、あまり信頼できないように見えますが、これは我々が行っているごく普通の方法を電子的に再現したに過ぎません。

さて、以上で分かった通り OpenPGP では、鍵サーバに多くの人の電子証明が置かれ、世界中のサーバに自動的に配布されて広がっていきます。勿論、この鍵サーバを自分で勝手にたててもよく、また他の鍵サーバと電子証明を交換しなくても一向に構いません。実際問題として、校内でのみ運用するならば、公開鍵を「公開」しないような運用もあり得る訳ですので、その場合公開鍵を誰が保証するかという問題もあまり考える必要はなくなります。

なお、GnuPG などには鍵サーバなどから自動的に鍵をダウンロードする機能があり、独自に鍵サーバも少し登録するだけで常に参照をしてくれますので、公開鍵の管理は鍵サーバをたてれば、自動化されます。問題はグループで共有する秘密鍵をどのようにして秘密にするかですが、それはセキュリティレベルによって異なることになるでしょう。例えば、全員が見ることが出来るレベルならば、共通の秘密鍵を常に全員が持てば良いでしょうし、あるグループだけ、あるいは管理をきちんとしていた場合には秘密鍵を USB メモリなどに保管し、復号するときだけそれを取り出して復号してみるという管理も考えられます。

2 実習編

2.1 VMplayer の起動

この実習では、サーバとして FreeBSD を、クライアントとしては WindowsXP を仮想マシンを実現する VMplayer 上で動かします。まず、スタートメニューから、VMplayer を動かしてください。すると、ファイル選択が出るので、そこで `d:\security\FreeBSD` のフォルダーに移動し、`FreeBSD.vmx` を選択してください。VMplayer が起動し、その中で仮想マシンが動きます。待っていると FreeBSD の起動メッセージが出ますので、最後に `login:` プロンプトが出たら、VMplayer の窓の中をクリックします。この状態で、仮想マシンに入ったこととなります。仮想マシンから抜けるには、`Alt+Ctrl` キーを押します。この操作は、VMplayer では基本になりますので、忘れないで下さい。

```
login: root
passwd: *****
```

パスワードは `wakhok` です。ユーザ名は管理者を意味する `root` を入れます。うまくログインできたら、以下のコマンドを打ちます。

```
# ifconfig lnc0
```

この実習ではなるべくウェブインターフェースで実習を行いますが、最初だけ自分の IP アドレスを調べる必要があります。このコマンドを入力した後に、以下のようなメッセージが出ます。

```
lnc0: flags=108843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,NEEDSGIANT> mtu 1500
      inet6 fe80::20c:29ff:fe8f:cce8%lnc0 prefixlen 64 scopeid 0x1
      inet 192.168.0.110 netmask 0xffffffff broadcast 192.168.0.255
      ether 00:0c:29:8f:cc:e8
```

ここで、`inet` の後の数字が自分の IP アドレスです (正確には仮想マシンの IP アドレスです)。上の例では `192.168.0.110` です。

次に、WindowsXP を仮想マシンで立ち上げます。そのために、`Alt+Ctrl` を押して、一旦ホスト OS に抜けて、もう一つ VMplayer を立ち上げて下さい。そして今度は `d:\security\WindowsXP` にある `WindowsXP.vmx` を指定して動かして下さい。この状態で、3つの OS が動いています。VMplayer を動かしているホスト OS (WindowsXP) と、ホスト OS 上で動いているゲスト OS (VMplayer で動かしている OS をこう呼びます) の FreeBSD と WindowsXP です。何故、WindowsXP をゲスト OS で動かすのかと言うと、管理者権限を持った状態でアプリケーションなどを入れたいからです。

さて、ゲスト OS の WindowsXP が立ち上がったら、ブラウザを動かして、URL に以下のアドレスを入力して下さい。但し、ここで使う IP アドレスは先に調べた自分の FreeBSD の IP アドレスです。

```
http://192.168.0.110:10000/
```

このブラウザでアクセスしているのは自分の仮想マシン上で動いている FreeBSD 上で動作しているウェブインターフェースの管理用ツール `Webmin` です。この管理ツールを使って FreeBSD を設定します。

`Webmin` の `Username` と `Password` には、`admin` と `wakhok` を入力して下さい。



うまく入ると以下の画面になります。



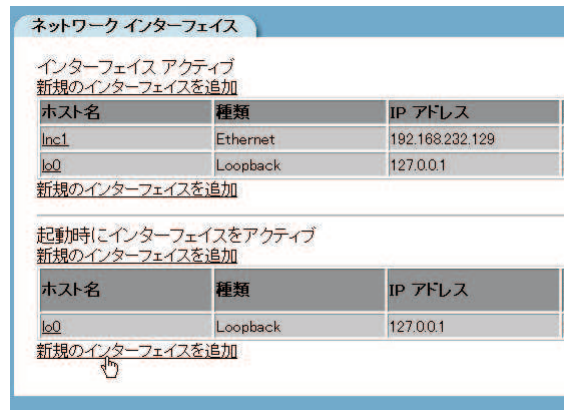
Webmin ではかなりの設定が全てウェブインターフェースからできます。もし、日本語表示でないならば、言語とテーマ設定で個人選択の項目から、**Japanese(JA_JP.EUC)** を選んで、**変更する** ボタンを押せば、日本語表示に設定が変更されます。

まず、上部のアイコンにある「ネットワーク」をクリックし、その次の画面で右下の「ネットワーク設定」を選択します。

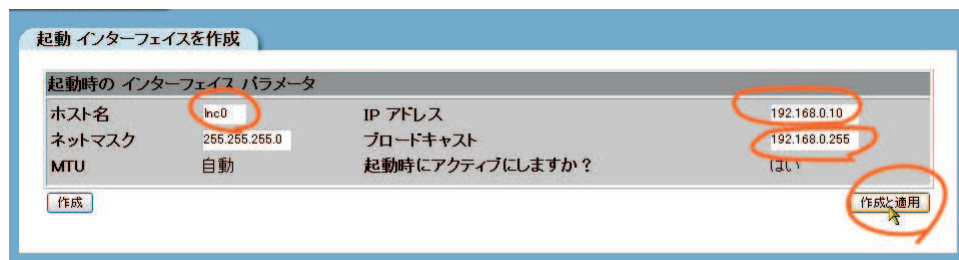


そして表示されるネットワーク設定画面で、ネットワークインターフェースを選択し、次の以下の画面

の起動時にインターフェースをアクティブの項目の一番下にある新規のインターフェースを追加をクリックします。ここを間違えないようにして下さい。



そして、以下のように IP アドレスを自分のマシンに与えられた IP アドレスに変更します。また、ブロードキャストアドレスは全員 **192.168.0.255** です。



ここは間違えないで下さい。他の人の IP アドレスにぶつけると、自分も相手もネットワークにアクセスできなくなります。最後に、右下の**作成と適用**ボタンを押すと、設定が保存され、適用されます。

これで、自分の FreeBSD の IP アドレスを固定のアドレスに変更できたので、FreeBSD をリブートします。仮想マシンの FreeBSD に入って、コマンドで **reboot** と打つか、Webmin からは**その他のコマンドシェル**で **reboot** を実行させます。

FreeBSD がリブートすれば、以降は固定 IP アドレスになりますので、名前でのアクセスが可能になります。

2.2 moodle を用いた管理

時間の関係で Apache におけるアクセス制御については最後にありますので、時間の余った方はトライして見て下さい。ここでは moodle を CMS として使う方法について実習を行います。

まず、仮想マシン上の WindowsXP のブラウザから、自分の FreeBSD サーバにアクセスします (既に moodle が動いています)。URL は、以下のものを使って下さい。

<http://fpc01.wakhok.ac.jp/moodle/>

ホスト名は先に決めた IP アドレスに対応するリストにあるホスト名です。

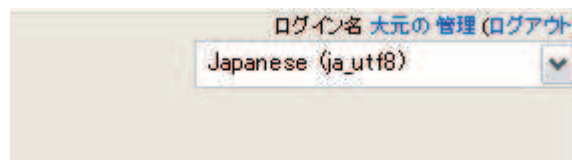
アクセスするとログイン画面が出ますが、既に管理者については設定されています。先に利用したパスワードと同じですが、慣習的にユーザ名は **admin** を使います (パスワードは **wakhok**)。



ログインすると以下のような画面になります。



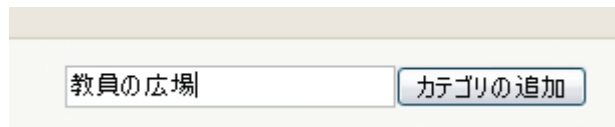
もし、日本語表示されていない場合は、右上の設定を以下のようにします。



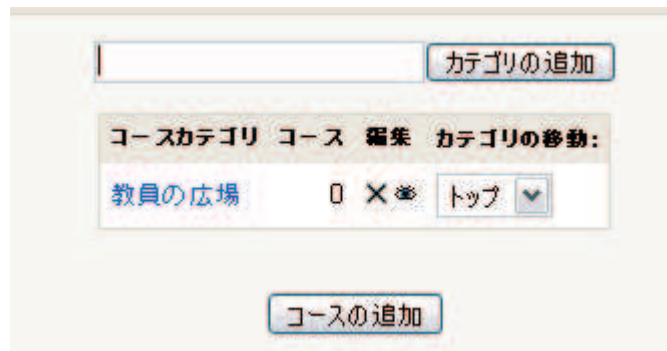
moodle の管理全体は左側にある管理ブロックで行えます。



また、現在見えているページ全体の構成を変えたい場合や、オブジェクトを追加する場合には、右上の編集 ON のボタンを押します。

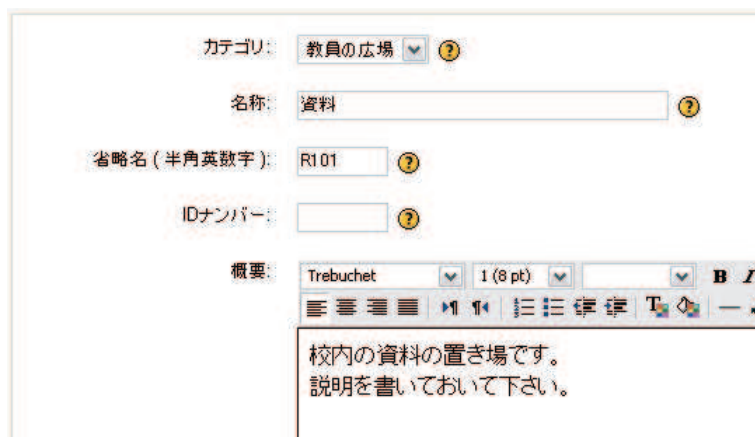


ここでは、教員全員で参照し、ファイルなどがアップロードできるページを作ってみましょう。まず、管理ブロックで**コース** をクリックし、その画面で、**教員の広場** を入力し、カテゴリの追加ボタンを押します。すると以下のように教員の広場がカテゴリに追加されます。



moodle では、コースと呼ばれるページはカテゴリの中に格納できます。カテゴリはフォルダーと同じで、カテゴリの中にカテゴリも作れます。さて、次はコースを作成しましょう。カテゴリの下にある**コースの追加**ボタンを押します。

カテゴリが教員の広場であることを確認して、名称、省略名概要を入力し、フォーマットはトピックフォーマットを利用するのが良いでしょう。フォーマットは、他にソーシャル、ウィークリーがあり、分からない時には右にある？マークをクリックすれば説明が得られます。



ここで大事なのは、コース利用に「学生が利用できない」を選ぶことです。学生というのは moodle が e-ラーニングのツールであるためにそうした用語を使っているからですが、簡単に言えば moodle に登録された一般ユーザのことであると思えば良いでしょう。逆に、権限のない人には moodle を使わせない設定にしておけば、ここは学生が利用できるにしても元々アクセス出来ないのですから、心配はいりません。

コース登録キーは言わばコースのパスワードで、一般ユーザがこのページを利用できないようにするもう一つの方法です。今の場合には、一般ユーザは利用出来ないのです、設定しなくても構いません。

開講日:	29	September	2006	?
利用有効期間:	無制限	?		
週外ピックの表示数:	10	?		
グループモード:	グループ無し	?	強制: No	?
コース利用:	学生はこのコースを利用できます。	?		
登録キー:	<input type="text"/>	?		
ゲストアクセス:	ゲストを許可しない	?		
セクションの非表示:	非表示セクションを折りたたんで表示する	?		
ニュース項目の表示数:	5 件のニュース項目を表示	?		
評価を表示:	Yes	?		
活動レポートを表示:	No	?		
最大アップロードサイズ:	2MB	?		

なお、moodle のデフォルト設定はアップロード出来るファイルのサイズは最大 2Mbyte までです。これを変更するにはソースレベルでの変更が必要ですが、通常の文書や資料をアップロードするには十分でしょう (手動で FTP などをすればこの限りではありませんが)。

設定が終れば、一番下の**変更内容を保存**ボタンを押します。

ここではまだユーザを割り当てていないので、コースを管理する教師もまだいません。そこで、ユーザを作ってみましょう。

まず、左上のリンクのトップをクリックします。(下の例では、**moodletest** がトップになります。)

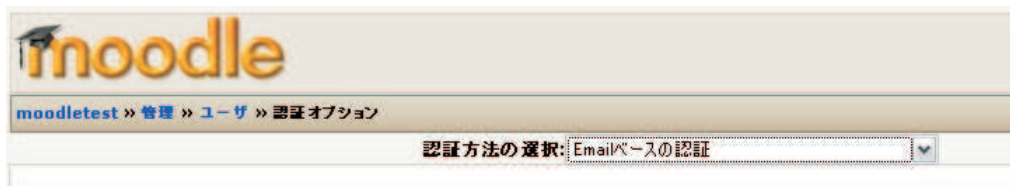


トップの管理メニューから**ユーザ**を選びます。

ユーザ

- 認証オプション** 内部のユーザアカウントまたは外部のデータベースを使用できます。
- ユーザアカウントの編集** ユーザアカウントリストの表示および編集を行います。
- ユーザの追加** 新しいユーザを作成します。
- ユーザのアップロード** テキストファイルより新しいユーザをインポートします。
-
- ユーザ登録方法** ユーザ登録管理を内部的に行うか、外部より行うか選択してください。
- 学生の登録** コース内の管理メニューより学生を追加します。
- 教師の割当て** アイコンを使用して選択したコースに教師を割り当てます。⑧
- コース作成者の割当て** コース作成者は新しいコースの作成およびコース内での教育を行うことができます。
- 管理者の割当て** 管理者はサイト内で全ての作業および移動を行うことができます。

ここで、まず認証オプションを変更しましょう。というのは moodle ではデフォルトでは誰でもが自由に自分を登録することが出来、E-mail を使って確認をすることになっています。これを許可されたユーザだけがアクセス出来るようにします。



これを**手動アカウント作成のみ**に変更します。
また、**ゲストログインボタン**も非表示にしておきます。

一般設定

ゲストログインボタン:

代替ログインURL:

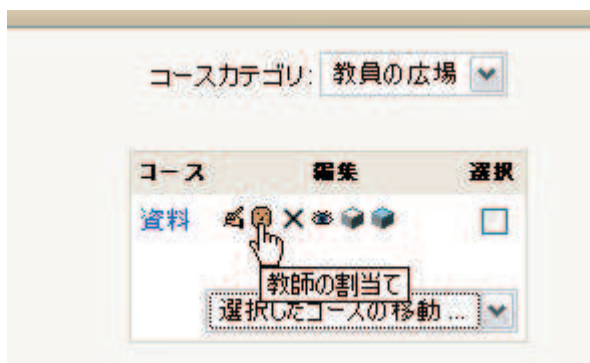
結果を保存し、トップメニューのユーザ (管理>>ユーザ)に戻ります。moodle では、管理者、コース作成者、教師、学生という権限階層があります。コース作成者は自分でコースを作成でき、自分と同じコース作成者を割り当てたり、教師をコースに割り当てたりできます。一方、教師はコースの中のみを管理します。これらの管理者、コース作成者、教師を割り当てるには、まず学生を登録しなければなりません。その上で学生を教師や、コース作成者に割り当てる訳です。ちなみに、教師や学生という言葉が違和感がある、あるいは間違えると思われる場合には、コース設定の中で呼び方を変えることができます (コース毎の設定です)。

ここでは、ユーザ **wakhok** パスワード **hoge hoge** を例に作成し、資料コースの教師に割り当ててみて下さい。割り当てたら、一度ログアウトして、wakhok ユーザーで入って、資料置場にトピックを追加したり、テキストを置いてみて下さい (編集 ON を押さないで編集出来ません)。

なお、ユーザ登録では、必要項目が入力されていないと必須の入力項目ではその右側に赤字で入力する旨が表示されます。国や都道府県なども入力が必要で、少し面倒ですが、仕方ありません (日本はドロップ

ダウンリストの下の方にあります)。メールアドレスもメールアドレスらしきものを入力しないと文句を言われます。

ユーザを登録したら、次に、wakhok ユーザを教師に割り当てます。管理ユーザメニューの教師の割り当てで教師に割り当てます。この部分は少し分かりにくいのですが、カテゴリを選ぶとコースが見つかります。ここで単にコースをクリックすると、そのコースに入ってしまいます。教師を割り当てるには下の図のように顔マークをクリックする必要があります。



後はリストから教師を選ぶだけです。さて、うまく行きましたでしょうか？うまく行ったら、一度ログアウトして、wakhok で入り、コースを編集してみてください。コースの編集をするには、右上のコースの編集 ON を押さないと出来ません。

トピックの数は後で追加して増やすことも出来ます。編集モードではトピックの中に色々なリソースを追加出来ます。例えばテキストを書いたり、ファイルをアップして、それへのリンクを張ったりが出来ます。ここでは試しに、適当なファイルをアップロードするために、

テキストの追加 -> ファイル・サイトへのリンク

を選んでみます。そして下の画面で、名称、概要などを適当に書き、ファイルの選択またはアップロードをクリックします。

すると、フォルダ画面になりますので、もしサブフォルダーを作成したい場合はここで作成します。そのままここにアップロードしたい場合には、**ファイルのアップロード**をクリックし、**参照**ボタンを押して、後はファイルダイアログを使ってアップロードしたいファイルを選びます。アップが終れば、アップしたファイルを選択することで、自動的にロケーションが設定され、トピックの中に新たにオブジェクトが置かれます。

他にも様々な形式のファイルを置いたり、リンクを張ったりすることが出来ますし、直接 HTML ファイルを書くこともできます。

ここまで出来て、他の人よりも進んでいるようでしたら、今度は今度は自分のユーザを作成して、管理者に追加して下さい。今度は管理者ですので、moodle の管理自体が行えます。

このようにして、moodle を管理ツールとして使うことができ、例えばある委員会のページを作りたければ、その委員を教師にし、ほかのユーザはそこに置かれたリソースにアクセスするだけという状態を作れます。e-ラーニングのツールであるために、少し違和感もありますが、予めコースなどのカテゴリーやアクセス制限がかかっているために意外と資料の閲覧や開示、お知らせなどにはうまく使うことが出来ます。

2.3 GnuPG の使い方

GnuPG は大分以前からありましたが、最近になって Windows 用の良いツールが出来、以前よりもかなり使いやすくなっています。既に、仮想マシンの Windows のデスクトップの上に置いてあります。

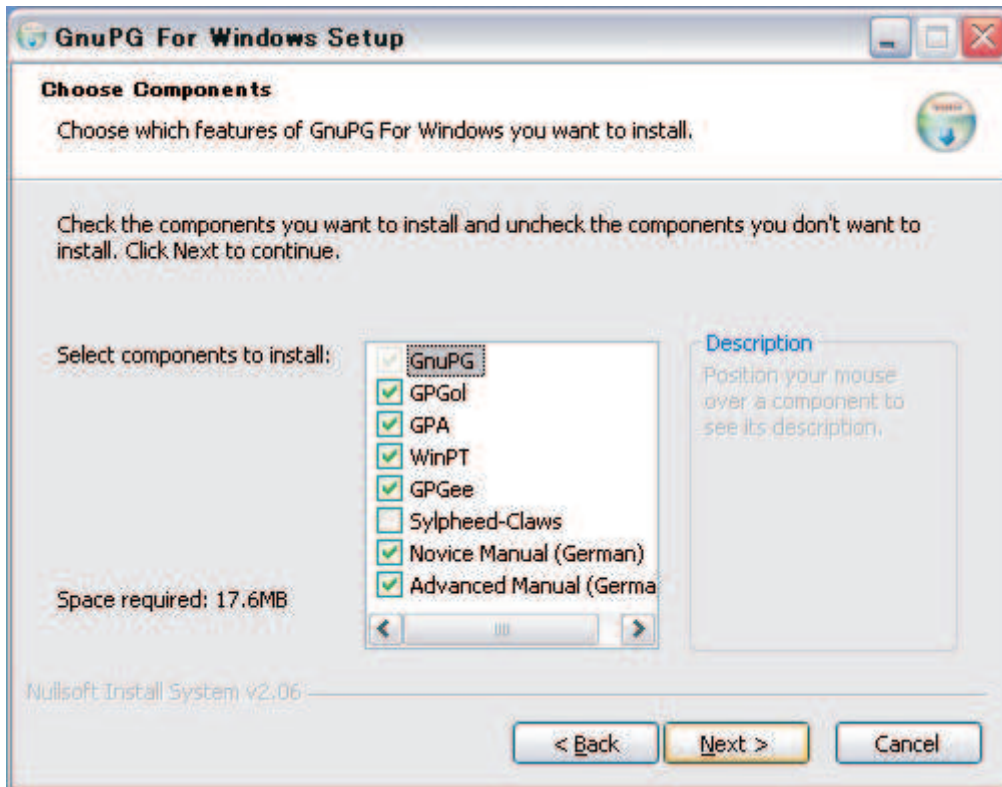


がそうです。ウェブでは、以下のところにあります。

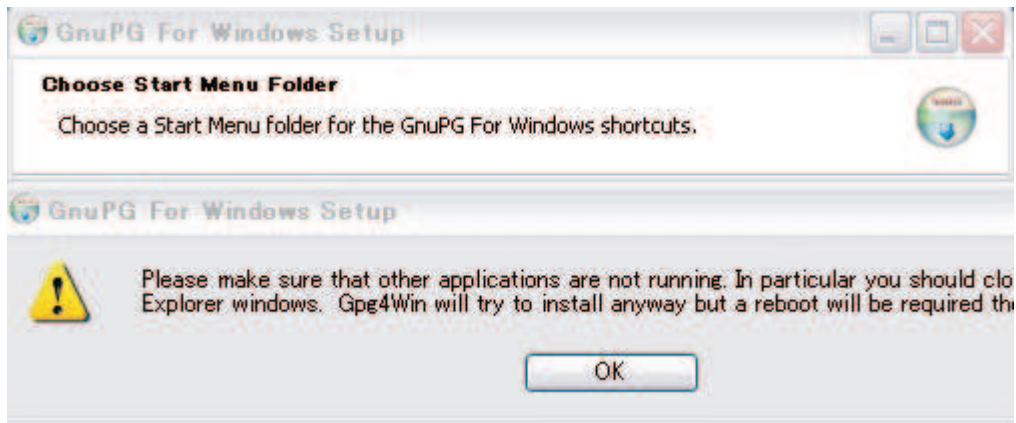
<http://www.gpg4win.org/>

残念ながら、ドキュメントなどはドイツなのでちょっと読めづらいのですが、色々な GnuPG 関係のソフトと GnuPG を詰め合わせにしたインストーラなので、非常に簡単に導入が出来ます。

インストーラを動かし、Next を押していくと、以下のようにどのモジュールをインストールするかを聞かれます。ドイツ語マニュアルが必要なければ、最後の2つは必要ないでしょう。また、Sylpheed-Claws は Sylpheed というメールクライアント用のプラグインです。Sylpheed を使わないのであれば、必要ありません。そのほかのツールはそのまま入れておけば良いでしょう。

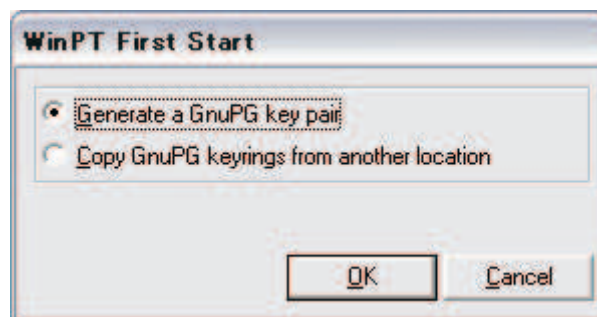


更に進めていくと、ショートカットをどこに作るかなどを聞かれますが、デフォルトで良いでしょう。すると、以下の注意が出てきます。



これは他のアプリケーション、特に Explorer を全て終了させておけという注意です。フォルダーを開いたりしていたら、全て閉じてから OK を押します。最後にインストールの終わりにリポートするようにメッセージが出ますので、リポートさせましょう。

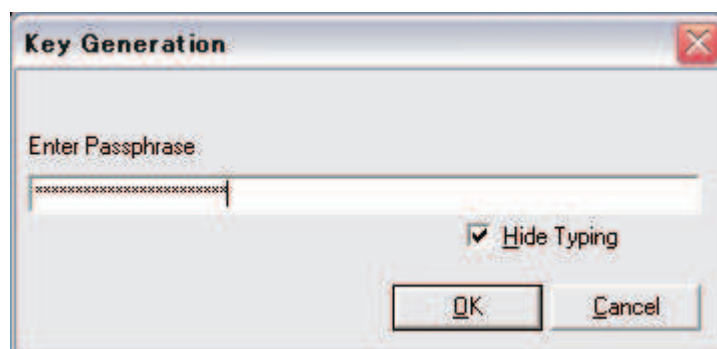
リポートしたら、スタートメニューの中の GnuPG for windows から WinPT を動かします。WinPT は 鍵を管理し、鍵のアップロードや検索、作成をしてくれる常駐ソフトです。スタートアップに入れておくと良いでしょう。WinPT を最初に動かすと、以下のように鍵ペアを作成するか聞いてきます。



次に、名前 (Real name) と Email address が聞かれます。この情報は他の人があなたの公開鍵を検索したり、識別するために用いるので、自分の正確な情報をローマ字で入れます (漢字は残念ながらご法度です)。一番下の 'Prefer RSA keys' をチェックすると DSA ではなく、RSA で鍵を作成しますが、デフォルトのままでも構いません。



次に、pathphrase を聞かれます。これは秘密鍵を自分の PC 上とは言え、そのまま置いておくと盗まれた時に困るので、秘密鍵を共通鍵暗号で暗号化するための言わばパスワードですが、パスワードと異なるのは空白を含む長い文字列を使える点です。なるべく長くて、忘れないパスフレーズを入力し、再度間違いがないかどうか聞かれます。



これが終わると、少し時間がかかりますが、鍵の生成が始まり、終了すると、'Key Generation completed' というメッセージが出て、鍵の生成が終了します。すると次に、生成した公開鍵と秘密鍵の鍵束 (keyrings) をバックアップするか否かを聞かれます。

これは必ず'はい'を選択してください。バックアップをどこに置くかは自由ですが、どこに置いたとしても本格的に利用する場合には、このバックアップをフロッピーや CD-ROM その他の場所に置いておきます。もし、HDD が故障したりして、秘密鍵をなくすと以前の鍵束を廃棄しなければなりませんし、公開鍵で暗号化したファイルも見えなくなります。

バックアップを選択すると、公開鍵、秘密鍵をそれぞれ保存する画面になりますので、適当なところに保存してください。

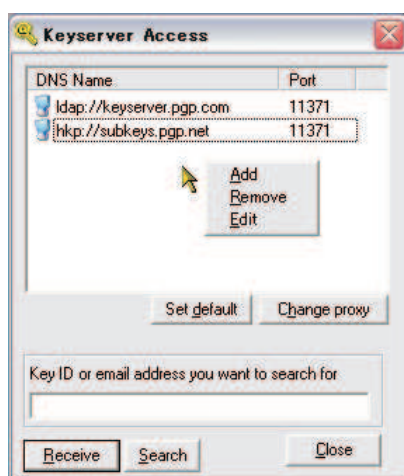
終了したら、常駐している WinPT を右クリックして、メニューを出し、KeyManager を左クリックします。

User ID	Key ID	Type	Size	Cipher	Validity	Trust	Cr
Noriyo Kanayama <kanayama@wakhok.ac.jp>	0x50691864	pub/sec	1024/2048	DSA/ELG	Ultimate	Ultimate	20

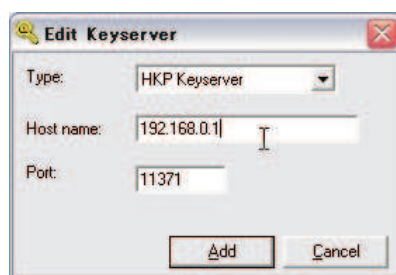
すると、自分の鍵が表示されます。

次に、自分のこの公開鍵を鍵サーバに登録します。鍵サーバは、192.168.0.1 です。WinPT を使うと自分の鍵を簡単に鍵サーバに登録できます。

先の KeyManager の Keyserver メニューをクリックすると、キーサーバのリストが表示されます。このリストの中で右クリックをすると、以下のようなメニューが出てきます。

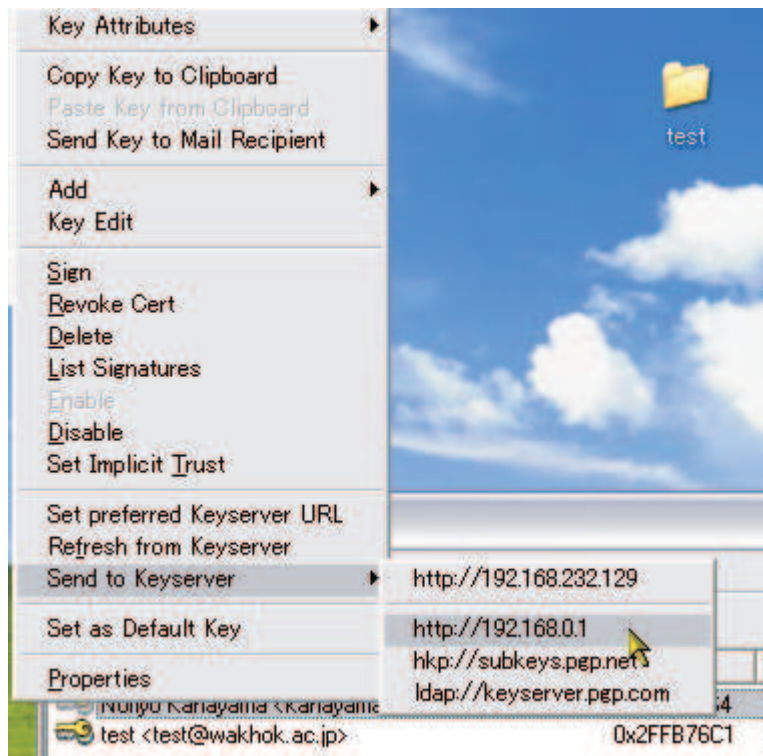


このメニューの add を選ぶと、新しい鍵サーバが登録できます。実習では仮の鍵サーバを用意していますので、その IP アドレスを入力して下さい。(他の鍵サーバは全部消してしまっても構いません。)



ちなみに、このダイアログボックスで鍵サーバのリストから、欲しいリストを探すことも出来ます。

さて、登録が終わったら、一旦 **Close** して、今度は自分の鍵の上でマウスを右クリックします。すると、以下のようにメニューが出てきますので、'Send to Keyserver' にマウスを持っていき、先程入力した IP アドレスのサーバを左クリックします。



すると、少し時間がかかりますが、自分の公開鍵をサーバにアップロードします。

さて、では自分の回りで終わった人を探して、その人の登録した E メールアドレスをきいて、その人のキーをダウンロードしてみましょう。

ダウンロードは先の Keyserver メニューで、ダイアログボックスを出し、そこに E メールアドレスを入れて Search(検索) するか、あれば Receive します。成功すると、KeyManager のリストにその人の公開鍵が表示されるはずですが。これで準備が整いました。いよいよ暗号化をしましょう。

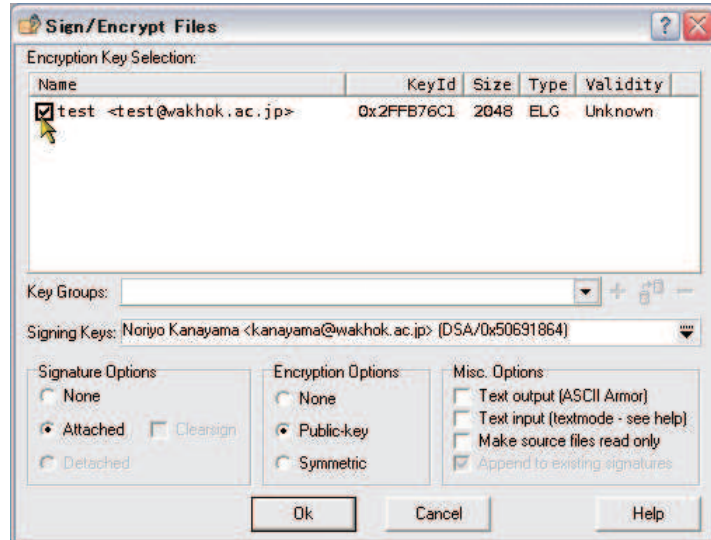
まずは、ノートパッドなどで適当なファイルを作成してください。但し、ファイル名は半角英数字でつけてください(漢字ファイルは流石に駄目です、中身は漢字も大丈夫です)。

出来たら、そのファイルの上で右クリックをしてみてください。



上のように GPGee がメニューに加わり、署名 (Sign) と暗号化 (Encrypt)、あるいは共通鍵による暗号化 (symmetric) などが出来るようになっていきます。

最初の **Sign&Encrypt** を選びます。この際に注意するのは、署名は自分の秘密鍵で署名し、暗号化は相手の公開鍵で行うという点です。GPGee では、自分の公開鍵は暗号化リストには出てきませんので迷いますが、署名に関しては自分が持っている全ての秘密鍵が出てくるので、注意が必要です。



上のリストボックスにあるのが、相手の公開鍵です。一方、下の 'Signing Keys' に出ているのが自分の秘密鍵です。相手の公開鍵が複数リストにある場合には、暗号化したい相手の公開鍵にチェックを入れてください。最後に OK を押すと、自分の秘密鍵を共通鍵暗号で暗号化しているのを復号するためにパスフレーズを聞いてきますから正確に入力をする、署名と暗号化をしてフォルダー内に、暗号化ファイルが出来ます。デフォルトでは、前のファイル名の後ろに .gpg をつけたファイルを作成します。

さて、このファイルをノートパッドなどで覗いてみましょう。本当に暗号化されているのが分かります。このファイルは、もう相手の秘密鍵でないと復号できません。そこで、相手のマシン名を聞いて、そのマシン上の moodle にユーザ wakhok でログインします (パスワードは hoge hoge です)。そこに、自分からの暗号ファイルであることが分かるように説明を書いて、ファイルをアップしてください。終わったら、相手にそのことを伝えて、ダウンロードしてから復号できるか確かめて貰います。

自分も、誰かから暗号化ファイルがいたら、それをダウンロードして、先と同じように右クリックで GPGee のメニューで Decrypt(復号) をしてみましょう。

さあ、うまく行ったでしょうか？

GPGee は自分の公開鍵で暗号化は出来ないようですが、WinPT ですと、自分の公開鍵でも暗号化できますので、WinPTの方が用途は広いでしょう。(自分の公開鍵で暗号化すると、自分の秘密鍵でしか復号は出来ませんので、誰にも見られないことが保証できます。)

2.3.1 PKS への登録

実習では既に鍵サーバを用意していました。実は、皆さんの仮想マシン上の FreeBSD でも pksd が既に動いています。FreeBSD では、pksd はパッケージにありますので、それをインストールして以下のように設定ファイルを書き換えます。

```
# cd /usr/local/etc/  
# cp pksd.conf.sample pksd.conf  
# vi pksd.conf
```

(vi でなくても、別のエディタでも構いません。X-windows 上には Gedit などもあります。)
ファイル変更点は一箇所、

```
#www_readonly 0
```

上の行の頭の # を削除して、pksd を再起動します。

```
# /usr/local/etc/rc.d/pksd restart
```

2.4 Apache とアクセスコントロール

ディレクトリ毎にアクセス制限を管理するには、`.htaccess` ファイルを各ディレクトリに置くことで行います。この方法では、apache の設定ファイルを変更する場合に比べて、再起動などの必要性がなく柔軟性があります。まず、`/usr/local/etc/apache22/httpd.conf` で設定すべき内容について説明しましょう。

2.4.1 AccessFileName 指定子

ディレクトリ毎のアクセス制御をする時に各ディレクトリに置かれて読み込まれる設定ファイルの名前を指定する。

```
AccessFileName .htaccess
```

`httpd.conf` には上記の様にデフォルトで `.htaccess` が指定されているので、変更する必要はありません。

2.4.2 AllowOverride 指定子

上述の `AccessFileName` 指定子で指定されたファイルが各ディレクトリで見つかった時、どのような設定について上書きを許すかをキーワードで記述します。キーワードは複数指定する事も出来ます。

```
AllowOverride キーワード
```

キーワードには次のものがあります。

None 上書きを許しません。

ALL 全ての種類の設定について上書きを許します。

AuthConfig 認証に関する設定について上書きを許します。

Limit ホスト名や IP address によるアクセス制御の上書きを許します。

Options Options 指定子で設定する機能について上書きを許します。

FileInfo ディレクトリ表示の設定について上書きを許します。

2.4.3 httpd.conf の例

/usr/local/www/apache22/data/test 以下のディレクトリで .htaccess の設定が可能になる場合を考えます。

```
<Directory /usr/local/www/apache22/data/test>
AllowOverride AuthConfig
</Directory>
```

2.5 .htaccess の設定

ディレクトリに以下のような .htaccess ファイルを書いて置く場合を考えます。

```
AuthType Basic
AuthName usercheck
AuthUserFile /usr/local/etc/apache22/users
AuthGroupFile /usr/local/etc/apache22/groups
<Limit GET POST>
require user user1
</Limit>
```

ここで、/usr/local/etc/apache22/users などは htpasswd コマンドを利用してパスワードを作成します (これは apache 用のパスワードでシステムとは関係ありません)。

パスワードファイルを最初につくりたいときには、以下のように、-c オプションを付け加えます (-c は create の略)。

```
# cd /usr/local/etc/apache22
# htpasswd -c users test
Adding password for test.
New password:
Rectangle-type new password:
```

上の例では、/usr/local/etc/apache22 に移動した後、そのディレクトリに users というパスワードファイルを作成し、ユーザ test を新たに作成しています。パスワードは 2 回聞いてきますので、間違えないように入力しましょう。

なるべくパスワードファイルを作りたいディレクトリに移動してから作成したほうが良いでしょう。

一旦、パスワードファイルを作れば、2 回目からは -c オプションは必要ありませんが、パスワードファイル名は忘れないようにしてください。

```
# htpasswd users test2
Adding user test2
New password:
Rectangle-type new password:
```

実際に作成される /usr/local/etc/apache22/ の users ファイルは以下のような形式になっています。

```
test:MdYVTQRDvFTU2
test2:WY5Vrq/rzSDIA
```

Unix の passwd ファイルと似ていますが、第 1, 2 フィールドしかない簡潔なものです。master ファイルと同様に読み書きの権限には注意しなければなりません。

同様に groups ファイルは、上のユーザを列挙したもので、unix の /etc/groups ファイルと同じようなものだと思えば良いでしょう。

もし時間があれば、自分で以上の設定を行い、

```
http://fpc01.wakhok.ac.jp/test/
```

にアクセスすると、ユーザとパスワードを利用しないとアクセスできないことを確かめてください。