

Software Use in the Physics Department

The Physics department at HMC uses computers in many ways in its laboratories, lectures, and courses. In the laboratories students use software to simplify and speed up data analysis and plotting in lab books and technical reports. Students in the lower division laboratories mostly use packages such as KaleidaGraph while in the sophomore and upper division laboratory work, we are standardizing on Origin, which has superior nonlinear fitting routines that most other packages lack, but are essential to our data analysis. Freshman courses are not yet making much use of software, except for the occasional use of Interactive Physics II, Gravitation Ltd., and images captured from the Internet for lecture demonstrations. Students are also expected to be able to program simple problem solutions in C or Pascal for coursework.

Greg Lyzenga's use of software in teaching sophomore and upper division classes breaks down into two categories: First, he uses software as a demonstration or illustration tool. For this he makes extensive use of World Wide Web pages to share text, graphics, and QuickTime animations with the classes. To generate these he uses general purpose Macintosh software such as ClarisDraw or KaleidaGraph, and more specialized applications such as Theorist (by Waterloo Maple) and Spyglass for plotting and visualization. You can see examples of these uses by visiting the Physics 51 Home Page at <http://www.physics.hmc.edu/courses/Ph51.html>. Included are GIF-format images of solutions to various problems in electromagnetism, as well as movies, source code listings and lecture figures that can be viewed and printed by a browser application like Netscape. The page also includes links to the syllabus and homework assignments, as well as homework answer keys (maintained as an "electronic reserve" by the Sprague Library staff). This represents a first experiment with extensive use of electronic distribution of course materials in Physics 51, and so far the response to this development has been very positive. In preparing these and other course demonstration materials, he also relies to a considerable extent on custom programs that he writes on the departmental UNIX workstation in C. The second way he uses software is as a direct calculation tool for the students. The most common approach is to allow students latitude in choosing the software tools they want to use to solve a given problem. For example, Mathematica is often used to solve problems that arise in his classes, and fitting/plotting programs such as Origin or KaleidaGraph are very popular in the lab courses.

The seismology lab also uses a combination of commercial Macintosh software and homegrown applications written with CodeWarrior C or Think C. The six recording seismometers in the lab are connected to a National Instruments analog-to-digital data acquisition board in a dedicated Macintosh IIcx computer. *(continued on page 6)*

DATA ENCRYPTION

IN THIS ISSUE

Software Use
in the Physics
Department --- cover

Data Encryption -- 2

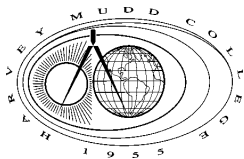
Editor's Notes ----- 3

Getting Help
from Academic
Computing ----- 4

Tricks & Tips ----- 7

Q&A ----- 8

Occasional Downtime is composed on a Apple Power Macintosh 6100/66 using Aldus PageMaker 5.0 and Microsoft Excel 4.0. The primary typefaces used are Times and Optima. We wish to thank Sally Rich Arroyo of the HMC Office of College Relations for all her help.



In the last issue we discussed simple ways to protect your computer files from prying eyes. For many needs, however, these simple protections may be insufficient. In this article we will discuss how to take things to the next level by encrypting your files so that even if someone were to gain access to them they would not be able to read them. This article will be in two parts. In this issue we will discuss the basic techniques that underlie most popular encryption tools. In the next we will describe some of the tools available and what features they offer.

There are a variety of reasons you might be interested in encrypting some of your files. For instance, while it is generally easy to keep intruders from gaining unapproved access to a computer over the net, you may be concerned that someone could gain physical access to your machine, thus bypassing any protections you apply to your network connection. Even if you use a password protection mechanism to restrict access to your machine, some of these utilities are very simplistic, and can be bypassed just by booting off a floppy. Even the better utilities can often be bypassed given enough time. Another reason to encrypt a sensitive file is if you need to send it over the network using e-mail or FTP. You may not have thought about it before, but when you communicate over the Internet, unless you take other precautions, everything you write is sent as plaintext and is extremely easy for those with the right tools to capture and reconstruct.

The solution to all of these problems is to encrypt your sensitive files and communications so that only you (or, in the case

of e-mail, the intended recipient) can read them. Encryption is the process of taking a source document (called plaintext) and converting it to an unreadable form (called ciphertext) that can only be understood if you know how to decode it. The art of encryption goes back thousands of years, and there are probably hundreds of different techniques, but all of them belong to one of three categories:

Rule Based Encryption. In this, the most basic type of encryption, some fixed rule is used to encrypt all documents. This rule might be as simple as "For each word, remove the leading consonants (if any), move them to the end of the word, and append the letters 'ay'" or as complex as "Take the pattern of bits making up the entire document, invert each bit (changing 1's to 0's and 0's to 1's), and reverse the order of the eight bits that make up each letter." The problem with rule-based encryption is that if the rules used to form the ciphertext are compromised, all encrypted documents are immediately compromised as well.

Secret-Key Encryption. For at least two thousand years, this has been the most common form of encryption. In secret-key encryption the rules used to form the ciphertext include reference to some "key" that can be changed for each document. In the days before computers the key was often some sort of lookup table that told the coder how to convert the document on a letter-by-letter or word-by-word basis. Due to the difficulty of coming up with new lookup tables, another common scheme is to pick some number (rather than a lookup table) as the key and use that in the rules. For example, the rule might be

“replace each word with the Kth following word in the current edition of the Oxford English Dictionary,” where K is the chosen key. Computerized implementations of secret-key encryption typically involve some bit-wise manipulation using the key: for example, multiplying the numeric representation of each letter or word by the numeric representation of the key.

While secret-key encryption is a good solution (and at present the preferred solution) for encrypting files that you wish to keep private, it has always had a serious flaw when used for communications: There must be some reliable way to communicate the secret key to the intended recipient. If that communication is compromised, then any message encoded with the intercepted key is compromised. Therefore, successful secret-key communication generally requires an initial face-to-face meeting or the use of a trusted intermediary to transmit the secret key.

The best known computer implementation of secret-key encryption is the Data Encryption Standard (DES) which was developed at IBM and established as a standard by the US government in 1977. Many implementations of DES are available, and almost all commercial programs for encrypting files on your desktop computer’s hard drive use DES.

Public-Key Encryption. In public-key encryption systems the key used to encrypt a message is different from the key used to decipher it. If you wish people to be able to send you encrypted messages, you send them your public key, which they can then use to encrypt messages intended for you. Even if the public key were intercepted, it would not matter, since that key cannot be used to decrypt messages. Rather, your private key, which stays strictly in your possession, is used for decryption.

Though it is founded on theorems first proved by Euler in the 18th century, public-key

(continued on page 5)

Editor's Notes

We are happy to have two contributions by faculty members in this issue of *Occasional Downtime*. The cover article is the first in what we hope will be a continuing series of articles on the use of computers and specialized software in the various departments at Harvey Mudd College. In this article the Physics Department discusses some of the ways in which they use computers in their laboratories, lectures, and courses.

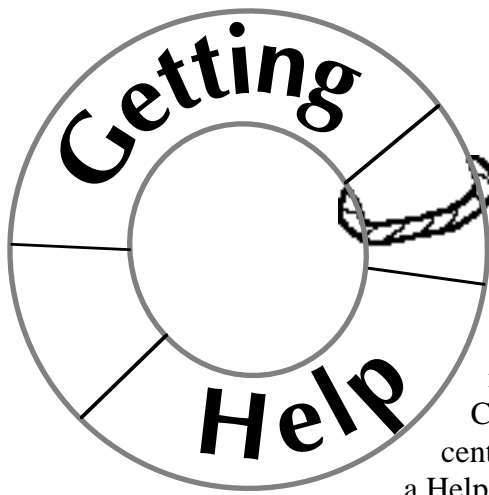
The second article is by Joshua Hodas from the Computer Science Department, and is the concluding piece in our series on data protection. In the first of a two-part article he introduces some of the concepts behind data encryption. In the second part, which will appear in the February issue, he will talk about some of the encryption tools available today.

The final article in this issue is a brief review of ways to get help from Academic Computing, with an emphasis on how to use our new Help Desk.

We hope you enjoy this issue of *Occasional Downtime*. If you have an idea for an article you would like to contribute, please let us know by sending e-mail to downtime@hmc.edu.

—Elizabeth Hodas

Occasional Downtime is published bimonthly by the Academic Computing Department at Harvey Mudd College. It is also available in plain text format on the HMC Gopher Server and in a variety of formats on the HMC Web Server. Comments and questions can be directed to downtime@hmc.edu.



From Academic Computing

Getting help when you have a computer problem should not feel like a big problem in itself. For that reason Academic Computing has introduced a centralized Help Desk built around a Help Desk phone number (7-7777) and a FileMaker Pro database to track reported problems. The main goals of the Help Desk are:

- To track ongoing problems, so that they are handled in a timely fashion.
- To simplify the hand-off of problems from the user to Academic Computing.
- To coordinate internally the handling of multifaceted problems.
- Ultimately, to provide means for users to report and check the status of their problems via the Web.

The main advantage of calling the Help Desk is that you don't need to know which specific AC staff member to contact. Anytime you don't know who can solve your problem or answer your question, call the Help Desk. The second advantage of the Help Desk is that if your problem/question can't be solved or answered right away, it will be logged in the database and tracked. Examples of situations that can be handled by the Help Desk are:

- You have a "How do I ...?" question.
- You have a software or hardware problem on your desktop computer.
- There is a problem with an AC resource (i.e. the network or one of the computers maintained by AC).
- There is a problem with a departmental computer lab facility and the department can't handle or fix the problem.

For problems with a computer in one of AC's microcomputer labs you should first try to talk to a consultant or send

e-mail to one of the system lists, but you should call the Help Desk if neither of these options is helpful or if the problem is urgent (i.e. Kato or the network is down).

When a problem is logged with the Help Desk the person assigned to the problem will contact you (usually by e-mail) to tell you that he/she has been assigned to your problem and to give you the log number of the problem and an expected completion date. When the problem has been solved, he/she will get in touch with you to tell you any necessary details of the solution. To get an update on the status of a reported problem you can call or send e-mail to the Help Desk (help-desk@hmc.edu) at any time. Be sure to include the log number of your problem if you have it.

Academic Computing has several other resources for getting help. The AC Documentation Library is being updated and moved from Gopher to the AC Web server. It can be found at <http://www.hmc.edu/comp/doc/>. While it is not yet complete, it already includes a great deal of documentation on topics such as e-mail, the World Wide Web, dorm networking, and other subjects. It also includes a FAQ (Frequently Asked Questions). In addition, this semester AC introduced some printed documentation on Pine, emacs, HTML, and AC computer resources. You can find them outside our main office as well as in electronic form in the AC Documentation Library in Acrobat (PDF) format.

Many faculty and staff have been taking advantage of our series of workshops on the Web, Eudora and other subjects. If you are not able to come to the workshops, all of the handouts from AC's workshops are available on the Web in Acrobat format at <http://www.hmc.edu/comp/workshop/>. (continued on page 7)

encryption was invented by Whitfield Diffie and Martin Hellman in 1976. While there are now several different public-key schemes, they are all based on the use of what are referred to as “trap-door” problems. A trap-door problem is one which is ordinarily hard, but which becomes easy with a little additional information. For example, suppose you have a 100-digit number that you know is the product of two large prime numbers. Finding those two primes would ordinarily require performing billions of divisions. But if you knew one of the prime factors, finding the other would require only a single division.

In public-key encryption one number (like the 100-digit number above) is used as the public key. The encryption algorithm is designed so that decryption is a trap-door problem. While the public key could technically be used to decrypt messages as well, it would require solving the trap-door problem the hard way. In contrast, the private key, which is used to decrypt the message, provides the extra information needed to solve the trap-door problem the easy way.

The downside of public-key encryption schemes is that, due to the nature of the mathematics involved, they are much more computationally intensive than typical secret-key schemes. Therefore, in most public-key schemes, the bulk of the encryption is done using secret-key techniques with a randomly generated secret key. That key is then encrypted using the public-key scheme and sent along with the encrypted document. The recipient then uses her private key to decrypt the secret key, which she in turn uses to decrypt the actual message. Since the secret key is itself encrypted with the public key, it is as though the entire message had been encrypted that way.

The algorithm now most commonly used for public-key encryption is called the RSA algorithm, named for its inventors, Rivest, Shamir, and Adleman. This

algorithm is patented and implementations that use it must get a license from RSA Inc.

An issue closely related to encryption is authentication: how do you know if a message was sent by whom it claims to have been sent by. As it turns out, independent of its use for encryption, RSA (and many other public-key schemes) can be used for authentication as well. The sender of a message uses a variation of the RSA algorithm, together with his private key, to generate a *digital signature* which is appended to the message. The recipient then processes the message with the sender’s public key. The algorithm can verify both that the message must have come from the claimed author, and further, that the message has not been edited in any way since it was signed.

Finally, it is important to understand that how hard it is to break a cipher or to forge a digital signature is directly related to the length of the key used, but so is the amount of time and hardware needed to process documents. Therefore, there is an inevitable tradeoff between security and expense. The standard rule of thumb is that you should use a scheme strong enough so that the expected cost to crack the system exceeds the value the document would have to the cracker. 🐾

by Joshua Hodas

WWW RESOURCES ON CRYPTOGRAPHY

- ▼ Steve Levy’s excellent non-technical article about encryption and the “crypto rebel” movement for *Wired Magazine*:
<http://www.hotwired.com/wired/1.2/features/crypto.rebels.html>
- ▼ A French graduate student cracked Netscape’s “secure” communications scheme. His story:
<http://pauillac.inria.fr/~doligez/ssl/>
- ▼ RSA Inc.’s cryptography FAQ:
<http://www.rsa.com/rsalabs/faq/>
- ▼ Quadralay Systems’ page with links to just about every cryptography page on the web:
<http://www.quadralay.com/www/Crypt/>

A custom recording program developed by students and faculty over the last few years (most recently by HMC junior Chris Erickson) acquires the data and saves it to disk. A series of automated UNIX scripts (prepared by senior Mike Heasley) perform a variety of tasks with the recorded data. The data are transferred via FTP from the Macintosh to permanent storage. They are also plotted in quasi-real time and made available for public viewing on the Web. You can see the past five days of data (as well as other earthquake information of interest) by visiting the seismo lab web page at <http://www.physics.hmc.edu/research/geo/seismo.html>.

There are other software tools that are used in seismology and geophysics, primarily on the departmental UNIX workstation. In addition to the use of custom-written C programs for digital signal processing of seismic data, a major task is the reduction of high-precision GPS (Global Positioning System) data for the measurement of crustal movement and deformation. HMC has a site license for the use of JPL's GPS analysis software package, GIPSY-OASIS, which allows measurement of baselines spanning southern California with a precision of a few millimeters. Lyzenga and his research students have been busy during the past two years learning to use this software and scientifically interpret its products.

A number of upper division courses use computers heavily. For example, the Computational Physics course (Physics 170) taught by Robert Wolf now demonstrates the use of computational techniques, such as numerical integration, ODE solution, and Fourier spectrum analysis, using Mathcad, Maple, and MATLAB. This course also presents the use of simulation techniques, such as Monte Carlo, simulated annealing, and percolation theory, using student written programs in C, FORTRAN, or Pascal.

In Astronomy courses taught by Professor Alex Rudolph the main piece of commercial software used is called Interactive Data Language (IDL). IDL is a high-level, graphically oriented computer language that will run on all the major platforms at HMC (workstations, Macintoshes, and Windows PC's). It is currently installed on a Sun Sparcstation20 workstation. Because IDL is high level and works on virtually all platforms, any code written in IDL is completely portable, even across platforms (e.g., workstation to PC). The strength of IDL for Astronomical purposes (as well as many other scientific applications) is its ability to handle large data arrays (such as Astronomical images or spectra) simply and easily.

In Astronomy 101, the Observational Astronomy course taught jointly between Pomona and HMC, we have been using a program called Voyager to display the night sky as an aid to selecting objects for study with the telescopes at Table Mountain Observatory (TMO), as well as IDL for the reduction of the Astronomical images and spectra taken with instruments mounted on these telescopes. Students in Astro 101 make extensive use of this and other telescopes at TMO and at Brackett Observatory located on Pomona's campus. In addition to using IDL to analyze data from TMO, the students will also be using IDL to develop an understanding of Fourier Transforms (FTs) as part of their study of Radio Astronomy, in particular the synthesis of extremely high-resolution images using the technique of Interferometry. IDL contains a built-in Fast Fourier Transform (FFT) algorithm that allows students to gain practical experience of the properties of FTs, to augment their formal introduction to the topic in lecture. This "hands-on" experience with what otherwise may seem to be an abstract mathematical construct is the kind of interactive learning that we are striving to provide in the Astronomy program at HMC. ☞

*by Greg Lyzenga, Alex Rudolph
and Bob Wolf*

Tricks & Tips

& Tricks

AC REFERENCE LIBRARY

- ▼ D. Browne, *Word 6 for Macintosh*, Peachpit Press, 1994.
- ▼ R. Cowart, *Mastering Windows 95-The Windows 95 Bible*, SYBEX, 1995.
- ▼ J. December & N. Randall, *The World Wide Web Unleashed*, Sams Publishing, 1994.
- ▼ A. Engst, *Internet Starter Kit for Macintosh*, Hayden Books, 1994.
- ▼ A. Engst, C. Low & M. Simon, *Internet Starter Kit for Windows*, Hayden Books, 1995.
- ▼ D. Gookin & A. Rathbone, *PCs for Dummies, 3rd Ed.*, IDG Books, 1995.
- ▼ I. Graham, *The HTML Source Book*, Wiley & Sons, 1995.
- ▼ A. Greif, *FileMaker Pro for Macintosh*, Peachpit Press, 1994.
- ▼ C. Kenny et al., *Using Microsoft Office 4.2 for Macintosh*, Que, 1994.
- ▼ G. Kidder & S. Harris, *HTML Publishing with Internet Assistant: Your Guide to Using Microsoft's HTML Add-on*, Ventana Press, 1995.
- ▼ O. Kvern, S. Roth & B. Fraser, *Real World PageMaker 5.0, Macintosh ed.*, Random House, 1993.
- ▼ J. Levine & C. Baroudi, *The Internet for Dummies*, IDG Books, 1994.
- ▼ G. Todino, J. Strang & J. Peek, *Learning the UNIX Operating System*, O'Reilly & Assoc., 1993.
- ▼ E. Weinmann & P. Lourekas, *Photoshop 3 for Macintosh*, Peachpit Press, 1995.

Getting Help continued from page 4

We are also developing a library of reference books and teach-yourself books on popular software packages and topics. These books may be borrowed from Academic Computing. The box above contains a list of books currently available. We will periodically announce new acquisitions in subsequent issues of *Occasional Downtime*. If you are interested in borrowing a book, please see Elizabeth Hodas. 🐾

EUODORA AND NETSCAPE AT HOME

Eudora and Netscape are probably the two most common Macintosh and Windows programs that people will want to use when dialing in to the HMC campus network using SLIP/PPP. Since most users of Netscape and Eudora rely on their own sets of personal bookmarks and nicknames, you may want to know how to use these shortcuts at home without having to retype them.

Your Eudora nicknames are stored as a separate file along with the rest of your Eudora mailbox and folder files. On a Macintosh use the Finder to open the System folder. Open the Eudora folder and look for a file called Eudora Nicknames. On a PC use the File Manager in Windows to open your Eudora directory. There should be three files, called `nndbase.toc`, `rcpdbname.txt` and `nndbase.txt`, all of which you'll need. Copy the appropriate file(s) to a floppy disk and then copy them to the same location on your home computer.

Transferring your Netscape bookmarks is even easier since Netscape has a built-in feature for just this purpose. Open the Netscape application on your office machine and select View Bookmarks from the Bookmarks menu. Press the Edit>> button (More Options on the Macintosh) to access the full set of options. Click the Export button to bring up a dialog box for saving the active bookmarks file. Save the file to a floppy disk. On your home computer open a SLIP/PPP connection to the HMC campus network and then open Netscape. Select View Bookmarks from the Bookmarks menu and click the Import button. Choose the saved bookmarks file from the dialog box, then press OK (Open on the Macintosh) to insert the bookmarks after the last one in the list. 🐾

QUESTIONS *and* ANSWERS

Q: I was using Netscape with my modem and got “Unable to locate host” when clicking on a large number of bookmarks. The next day, I was able to access them fine. I suppose the system was overloaded, but why don’t I get a “timed out” error message instead of an unconditional “Unable to locate host”? When do I take such an error message seriously, and when should I just wait and try again? Is hoping for reliability futile?

contributed by Melvin Henriksen

A: There are many different reasons why you might get the “Unable to locate host” error when trying to access a site.

- 1.) The server is too busy to accept any new requests.
- 2.) The server is down.
- 3.) The network is congested or down at some point between your machine and the server you are trying to access.

Netscape often can’t tell what the specific problem is. All it knows is that it can’t connect to the server you want. The best strategy in these cases is to try again after 5 minutes or so. If that doesn’t work try again at a later or earlier time in the day (as you did). If you can’t get to a large number of sites then this might indicate a problem with the network (not necessarily our local network, but somewhere on the Internet). In that case what I usually do is try accessing one or more sites that I know are very reliable. If I can’t get to them then that’s a pretty good indication that there may be a problem on the Internet itself.

Unfortunately the reply to your question “Is hoping for reliability futile?” is, at present, yes. The Internet was originally designed as a research experiment and was certainly never intended for all of the many commercial, research, educational, and personal purposes for which it is being used now. While many smart people are working on new ideas and technology for improving the Internet, it will likely be several years before it is as reliable as the phone system. All things considered it works amazingly well.

Q: I can’t login to Kato. When I try I get the error message: “UserID not valid.”

A: Students are allowed only one Kato login at a time. If you are logged in to Kato from another computer you won’t be able to login again. Go to the last computer you were on and logoff. If you can’t remember which computer you were on come to the AC main office and ask Cindy to add you to the list of Kato sessions to be “killed.”

If you use Windows 95 on your dorm computer then you must first logoff from Kato before choosing Shut Down from the Start button menu. Shut Down only exits your Windows session; it does not log you off from Kato. If you do mistakenly Shut Down without logging off from Kato then you must come to the AC main office and ask Cindy to add you to the “kill” list. Be prepared to show your ID. 🐾