関西大学学術リポジトリ
Kansai University Institutional Repository

# ITモラル教育と情報セキュリティ実習

| | |
|---|---|
| その他のタイトル | Laboratory Syllabi of Computer Security for an IT Moral Education |
| 著者 | 江澤 義典, 小林 孝史 |
| 雑誌名 | 情報研究 : 関西大学総合情報学部紀要 |
| 巻 | 21 |
| ページ | 59-77 |
| 発行年 | 2004-10-30 |
| URL | http://hdl.handle.net/10112/6735 |

# ITモラル教育と情報セキュリティ実習

江澤　義典[*1]　　小林　孝史[*2]


## 要　旨

大学における情報教育には何が期待されているだろうか．実践的なコンピュータ利用技術の習得についてはコンピュータ関連の専門学校に任せるとして，大学の文系学部や総合情報学系の学部では，情報教育として何を教えるべきであろうか．関西大学総合情報学部において科目「基本ソフトウェア実習」を，学部の創設以来担当してきた視点から，いわゆるITモラル教育と情報セキュリティ実習について考察する．


# Laboratory Syllabi of Computer Security for an IT Moral Education

Yoshinori EZAWA[*1], Takashi KOBAYASHI[*2]


## Abstract

What should be the nature of moral education in information technology (IT)? In this paper, we discuss it from the view point of the philosophy of social morals. It is shown that the scientific comprehension of IT constitutes the fundamentals. Then, the reason why a laboratory session for IT education might be required is discussed. Finally, the laboratory syllabi of computer security education at the Department of Informatics is explained as it is a unique precedent among Japanese colleges.



[*1] 関西大学総合情報学部
[*2] 関西大学総合情報学部

## 1．はじめに

大学における情報教育には何が期待されているだろうか．文部省(現文部科学省)の学習指導要領が改訂され，2003年度以降に高等学校普通科に進学する生徒は，科目「情報」を必修科目として履修することになった．したがって，2006年度以降の大学新入生は，ある程度の情報教育を受けていると想定してよい．その文部省が発行している指導要領の解説書[1]によると，初等・中等教育では，情報モラル教育が重要であると説明されている．実践的なコンピュータ利用技術の習得についてはコンピュータ関連の専門学校に任せるとして，大学の文系学部や総合情報学系の学部では，情報教育として何を教えるべきであろうか．関西大学総合情報学部において必修科目「基本ソフトウェア実習 Ⅰ，Ⅱ」を，学部の創設以来担当してきた視点から，いわゆるITモラル教育と情報セキュリティ実習について考察する．

## 2．ITモラル教育とは

ITモラルとは，最先端情報技術を用いる市民のモラル[2]をいう．もちろん，情報技術に関わる専門家のモラル[3]は社会的に重要であるが，ここでは将来の世代の中心となる一般的な市民のモラルを想定して，学生や生徒を対象に実施するモラル教育を考える．

### 2.1 モラル教育の原点

モラルといえば道徳と同義語である．そして，ほとんど全ての道徳理論は公平性の考えを含んでいる点に注意すべきである．その基本思想は，各個人の利益が等しく重要であるというものである．ゆえに，我々一人一人は，他人の幸福が我々自身の幸福と同じように重要であることを，認めねばならない．公平性の要求とは，言葉を変えれば，恣意性に対する禁止に他ならない[4]．

情報技術教育に関して，上園は次のように述べている[5]．『ある行為が正しいか正しくないか，あるいは適切な行為であるか不適切であるかは，教えられなければわからないものである．』ここで，適切さは，その行為による他人への影響を考えることで判断することになるが，その相手が具体的にイメージできる状況とイメージし難い状況があることに注意する必要がある．とくに，インターネットにアクセスしている場合には，世界規模のコンピュータネットワークに接続したコンピュータを利用しているので，自分の行為が影響を与える可能性のある相手を具体的にイメージできるためには，コンピュータ科学に関する素養が必須になる．

また，上園は，小学校や中学校の情報教育では，「他人をみたら泥棒と思え」というスタイルの教育に陥りがちなことを憂い，「君子は独り居るを慎む」という観点の情報モラルが望ましいことを示唆している．この言葉は『他人が見ていない所でもその行いを慎む』ということであり[6]，情報モラル教育の原点といえよう．さらに，金谷の注釈[7]によると，『いわゆる

その意を誠にすとは，自ら欺くなきなり．ゆえに，君子は必ずその独を慎むなり．』つまり，『自分で自分をごまかさないこと』が誠実ということであり，君子（道徳を学ぶもの）は『必ず内なる己れ自身を慎んで修める』のである．これらの言葉は，昔から子供たちによく言われた，「おてんとうさまがみているよ」という言葉と，全く同一の趣旨であることが興味深い．

## 2.2 IT とモラル

IT とは Information Technology の略であるから，直訳すれば情報技術ということになるが，情報通信技術や情報記録技術，情報保存技術，情報複写技術，情報加工技術，などの総称である．

太古の昔から，情報通信技術の課題は，送受信者間の，時間的制約と空間的制約を克服することであった[8]．すなわち，生の音声による会話が可能な範囲は，特別な手段を使わない限り，高々数十メートル以内の即時的なものに限られるのであり，その限界を克服する技術がさまざまに工夫されてきたといえる[9,10]．

実際，文字による情報記録の歴史は古い．その後，15世紀になって，印刷術が普及し，現代に至る書籍文化の基盤技術になった．また，19世紀以降には，電話やレコード・映画・ラジオ・テレビ・ビデオなど電子技術を応用した発明が続き，広く普及している．

その結果，最近になって，とくに情報モラルの重要性が強調されるようになった背景には，高度情報通信技術とコンピュータ技術を応用した，インターネットの普及がある．すなわち，通信の発信者にとっては，受信者の物理的な位置が特定できず，受信時刻も不明な状況での情報発信が可能になってきているのである．また，通信の受信者にとっても，発信者の物理的な位置が不明でありかつ，発信時刻も分からないケースが多くなってきた．例えば，電話で話をしている相手が，海外からも送受信可能なグローバル対応型の携帯電話を用いているのであれば，通話先の電話番号が判っても，その人がその時点でどこの国にいるかが直ちに分かる訳ではない．日本国内にいるときと同じ番号にダイアルするだけで，そのまま滞在先の携帯電話に接続されるのである．電子メールを受信した場合には，その発信者の ID や発信サイトは解読できるが，実際に誰がどこから発信しているかは確認できないことが多い．いわゆる，WebMail を利用している場合には，Web サーバの物理的(地理的)な位置と，Web にアクセスしている利用者の操作端末の物理的(地理的)な位置との関係は，受信した Mail のヘッダ情報だけでは判別できないのである．WWW の場合には，不特定多数の受信者を意識して，コンテンツを工夫しなければならない．そして，WWW から得た情報の信頼性や信憑性をどのように評価するかは，受信者が備えている常識の正確さや教養の深さに関わってくる．

結局，時間的制約および空間的制約を克服できる通信手段の発明により，我々は通信の相手が誰であるかさえも確認できなくなりつつある．もちろん，主観的な直感だけでは，通信で得た情報に対する正当な判断ができないのも当然である．

### 2.3 IT教育

情報教育について，文部省の指導要領[1]を紹介したが，そこで，強調されているのは「情報の科学的理解」である．では，情報の道具であるコンピュータを教材と考えたとき，その科学的理解を目指すには，どのような教育が可能であろうか．

コンピュータを操作してデータを入力するときに最も一般的だと思われるツールは，キーボードである．そのキーボード装置の科学的理解とはどのようなものであろうか．もちろん，キーボードのキー配列を暗記することは，実際にタイピング技術を向上させるとき以外には全く無意味な行為といえよう．現時点で広く普及している，いわゆるQWERTY配列が，「人間工学的に優れている」という誤った解説を掲載しているWebサイトがあるが，そのような不適切な説明に対して，学生自身が科学的にきちんと反証できるような教育が大切であろう．そして，人間工学的に改善されたDvorak配列がなぜ普及しないのか，将来のキー配列はどのように変化するだろうかといった見識を養うことが肝要である[11]．そのためには，実際にコンピュータのキーボードに触れて，タッチタイピング練習などの実習経験を積むことが有用である．このようにキー配列の開発史を踏まえた文化的総合的な理解が進めば，キートップの刻印部を部分的に交換するなどの悪戯がいかに稚拙であるかも了解できるだろう．

一方，コンピュータを使って文書を作成したり，表計算ソフトウェアでデータ集計をし，プレゼンテーション支援ソフトウェアを使うことなどの簡単なコンピュータ操作は，高校普通科の科目「情報」の授業にとって基盤となる技術である．先進的な教育を逸早く実施している一部の小学校や中学校では，「総合的な学習」や既存科目の学習課程にこのようなソフトウェアを活用したコンピュータ実習を導入して，生徒たちの興味を引き付けている[12]．しかし，コンピュータを操作したりインターネットにアクセスすること自体に，新奇性があるという初期段階を過ぎれば，このような手段だけでは生徒たちの興味を持続させることは困難になるに違いない．

全国レベルでの高校普通科における情報教育の実状を把握するには，実際に高校現場で展開されている授業を参考にすべきであるが，まだまだ，多くの高校におけるその取り組みは試行錯誤の段階といえる．とはいえ，その教科指導は文部省が作成した指導要領に従っているのであり，その解説書[1]は，それらの指導内容の枠組みを理解するためには重要な参考資料といえる．

## 3．コンピュータ科学と実習教育

学校教育においては，複数学生を相手の一斉授業が前提となっている．その教室には黒板や机以外にも，最近では，ビデオ装置やパソコンなどが，並んでいるかもしれない．しかし，授業で教えなければならない大切な事柄は多様であり，教師のおしゃべりや漫談が，学生たちの学習にとくに実効があるわけではない．

教科「情報」の指導要領にも解説されているのであるが，コンピュータを用いた実習教育の重要性が強調されている[1]．実際，情報Aでは授業の半分以上を実習にあてることになっており，情報Bおよび情報Cでも3分の1以上の時間を実習時間とすることが求められている．

黒板と机だけを使った，既存タイプの教室だけでの授業には致命的な欠陥がある．たとえば，生命の尊さを教育する授業は，どのようにすれば良いだろうか[13,14]．人生において最も大切な概念であるにも関わらず，学校や大学などの教室において，教師が教壇で，図解したりビデオを使ったりして，その重要性を強調しても，それだけで生徒や学生の理解が進むとは思えない．むしろ，実際に身近な生命の喪失を体験する機会があれば，そこで多くの生徒たちや学生は実感できるのではないだろうか．また，私有財産保護の大切さを教育する場合はどうだろうか．この場合も，教室での説明より，実際に財布を紛失したときや，自分で使える「お金」がなくなったときに，初めて実感できる[15]と思われる．では，自宅のセキュリティ（安全性）についてはどうだろうか．留守中に空き巣被害にあった者は，戸締りの重要性を実感できるだろう．情報セキュリティについても，状況は似ていると考えてよい[16]．実際，コンピュータウィルスの被害にあった者は，ワクチンソフトウェアの有用さが実感できるし，ファイアーウォールに守られたLAN環境の便利さに気づくことになる[17]．

### 3.1 科学的理解のポイント

私たちの日常生活においても，それが初めての観察や初めての経験であれば，どのように判断すればよいか解らず，戸惑うことが多い．しかし，その背景にある現象や原理などの科学的理解がすすめば，適切な判断ができるようになる．一般的に，直感に頼るのではなく，客観的で適切な指標や測度（measure）を考え，対象となる物体や装置の動作やその現象を表現できるモデルを構築することが重要である．しかし，様々な実際の現象に適合した正確なモデルを個別に発見するのは容易ではない．最初は単純なモデルから初め，少しずつ，改良版のモデルを工夫していくことで，対象の理解が徐々に洗練されていくのである．

ソフトウェア工学の重要なパラダイム（paradigm）として，適切なメトリックス（metrics）の活用とモデルの段階的詳細化（step-wise refinement）があり，その理解と習得は，コンピュータ科学を学習するうえでの目標の一つである．

### 3.2 コンピュータの実習教育

実習授業の目的は講義形態の授業の理解を助けるという役割が大きい．机上の空論ではなく，実際的な知識を体験し，さらなる学習の進展を図るのが目的となる．

体育実技や外国語会話の授業では，学生自身が自らの身体や発声器官を使って，実践する技能の習得が目標になる．しかし，数学の演習や理科の実験，社会科の調査などは，そこで使われる様々な技法の応用を部分的に体験することによって，学生が自発的に学習を進める助けになることが期待されている．

同様に，コンピュータ科学の学習においては，理論的な背景を学習するのと併行して，簡単なコンピュータ実習を実施することは，とても有意義である．一方，様々な企業が開発した応用ソフトウェアの習熟訓練をすること自体には，コンピュータ科学の学習に益する点はほとんど無い．むしろ，偏狭な仕様の細部に拘るという意味で，コンピュータ科学のパラダイムを学習する妨げになる点が多い．

コンピュータの使い方として，インターネットを使ったWWW検索や電子メールの交換は，学生にとっては，携帯電話を用いて既に体験済みであることが多くなってきた．学生たちが「パソコンでもメールやネットができるんやぁ」という感想を漏らすのは，実習クラスでは珍しくない．また，レーザプリンタによる印刷出力について「コピーする」と表現する学生が多いのも事実である．これらは，街角のコンビニエンスストアなどで電子複写機を日常的に使っている結果であろう．このような便利な情報機器を支える理論的な背景を学習することは，今後ますます発展するであろう，電子機器について科学的な理解を深めるために必須である．しかし，わざわざレーザプリンタの使い方を練習する実習が必要になるわけではない．

また，コンピュータの応用として，ワードプロセッサを用いて原稿の清書を行うとか，表計算ソフトウェアを用いて単純なデータ集計ができるのは，実用的にとても便利である．しかし，これらの道具の使用法は，実際に当該ソフトウェアを操作してみれば，その基盤となる考え方は単純な原理に基づいていることが直ちに理解できるものであり，わざわざ大学の教室で実習授業をする必要はない．もちろん，これらのソフトウェアに習熟すること自体は，ペン習字の練習やソロバン訓練が我々の現実生活で極めて有用であるのと同様に，コンピュータを活用する実生活の上では大変有用である．しかし，ソロバン訓練によって暗算が得意になることと算数や数学の学習とが無関係であることや，ペン習字の練習によって文章の清書が綺麗にできることと国語の学習とが無関係であること，はいうまでもない．同じように，情報にかかわる科学的な理解を深める上で，ワードプロセッサや表計算ソフトウェアの実習が必要だというわけではない．

## 4．情報セキュリティ実習

高校現場で実際に使用されている教科書[18]から，情報セキュリティに関する事項を拾い上げてみると，下記のとおりである．ここでは，とくに全国的に開講クラスの多い教科「情報A」のケースを紹介する．

1．情報の信頼性を考える：デマ，ねずみ講，Webページの改竄
2．メール発信の責任：チェーンメールを受信したときに回送してはいけない理由
3．情報モラルを考える：プライバシー，ダイレクトメール
4．コンピュータの取り扱い：不正アクセス禁止の意味，ファイアウォールの効果，コンピュータウィルスの被害

5．知的財産権の意味：著作権，など

たしかに，このような項目を高校で必修科目として学習すべきだということは，多くの人たちも共感するであろう．しかし，個々の具体的な項目をどのように教育するかについては，さまざまな見解があり得るし，科学的な理解を目指すとき，どこまで精密なモデルを提示できるかは高校現場に依存することになる．たとえば，「情報の信頼性」や「情報の信憑性」を科学的に捉えることは，容易ではない．また，コンピュータの不正アクセス禁止などの具体的な行為であっても，その意味を正確に把握できるモデルは単純ではない．実際，コンピュータ利用の歴史を遡ってみれば，利用者 ID やパスワードの登録が，当然だとみなされるようになったのは，最近のことである．いまでも，自宅や個人研究室内でのコンピュータ利用では，多くの人たちが利用者 ID やパスワードを登録せずに使っているという実状がある．一般的に，不正アクセスは，それが意図的かどうかに関わらず，ユーザが誤ったシステムの使い方をしてしまって気づかないうちに，ネットワーク環境に影響を与え，運用に支障が生じるということもある[19]．

### 4.1 利用者 ID と個人認証

近年のコンピュータ環境の進展やインターネット環境の整備により，情報教育はどのように変化してきたか，振り返ってみよう．

大学での情報教育は1960年代に工学系の一部の学科で，FORTRAN の文法教育から始まったといえる．そのころは，大型汎用コンピュータをバッチ処理で研究利用する形態が中心であり，コンパイラを使う実習もなく単に FORTRAN の記述法を解説するくらいであった．その後，1970年代には TSS 処理方式が普及し，BASIC インタプリタを使った対話型の即時処理が可能になった．しかし，パーソナルコンピュータが普及するまでは，コンピュータにアクセスできる人間の数は限られたものであり，情報セキュリティ教育の重要さを実感するにはいたらなかった．実際，指導教員が研究用に申請した ID を，研究室 ID と称して，研究室の学生や院生が共同して利用していた．個人の ID というよりも，コンピュータ使用料などの会計処理に必要な部類コード，という認識であった．したがって，パスワードは研究室の掲示板に大書されていたものである．

### 4.2 パスワード教育の困難さ

パスワードを考案するとき，「辞書にある単語は避ける」「8 文字以上にする」「数字や記号を混ぜる」など色々なアドバイスがあるが，これらを守れば本当に安全なのだろうか．とくに，ここでいう辞書が「国語辞典」や「英語の辞書」だけでなく「人名辞典」や「地名辞典」さらには「パスワードによく使われる文字列」なども，含まれている点に注意が必要である．たとえば，英字キーボードのキー配列の一部である「qazxdrt」とか「asdfghjk」などもパスワードには不適切な文字列だといえる．実際には「パスワードを推測しようとする悪意ある人に推

測されないもの」が，強いパスワードの条件だということになる．

パスワードの強度を確認するソフトとして「AntiCrack」は有名である[20]．自分で考案したパスワードの簡単な検査が手軽にできるので便利であるが，山口大学のサイトで運用されているシステム[21]でクラックされるのは，極めて稚拙なパスワードに限られており，このソフトでクラックされなくても，そのパスワードが決して安全だとはいえないことに注意する必要がある．

実際，パスワード・クラッキング・ツールと呼ばれるソフトウェアで最も有名なものが「John the Ripper」だといわれている[22]．だれでも簡単な操作でWeb検索でき，ダウンロードしてコンパイルすれば利用でき，ほとんどのUNIXで稼動するし，Windows版もある[23, 24]．このようなソフトウェアで悪意をもってクラックされると，クラッキングに使用するコンピュータが市販のPCであっても，8文字から10文字程度のパスワードでは，どのように工夫しても決して安全とはいえない．定期的なパスワードの変更が推奨される理由の一つである．

### 4.3 バイオメトリックスの応用

関西大学総合情報学部において実習用のPCへのアクセスコントロールとして，バイオメトリックスによる認証システムを導入したのは，2001年の秋学期からである[25]．このシステムではログインIDごとに,各利用者の指紋パターンの特徴を抽出したミニューシャ（minutia）を約300バイトの記号列で表現し，これを暗号化してパスワードとして利用しているのである．つまり，従来のシステムでは10バイト程度のパスワードが，約30倍の長さに拡張されたことになり，クラッキングへの耐性が格段に向上しているのである．その結果，パスワードの場合には3ヶ月程度の有効期限を設定しているが，バイオメトリックス利用者にはその学生が卒業するまで更新の必要が無いのである．

大学の実習教室へのアクセスコントロールとしては，名古屋音楽大学が2000年度から指紋認証システムを導入しており[26]，高知工科大学の電子・光システム工学科ではICカードと指紋認証システムを併用している[27]．しかし，これらの両大学における導入事例は部屋の鍵としての使い方に限られており，コンピュータシステムへのアクセスコントロールとしては，関西大学での導入事例が先進的といえる[28]．

また，米国の公立学校で普及している指紋認証システムの例としては，ペンシルバニア州で1998年から使われているケースが著名である[29]．そこでは，3000人以上の生徒がカフェテリアでの昼食の決済に指紋認証システムを利用している．また，全米では45学区で使われており，毎日25万人の生徒が学校で指紋をスキャンしている．

これからのビジネスの発展という観点からは，IEEEのSpectrumに掲載されたバイオメトリックスの記事が有用である[30, 31]．そこでは，バイオメトリックスのビジネス規模が，2003年度に7億1,900万ドルから2008年度までには6倍以上の46億ドルを超えると予想している．

そして，バイオメトリックス技術の中でも指紋認証技術が最も急速に普及し，指紋認証ビジネスだけでも2008年度には15億ドル規模に達すると予想している．

バイオメトリクスはオフライン機器へのアクセスコントロールが主な用途であったが，今後はネットワークアプリケーションとの連携が重要な課題である[32]．関西大学総合情報学部の実習用ワークステーションへのアクセスコントロールは，未だにパスワードに限定されている．ネットワーク対応の認証ソフトウェア開発がメーカで進展するのに期待している．

### 4.4 暗号化技術

情報セキュリティ技術としては，個人認証技術やアクセス制御技術・侵入検出技術・ファイヤウォール技術など，色々あるがその基盤となるのが暗号化技術である．パスワードファイルには暗号化されたパスワードを保存して，利用者の認証に用いている．したがって，このパスワードファイルを解読することが，クラッキングソフトウェアの目標になる．

一般的に，暗号の基本的機能は守秘と認証である[33]．守秘は不正な読取や複写から秘密を守る機能であり，認証は不正な改竄のないことを確認する機能である．

パスワードの保護だけでなく，利用者のデータを保存したファイルでも，不注意なアクセスからの保護を目的として暗号化技術が有用である．大学の研究室のメンバーだけで討論をするような場面でも，インターネットの掲示板などを活用すれば，メンバー間の時間的な調整が楽になり便利であるが，メンバー以外には公開したくないデータがあるとき，暗号化技術を応用することで問題が解決できる．

### 4.5 PGP 実習の導入

総合情報学部の1年次生に開講している「基本ソフトウェア実習Ⅰ，Ⅱ」では，2001年度からPGP実習を導入している[34]．PGP（Pretty Good Privacy）は世界的に広く普及している暗号化ツールの一つであり[35, 36]，大学の実習教材としては非常に有用である[37]．具体的なPGP実習課題としては，クラス規模（学生数が約50名で同時並行して3クラスの実習がある）に見合う実習課題を，サーバシステムの性能などを考慮しながら考案する必要がある．

本論文の付録に添付したスクリプトは著者の一人である小林が開発したものであり，多くのクラスで活用されている．このスクリプトでは，わずかに3つのファイルを準備するだけで自動的にクラス全員宛にPGP署名付メール1通とPGP署名無しメール9通をランダムな順にして，一斉に送信できる．

1．クラス学生のメールアドレス（SAやTAのメールアドレスも含められる）
2．送信用の本物メッセージ
3．送信用の偽物メッセージ

クラスの各学生は，個々にクラスの担当教員から届いたメールを確認することになるが，各自が受信した10通のメールの何番目がその教員のPGP署名付メールであるかは，一定してい

ない．つまり，PGP 署名が確認できたメッセージが本物であり，PGP 署名のないメッセージは偽せ物ということになる．実際に PGP 署名の有無を個々の学生が自分の端末で確認できるか否かが容易に把握できるので，この課題はディジタル署名の実習として極めて有効である．

## 5．おわりに

IT モラルの教育は非常に重要であり，学校教育に対する社会的な期待も大きい．しかし，日進月歩の電子情報機器の発展に合わせて，タイムリーな教材を用意し，効率的に学生たちに提供することは，決して容易ではない．関西大学総合情報学部では，学部の創設以来，各年度のシラバスを検討するたびに，基本ソフトウェア実習担当者たちは，その実習内容を再検討し，改善を積み重ねてきた．今後も，このような努力は継続することが望まれるが，そのような議論の概要を公開して，他の学部教員などから建設的な意見をもらう努力をするべきであろう．従来は，意図していなかったとはいえ，閉鎖的な運用だと見なされていたかもしれない．

## 謝辞

本研究の主要テーマは，関西大学総合情報学部の 1 年次生に対して開講されている授業科目である，「基本ソフトウェア実習 Ⅰ，Ⅱ」の担当者が専用のメーリングリストなどを通して，毎年のように議論を積み重ねてきた内容を含んでいる．ここに記して謝意を表したい．とくに，学部創設の準備段階からどのような実習が必要になるかについて，熱心な討議に参加して下さった，辻教授や宮下教授，上島教授には深く感謝している．また，関西大学 IT センターの職員の方々や高槻キャンパス・ネットワークセンターの職員の方々にも，技術的な面で色々とご教示頂き，深く感謝している．

なお，本研究の一部は，平成15年度関西大学学部共同利用研究費によって行った．

## 参考文献

[1] 文部省：高等学校学習指導要領解説 情報編，開隆堂出版（2000）．
[2] S. Baase（日本情報倫理協会訳）：IT 社会の法と倫理，ピアソンエデュケーション，(2002)．
[3] 情報処理学会：情報処理学会倫理綱領，http://www.ipsj.or.jp/gaiyo/ipsjcode.html（1996）．
[4] レイチェルズ（古牧・次田 共訳）：現実をみつめる道徳哲学，晃洋書房（2003）．
[5] 上園：情報セキュリティと情報倫理，宮地・菊池編，情報セキュリティ（2003）所収．
[6] 新村，広辞苑第 5 版，岩波書店（2003）．
[7] 金谷 訳注，大学・中庸，岩波文庫（2003）．
[8] 笠原：情報通信倫理，宮地・菊池編，情報セキュリティ（2003）所収．
[9] 松岡：情報の歴史，NTT 出版（1990）．
[10] 江澤：IT 革命と情報倫理，システム／制御／情報，Vol.45，No. 9，pp.523-527（2001）．
[11] 江澤：情報処理Ⅰ 講義レジュメ，関西大学出版会（2003）．

[12] ThinkQuest Japan：http://www.thinkquest.gr.jp/library/，http://www.thinkquest.org/library/，(2004).
[13] 永井・小泉：なぜ人を殺してはいけないのか？河出書房新社 (1998).
[14] 鷲田：死なないでいる理由，小学館 (2002).
[15] 邱永漢：お金の原則，光文社 (2001).
[16] 村田：今，必要な情報セキュリティマネジメント (2003).
[17] 舟木・峰岸：「知の共有」と情報セキュリティマネジメント，ダイヤモンド社 (2004).
[18] 水越・村井監修，情報A，情報B，情報C，日本文教出版 (2003).
[19] 山口，ブロードバンド時代のインターネットセキュリティ，岩波科学ライブラリー85 (2003).
[20] 冨永，抗クラック剤の開発，Programming Tools and Techniques 185 (1993).
[21] anticrack, http://web.cc.yamaguchi-u.ac.jp/crack.php3 (Feb. access 2004).
[22] Daiji Sanai，クラッキングに強いパスワード選択の極意，http://itpro.nikkeibp.co.jp/NBY/SecurityStadium/result2002/password.htm，(Mar. access, 2004).
[23] 井原，パスワードはこうやって盗まれる，日経ネットワークセキュリティ Vol. 2，日経 BP 社，pp. 22-33 (2002).
[24] 三島・小森谷・Urity，ブルートフォースに勝つ“強いパスワード”とは，日経ネットワークセキュリティ Vol. 2，日経 BP 社，pp.36-47 (2002).
[25] 江澤・小林・中芝：大学における情報モラル教育支援環境の課題，情報研究，第19号 (2003).
[26] 高橋，音大生のためのマルチメディア教育環境の構築をめざして－名古屋音楽大学－，大学教育と情報，Vol.10，No. 1 (2001).
[27] 矢野，指紋認証を“ドアコントロール”に先駆的活用，http://www.necsoft.com/itsval-way/pdf/03/03-edition-part02.pdf (2001).
[28] 江澤・中芝：セキュリティ対策と個人認証システム，教育の情報化フォーラム，私立大学情報教育協会，pp.74-77 (2002).
[29] C. Graziano，米国の公立学校で普及進む「指紋認証システム」，http://www.hotwowired.co.jp/news/news/culture/story/20030811201.html (2003).
[30] G. Weiss : Biometrics Boom, IEEE Spectrum, Mar, (2004).
[31] Latest Tests of Biometrics Systems Shows Wide Range of Abilities, IEEE Spectrum online, Web Only News, http://www.spectrum.ieee.org/WEBONLY/wonews/jan04/0104biom.html, (Mar. access, 2004).
[32] 日本バイオメトリクス認証協議会，バイオメトリクス認証の潮流，http://www.biometrics.gr.jp/JBAA/，(Mar. access, 2004).
[33] 今井・佐々木：情報セキュリティ対策の概要，宮地・菊池編，情報セキュリティ (2003) 所収.
[34] 関西大学総合情報学部：基本ソフトウェア実習テキスト2003年度版 (2003).
[35] S. Garfikel (山本監訳)：PGP 暗号メールと電子署名，オライリージャパン，(1996).
[36] IIJ 技術研究所：日本の公式 PGP ホームページ，http://pgp.iijlab.net/ (Feb. access 2004).
[37] 東京工業大学全学情報科目実施委員会編：コンピュータリテラシ，昭晃堂 (2001).

# 付録　PGP/MIME 仕様一括メール送信スクリプト

```
#!/bin/sh
#-----------------------------------------------------------------------
#  PGP/MIME 仕様に基づいた一括メール送信（PGP 署名自動添付）スクリプト
#
#  Copyright (C) 2000, Takashi KOBAYASHI
#  $Id: pgp-jisshu.sh,v 1.2 2001/12/12 01:48:09 kobayasi Exp $
#  本体 118 行目以降
#-----------------------------------------------------------------------

clear
echo ""
echo "-----------------------------------------------------------------------"
echo "                一括メール送信（PGP 署名自動添付）スクリプト                "
echo ""
echo "  このスクリプトは実習用ですので，１０分の１の確率で本物メールを        "
echo "  送信します．ご注意下さい．                                            "
echo ""
echo '  $Revision: 1.2 $'
echo "                Copyright (C) 2000, Takashi KOBAYASHI                  "
echo "-----------------------------------------------------------------------"
echo ""

#-----------------------------------------------------------------------
#  各種関数定義
#-----------------------------------------------------------------------

#
#  署名を作成するメッセージボディの作成
#  7bit 化＋行末<CR><LF>化
#
makebody () {
    sed 's/
$//g' ${REALARG} > ${SEDWORKFILE}
    (echo 'Content-type: Text/Plain; charset=iso-2022-jp'; echo
'Content-Transfer-Encoding: 7bit'; echo ''; cat ${SEDWORKFILE}) | ${NKFCOMMAND}
}

#
```

```
#  偽造したメッセージボディの作成
#  7bit 化＋行末<CR><LF>化
#
makefake () {
    sed 's/
$//g' ${FAKEARG} > ${SEDFAKEFILE}
    (echo 'Content-type: Text/Plain; charset=iso-2022-jp'; echo
'Content-Transfer-Encoding: 7bit'; echo ''; cat ${SEDFAKEFILE}) | ${NKFCOMMAND}
}

#
#  メッセージヘッダーの作成
#  マルチパートの boundary 設定
#
makeheader () {
    echo "Subject: <<SUBJECT>>"
    echo "X-Mailer: Auto-signature script V0.90"
    echo "Mime-Version: 1.0"
    echo 'Content-Type: Multipart/Signed; protocol="application/pgp-signature";'
    echo ' micalg="pgp-sha1";'
    echo " boundary=\"--Security_Multipart(${DATE})--\""
}

#
#  署名作成
#
signmessage () {
    pgps -q -ba +language=en +batchmode=off -o ${SIGNEDFILE} ${WORKFILE}
    if [ "$?" != "0" ]
    then
echo 電子署名の追加に失敗しました.
exit 1
    fi
}

#
#  メッセージボディの前の boundary 挿入
#
bodyseparator () {
    echo ""
    echo "----Security_Multipart(${DATE})--"
#    (echo "Content-type: Text/Plain; charset=iso-2022-jp"; \
#     echo "Content-Transfer-Endocing: 7bit"; \
```

```
#     echo "") | ${NKFCOMMAND}
}

#
#  署名の前の boundary 挿入
#
signseparator () {
    echo ""
    echo "----Security_Multipart(${DATE})--"
    echo "Content-Type: application/pgp-signature"
    echo "Content-Transfer-Encoding: 7bit"
    echo ""
}

#
#  最後の boundary 挿入
#  行末の "--" が多いことに注意
#
tailseparator () {
    echo ""
    echo "----Security_Multipart(${DATE})----"
}

#
#  シグナルトラップ関数
#  作業ファイルをすべて消去
#
cleanup () {
    echo ""
    echo "＊＊＊＊＊　中断します　＊＊＊＊＊"
    rm -f ${WORKFILE} ${SIGNEDFILE} ${FAKEFILE} ${MESSAGEFILE} ${FAKEMESFILE}
${SEDWORKFILE} ${SEDFAKEFILE}
    echo ""
    exit 1
}

#
#  ヘルプメッセージ
#  利用方法を表示
#
USAGE () {
    prog=`basename $0`
    echo ""
```

```
    echo "利用法:"
    echo ""
    echo "  ${prog} [-l members] [-s subject]"
    echo ""
    echo "    -l members:   受信者リストファイル名の指定"
    echo "    -s subject:   Subject: フィールドの指定"
    echo "                  (1バイト文字のみ)"
    echo ""
}

#-------------------------------------------------------------------------
#  PGP/MIME 仕様に基づいた自動署名添付スクリプト
#
#
#-------------------------------------------------------------------------
umask 0077
MESSAGEFILE=/tmp/pgp-mess.$$
FAKEMESFILE=/tmp/pgp-fmes.$$
SIGNEDFILE=/tmp/pgp-sign.$$
WORKFILE=/tmp/pgp-work.$$
FAKEFILE=/tmp/pgp-fake.$$
SEDWORKFILE=/tmp/pgp-sedwork.$$
SEDFAKEFILE=/tmp/pgp-sedfake.$$
RARG=
FARG=
SARG=
LARG=
#
NKFCOMMAND="/usr/local/bin/nkf -j -c"
DATE=`date +%Y%m%d%H%M%S`
#
trap cleanup 1 2 9 15
#
MES=`echo -n echo`
case ${MES} in
-n*) preecho="";   postecho="\c" ;;
*) preecho="-n"; postecho=""   ;;
esac
#
set -- `getopt f:s:l:h $*`
if [ $? != 0 ]
then
    USAGE
```

```
    exit 2
fi
for i in $*
do
    case $i in
    -s)  SARG=$2; shift 2;;
    -l)  LARG=$2; shift 2;;
    -h)  USAGE; exit 1;;
    --)  shift; break;;
    esac
done

nofile=1
while [ ${nofile} = 1 ];
do
    echo ${preecho} "本物のファイル名を入力して下さい: ${postecho}"
    read RARG
    if [ ! -f ${RARG} ]; then
echo "ファイル ${RARG} が存在しません. "
nofile=1
    elif [ -d ${RARG} ]; then
echo "ディレクトリ ${RARG} が存在します. "
nofile=1
    else
nofile=0
    fi
done
REALARG=${RARG}

nofile=1
while [ ${nofile} = 1 ];
do
    echo ${preecho} "偽物のファイル名を入力して下さい: ${postecho}"
    read FARG
    if [ ! -f ${FARG} ]; then
echo "ファイル ${FARG} が存在しません. "
nofile=1
    elif [ -d ${FARG} ]; then
echo "ディレクトリ ${FARG} が存在します. "
nofile=1
    else
nofile=0
    fi
```

```
done
FAKEARG=${FARG}

if [ x"${SARG}" = x"" ]; then
    echo ${preecho} "Subject: が空ですがよろしいですか？ (y or n) ${postecho}"
    read ans
    case ${ans} in
    [Yy][Ee][Ss])  nosubj=0;;
    [Yy])          nosubj=0;;
    [Nn][Oo])      nosubj=1;;
    [Nn])          nosubj=1;;
    *)             nosubj=1;;
    esac
    while [ ${nosubj} = 1 ];
    do
echo ${preecho} "Subject: に指定する文字列を入力してください: ${postecho}"
read SARG
if [ x"${SARG}" != x"" ]; then
    nosubj=0
fi
    done
fi

if [ x"${LARG}" = x"" ]; then
    nolist=1
    while [ ${nolist} = 1 ];
    do
echo ${preecho} "メンバーリストファイルを入力してください: ${postecho}"
read LARG
if [ ! -f ${LARG} ]; then
    echo "ファイル $LARG が存在しません. "
    nolist=1
elif [ -d ${LARG} ]; then
    echo "ディレクトリ $LARG が存在します. "
    nolist=1
else
    nolist=0
fi
    done
fi

#
makebody > ${WORKFILE}
```

```
makefake > ${FAKEFILE}
signmessage
#
#
# ヘッダの定義
# 入力された Subject: に置き換えます.
makeheader | sed 's/<<SUBJECT>>/'"${SARG}"'/' > ${MESSAGEFILE}
makeheader | sed 's/<<SUBJECT>>/'"${SARG}"'/' > ${FAKEMESFILE}

bodyseparator >> ${MESSAGEFILE}
bodyseparator >> ${FAKEMESFILE}

#
#
# 本文を追加
cat ${WORKFILE} >> ${MESSAGEFILE}
cat ${FAKEFILE} >> ${FAKEMESFILE}

#
#
# 署名の前に挿入する
signseparator >> ${MESSAGEFILE}
signseparator >> ${FAKEMESFILE}

#
#
# 電子署名を追加する
cat ${SIGNEDFILE} >> ${MESSAGEFILE}
cat ${SIGNEDFILE} >> ${FAKEMESFILE}

#
#
# メッセージの最後に挿入されるフッター
tailseparator >> ${MESSAGEFILE}
tailseparator >> ${FAKEMESFILE}

#
# メールを送信 (sendmail を直接起動)
echo ""
echo "指定された本文ファイルを指定のリスト宛てに送信します. "
echo "  メンバーリスト: ${LARG} (" `egrep -v '(^$|^#)' ${LARG} | wc -l |awk '{print
$1}'` "名 )"
echo ""
```

```
count=0
number=0
for receipt in `egrep -v '(^$|^#)' ${LARG}`;
do
    notfake=`/usr/local/bin/perl -e 'print int(rand(10));'`
    while [ "`expr ${number} \< 10`" = "1" ];
    do
if [ "${number}" = "${notfake}" ]
        then
    cat ${MESSAGEFILE} | /usr/lib/sendmail ${receipt}
else
    cat ${FAKEMESFILE} | /usr/lib/sendmail ${receipt}
fi
number=`expr ${number} + 1`
    done
    count=`expr ${count} + 1`
    number=0
done
echo ${count}名に送信しました.
rm -f ${WORKFILE} ${SIGNEDFILE} ${FAKEFILE} ${MESSAGEFILE} ${FAKEMESFILE}
${SEDWORKFILE} ${SEDFAKEFILE}
```