PWG Proposed MIF & White Paper Securing, monitoring and managing shared printers in the new economy.

Version 1.0 22.11.95 (PAYBACK.DOC)
Presented at Omni Hotel, Austin, Texas December 6-7th 1995.
Author(s): Geoff Slater, Peter Gutmann & Russell Coker.

Abstract.

There seems broad agreement that we need job monitoring and accounting. Several vendors have adopted these features over the years.

This paper goes back a few steps and asks why we need job accounting and monitoring in the first place and looks at the applications for accounting and monitoring. The conclusion is reached that management security is part and parcel of accounting and monitoring; that attempting to treat management as a separate item is fundamentally flawed.

A Dog's Breakfast.

Page accounting and job monitoring are not new. Various vendors have implemented features differently at various times for a dizzying number of usually unarticulated perceptions and assumptions made about the needs of their end-users. These perceptions and assumptions, even where they are documented and articulated, are rarely tackled in any disciplined or consistent manner.

If this were otherwise there would be no need for the PWG !

Every now and then a new printer security product becomes available. For example, Axis Inc make JetDirect equivalent cards which include limited security. Hewlett Packard has also introduced limited security with the latest JetDirect firmware revision to prevent anonymous telnet; and Apple is currently doing some interesting work along the lines discussed in this paper.

Security is not new. The real question is why security has had such patchy commitment from the various vendors. Tom Hastings has observed that agreement upon printer management is *controversial*, though he does not articulate why this is so. Unfortunately, we are not privy to the politics behind the controversy, but we are privy to the effects this has downstream on users and the problems of developing drivers and accounting applications in the flotsam.

To date there have been vague rumours of forthcoming security and tantalising promises that manufacturers are slowly figuring out that something is missing. Andy Reinhardt in his article for *Byte* magazine "Smarter Copiers, Printers and Fax Devices Are Coming", November 1994, at p.82 hints that the Microsoft "At Work" standard offers RSA security (public key encryption and authentication) as well status feedback such as paper jams and job completion.

The same tantalising theme of intelligent printers was in the February 1995 edition of *Byte* where it was written that:-

> "Far more useful is the software that reports what's happening with your print job, especially on a network, where it is nice to know if the printer down the hall is out of paper *before* you try to fetch your printout".

Despite these encouraging noises, printer security and intelligent monitoring still appears to be as real as the chimera in the Sahara desert.

Whither the customer.

We would be remiss if we did not analyse the reasons behind this historically *ad hoc* approach which has culminated in this paper . It is hoped that all of this will come to assume a high position on the PWG's agenda.

Thanks is offered to Ray Lutz of Cognisys Inc, Paul Reilly of Apple Computers, Harry Lewis from IBM and Forrest Tanaka at Hewlett Packard for providing the some of the primary data that has enabled the following analysis. There are others who wish to remain nameless who have allowed us access to some very interesting internal memos which form the evidence for what we are about to say.

One of the greatest problems in the IT industry is the communication gulf between users and the engineers who design printers and their features. Corporations have become wedding cake bureaucracies where vital customer information and feedback is constantly being interpreted, massaged and often just plain ignored by various information gatekeepers and information hoarders employed by the companies.

Various players in the corporate information chain have vested interests in passing along some information and holding back other bits of information. All sorts of exotic and often irrational forces and politics come into play. Jealousy of somebody with better ideas may cause supression of a good idea, as might information that contradicts somebody's pet project. We need not go into details. Most of us here can relate to corporate information politics.

As if this were not enough of a problem, there seems to be a common view that engineers need to be chained up in a lab and under no circumstances let loose on end-users or talk to resellers and/or developers.

Perhaps what many of you have secretly suspected is true. Maybe the current situation really is a management and marketing department conspiracy. Perhaps the rules in Parkinson's Law are true:-

> "Work expands so as to fill the time available for its completion. Granted that work (and especially paper-work) is thus elastic in its demands on time, it is manifest that there need be little or no relationship between the work to be done and the size of the staff to which it may be assigned. A lack of real activity does not, of necessity, result in leisure. A lack of occupation is not necessarily revealed by a manifest idleness. The thing to be done swells in importance and complexity in a direct ratio with the time to be spent [and inversely proportional to useful outcomes].
>
> - *Nicholas Parkinson, 1958.*

The engineer isolation factor is possibly because if engineers had direct user contact, why would companies need marketing departments to do "research" ? Taking it another step - well, lets not go too far. Management is needed. Probably.

As humorous as all this sounds, there is some truth in it. Putting aside employees' self-preservation, the more likely reason is that the marketing department is already isolated from customers itself. Let's take a recent real case study where the company shall remain anonymous. Security as a feature is still being deferred because of a perception by marketing that other "features" or perhaps gimmicks, are needed. Programming documentation has similarly been deferred in pursuit of more and more dubious features. Code maintainence is taking longer and longer and yet despite all of the features, the market share and sales keep slipping. The engineers are intensely frustrated. So are the dissatisfied customers - or what is left of them.

Now let's look at Mary the Marketer. Mary does not understand the fuss about security and job accounting. After all, she has a network printer at work, and a beautiful 18 page-per-minute model at that. If she is studying part-time and needs to print a paper - no problem. If her friend wants to print that thesis - no problem. If she wants to run off some flyers - no problem. Nobody at the office minds. After all, Mary does work for a printer manufacturing company. Why would it be a problem ?

In her case it is not a problem.

The downside is that the marketing department makes the mistake of looking at the world through rose-tinted, or perhaps we should say toner smudged, glasses. Mary assumes everyone has a nice employer and a network printer, and that the marginal cost of providing this is so little that it is not worth worrying about. This attitude permeates the entire decision making process. Unquestioned assumptions are a central reason why printer security has been neglected so long.

Many employers do not mind extra prints being run off, but they do want the *option if they choose* of knowing how many "extras" get run off every month, as well as deciding who gets printing privileges and who does not get privileges. How many and who means accounting. And accounting without security is no accounting at all. With estimates that up to 50% of printers sold will soon be connected on networks, accounting - and security and management for accounting - is something that can no longer be ignored.

Unfortunately, Mary the marketer's perspective discussed above does not hold true for the rest of the world.

Many larger corporations such as aircraft makers, law firms and accountants have long had billing processes that track the exact costs associated with individual projects. Laser printing is a often significant part of their cost base, as are photocopies and phone calls. They need tight accounting controls. Tight implies a high level of security. Even a small "leakage" of a few percent can translate into an alarming amount of consumables.

The Missing Link.

We suggest that printer companies create, as an urgent priority, a technical suggestion box on the internet and make this known to their user base. This should not be a day-to-day operational or technical support box. It should focus on things like why a particular plastic knob takes the skin off the user's knuckles when he or she changes the toner cartridge; how about the option of a buzzer to sound when out of paper ? Or a clear plastic ink cartridge so the user can see how much ink is left ?

Such a suggestion box would close the product time-cycle between growing user expectations for needed product features. It would replace the existing marketing department guesswork shuffle with some direct input . The upside would be useful features and a clear trail as to where marketing efforts should be directed. So if lawyers suggest cryptographic capable printers, target the law firms. If network administrators at a university want secure page accounting - there is the marketing clue.

Smaller companies do this intuitively. Larger companies forget the basic need to define themselves in terms of what customers actually want, as opposed to what the larger companies think (usually wrongly) customers want. The marketing department will immediately counter this, saying they talk to account representatives from larger companies who buy their products.

But that is the very problem. The information is probably already filtering out the support staff and end-users who have to deal with the nitty-gritty details. And in the end, because printers are technical items that are heavily interdependent on the network they are connected to, it is the nitty-gritty that really counts. Every layer between the person who finds nitty-gritty problems and comes up with fix and the engineers who know what can be done about it is an opportunity lost. Furthermore, the do not listen to smaller organisations at all.

We are deliberately NOT distinguishing between users and developers, because the the two are not - and never have been - neatly separated.

Without boring the reader with clichés about the new re-engineered or virtual corporation and so on, the old distinctions between supplier, maker, re-seller and customers have, in reality, become blurred. Take the simple act of e-mail at your workplace. This fundamentally changes the possible sources and flow of information in the corporation, allowing instant outside feedback and participation by customers in the product cycle. Ten years ago this would have been unheard of.

Of course, getting the e-mail addresses or phone numbers of the engineers requires Herculean effort from customers, so direct access is largely an illusion at the moment. Some companies even go to the extent of prohibiting such contacts.

However, the option is there for companies smart enough to take advantage of the goldmine of feedback that the free product testers (ie. end-users) can provide. Several companies, such as GDT, who make Apple Macintosh Quickdraw to PCL drivers, actively cruise newsgroups to get in touch with people using their products, new sales leads and competitors - what bothers them, inspires them and so on. Bill Hassell (blh@hpuerca.atl.hp.com) from HP-UX support centre in Atlanta, in his spare (and unpaid) time makes a heroic effort in connecting HP users with the various other parts of HP and offering all sorts of useful printer tips and tricks.

Some of you reading this will say this is a management issue. But it is not. Printers are technical in nature and account for a large amount of lost productivity. Studies indicate that printers account for about a third of all network administration problems. Problems need to be exactly pinpointed for accurate technical solutions to be devised. Yet every layer of bureaucracy means lost time; the chance of the message being garbled increases exponentially and keeps the product designers at unreasonable length from the people who actually use the product and can tell you precisely what improvements are needed.

The new economy and how it affects printer manufacturers.

We can now return to the printer manufacturer's marketing department and Mary's toner smudged glasses.

The glossy images in leading business magazines of network printers in nice offices of large corporations in fact only represents a small percentage of total business users. It is true that the same small percentage buys a lot of printers and demands little in return.

But it is still a small percentage of the total business base. Increasingly though, companies are downsizing and contracting out. The marketers and advertising agencies have apparently not yet caught up with the new economic trends. They still secretly fantasize about the years gone by when things were mass produced and recession and unemployment were unheard of. For advertising agencies and their clients, the sexiest images are of large customers that demand little and have fat cheque books. Account executives provided, in theory at least, a nice clean and cordial abstraction layer. Difficult questions are avoided. In this model, end-users are excluded from messing up such a neat model. This image still persists, and we can only ascribe it to the isolation Mary the marketer lives under in her safe, predicatable, air-conditioned 9-5 job at the office. Many printer companies fit this model.

Allow us to give a real life example. We wanted to buy a 80 page per minute printer/copier for a Fortune 500 client at a cost of about $US 85,000 from an extremely well known industry leader. As a clue, X does not appear in their name. The sales "executives" had beepers and, despite their age being in their mid-twenties, drove BMW's. They were entirely clueless about networking and the driver software for the machine. They had no idea whatsoever about how to contact the engineers or authors of the driver software. When asked about APIs, they though we meant some kind of military equipment !

The list price was an absurdly high ambit amount. As many of you will know, car rental companies quote say 10 cost units per day, but if you meet a qualification, like being present at the airport or having a boarding pass, the cost is often immediately reduced by 40% or more. The same applied in this example, though the sales people were highly indignant, and we quote, that "price is an issue; we thought you wanted a solution". Apparently, they were of the belief that Fortune 500s are not supposed to ask about grubby details like price or technical connectivity. We finally got a price out of them, but we never got the technical details.

These people typify the superficial and inadequate traits of most printer companies' marketing people we have dealt with so far. Their first question, which they ask in order to decide if they even want to bother listening, is how fat the customer's chequebook is, as opposed to the underlying idea or opportunity. And even if the customer passes this test, the marketing folks get angry and indignant at having to answer so many technical questions to make the sale.

You do not find the Marys of this world copying at the local library or Kinko's or thinking about the practical problems small business or mobile users face everyday. They do not experience this, and so their view of the world is extremely limited to a take it or leave it approach.

Mary is isolated. Mary also has considerable psychological resistance to having to deal with uncertain and fickle demands of the new economy, that want everything smaller, cheaper, faster and vastly more sophisticated.

The current situation is rather like IBM failing to see the rise of the PC and Xerox failing to appreciate its PARC creations of windows and the mouse as detailed in *Fumbling the Future* by Douglas Smith & Robert Alexander. The marketplace for printers is rapidly changing. Despite these changes, corporate politics, timidity and a fundamental misunderstanding of the marketplace has ensured that something as simple as a self-serve printer with adequate security capabilities in the firmware for cost-recovery is yet to be invented.

Unlike in the fantasy corporate model, there are no neatly identifiable IT representatives amongst the multitudes of end-users, developers and re-sellers downstream in the new economy. The idea of inviting end-users to participate in product development fills marketing people with dread of being swamped by detailed enquiries and raises the question of why marketing is needed in the first place if they can be bypassed by direct input to the engineers. The instinct for self-preservation by the marketing folks is often a higher priority than the preservation of the company through innovation, though never acknowleged as being paramount.

Brave new world - it's already arrived.

Many organisations are now sharing offices and equipment amongst people who don't even belong to the same company. Some organisations exist for no function other than renting out office facilities to casual users. One of these is called Kinko's Copies, with 700 branches. They pride themselves on their motto "Your branch office". Printers, computers and copiers are their bread and butter. Cost recovery is at the top of their list of things to worry about.

Their spectacular growth is just one manifestation of the new economy which we maintain the marketing departments seem to studiously ignore. They have card readers on the copiers. If only they could have the same smarts in printers that have been in copiers for years !

But there are crucial differences. Like a payphone, the copier is a relatively tamper-proof box. Printers differ in that they involve drivers, cables, networks and often, but not necessarily, printer servers. Every step in the process is vulnerable to wily hackers. Just as if a payphone had the jack it connects to exposed so that anyone could plug in their own $20 phone, a printer can readily have a laptop placed on the network resulting in free copies. Accordingly, we have to look at the firmware in the printer for a solution.

Kinko's are only one of many such chains worldwide that face this problem daily. It is exacerbated by their trying to accommodate laptops which just want to print. This seemingly simple item - much needed in hotels - is regularly plundered by people who take their copies but do not bother paying.

The simple solution is a pre-paid card reader or other payment device and a micro-controller or printer driver that acts as a front end to the printer. Currently this is not possible because printers need - but do not have - the necessary firmware that would enable such a device to buffer the pages, count them and authenticate itself to the printer, finally receiving page-by-page completion from the printer. The same device would also know that pulling out a cable will cause the printer to flush the job, and so closes the cable puller loophole.

The demand for this stuff is right now.

Stanford University is just one of countless victims of the dearth of printer firmware that is long overdue. The present system requires an account on a Unix machine and manual payment by cheque from students in advance. This creates endless problems, such as cables being pulled before the job finishes plus a multitude of regular student hacks on the system. You can get a glimpse of this as a practical application by visiting http://rescomp.stanford.edu/printing.html.

The Stanford system uses a "Macintosh Authenticator" extension which effectively sends the job to a Unix server after authenticating the user and then debiting the user the appropriate amount. There are hundreds of users.

Fortunately, by using HP printers which are PJL equipped, the actual printout can be validated against the page count done by the Unix box - when the cable is in place; when the users do not change the printer IP address and send work to it - and so on. Password protection is achieved indirectly via the Unix box.

The problem with the Stanford system is that it assumes the existence of a printer server, which is expensive to buy and even more expensive to run. It also needs somebody (another salary) to physically guard the printers and hand over the output to users who have to walk down the hall to collect their work. These two items immediately disqualify this as a "solution" for most of the applications where cheap and efficient cost-recovery is most needed.

The Stanford type of solution is common at universities. Auckland University has a similar setup. Interestingly, Auckland also implements encryption of printer traffic. All print jobs are submitted to a Unix box which authenticates the print job (via a user password), takes care of accounting, and spools it to a remote printer controller (an old Macintosh).

A known loophole - and this is still used at places like Kinko's - is to edit PS files to have negative page counts, so that every time something gets printed Harry the Hacker's print quota gets *increased*.

The password protection and cable removal=no page accounting are needed in the printer's firmware. Drivers will quickly be developed to take advantage of this, especially given the rise and rise of the 1284 specification.

Apart from Kinko's Copies, schools and universities, there are the huge numbers of public libraries. There is an entire mailing list dedicated to public computer access from libraries accross the world. Vending CD ROM access and print-outs is a very big item on the agenda. Have any of you, or Mary the marketer, been in a library lately ? On average, there are more computers than photocopy machines.

Another good reason to be here today is to tell you about the frustrated librarians who cannot understand why something as simple as vending photocopies cannot be done on printers because there are insufficient firmware features for developers to write control software for.

Public libraries are big business, as Microsoft's Bill Gates has realised with his recent $3 million sponsorship of American Library Association (ALA). If you have ever seen a coin- or card-operated photocopy machine, this should make you wonder why printers are treated differently and completely ignored by printer vendors to date. Indeed, why ?

When was the last time you used a photocopier at a library or sat down to use a CD-ROM on a computer there and print out the search result ? As we said earlier, many of you have been looking at the world through toner smudged glasses. And believe us, students are time-rich, cash-poor and just plain curious. If there are any weaknesses, expect an attack.

Accounting without security management does not work.

Apart from the copier chains, the schools, the universities, the legions of serviced offices and per-project cost-tracking accounting in larger corporations, all of them environments sharing business equipment, who else needs this ?

Well, there are hotels, cybercafés and the rest of the economy springing up now to meet the needs of telecommuters, road warriors and travellers. Even backpackers are more inclined to use e-mail these days than post restanté.

The cyber café army mentioned above will become (within five years) as ubiquitous as a payphone, and can be expected to match the spectacular growth of the internet. The demand for internet terminal access and e-mail addresses will soon be as common as fax machines. You can expect to see casual-user printers at a hotel soon - firmware changes pending.

Then there are law firms - and there is no shortage of lawyers - as another example. Most legal associates have to account for their time in six minute blocks, such is the penny-pinching and obsession with profits. ,Lawyers, for self-explanatory reasons, do not trust any of their staff, particularly the professional ones, and have had hardware controls internally mounted on their photocopiers and/or PIN number panels since the mid 1980's. Although the same is needed for the office printer, the printer vendors are yet to take up the challenge. Sue, Grabbitt & Run are eagerly waiting.

What you are seeing is the new economy that everyone is talking about. Xerox is looking long and hard at distributed systems of paper matter accross networks, now calling itself the document company. This is precisely what Ray Lutz from Cognisys is looking at.

For some strange reason, printer vendors carry on thinking that their firmware is frozen and this huge slice of the secure cost-recovery market should be ignored.

Compuserve has just taken the "compu" out of their name and allow text-to-speech technology-read email to subscribers via an 800 number. My bet is that the population will continue to become more mobile and that mobility is going to drastically increase the number of printers that will find their way into shared environments needing cost recovery, such as airport lounges, Kinko's (currently about 2500 shared printers and climbing), cybercafés.

The management security on printers is well overdue. The authors believes this is best achieved by a rich - and redundant - number of objects that the host can set or poll, as the case requires. We are sorry to have given such a long (and perhaps repetitious) introduction before getting down to technical matters, but we believe the issues discussed are the root of the problems experienced to date with what is, in reality, a fairly straightforward firmware library modification.

It is time to remove those toner smudged glasses that seem to afflict printer vendors and cater to what is needed right now, the need for which is growing daily. Let's turn to where this fits in with Tom Hasting's work so far:-

Job Monitoring MIB - The Must Have Companion.

Jobmon.doc v.0.7 recycles much of ISO 10175 Document Printing Application (DPA) standard clause 9.2, job attributes. It also needs to be read against jobmontc.mib.

The essential problem is that accounting and information, and indeed SNMP itself **assumes** the connection of a device. In the case of a printer, the device has real costs associated with consumables. Printers are already widely used in cost-recovery situations and this trend is increasing. Colour printers generally use expensive media.

Human nature, in the guise of hackers and non-payers, will attack this very SNMP link upon which everything to do with accounting and who gets to have access hinges upon. The SNMP link is of little use if there is nothing to connect host and printer. This is a fact of life, not a theory.

The author of the monitoring specification has stated specifically in the Change History the following very interesting observations that occurred sometime between version 0.5 and version 0.6, the catalyst apparently being the September PWG:

• Job information is specifically kept for accounting and end-user monitoring
• That the MIB is intended to be used with non ISO DPA print systems
• That the MIB is intended to incorporate SNMP security in its actual use

But most interestingly:-

> "Clarified that if a job control functions (cancel job, etc.) are to be provided, they should be provided by a separate job control MIB that would be a companion MIB to this job monitoring MIB"
>
> "Clarified that this MIB is only for job monitoring and that another MIB should be developed if and when job management (or job control) is wanted, if ever. Reason: job control is controversial. However, having a separate MIB that augments the job monitoring MIB seems a simple way to make everyone happy and removes the completely from the job monitoring MIB efforts"

Controversial ? Battle ?

The two are so closely related that the MIB as it stands should not proceed until management is addressed, battle or no battle !

Management Applications and Needed Features.

The needed real-life application features should drive the MIF inclusions rather than vice-versa. At the expense of being repetitious, some assumptions need to be explicitly articulated:-

• We are assuming that the PWG will be convinced by this paper that there is a large market that uses printers in cost-recovery situations and that this involves human nature and deliberate evasion of the cost-recovery regime.

• Some users will be authorized, but anonymous. Examples include casual users who pay per service. Much of this will take place in site operator unattended automated applications who cannot economically monitor users using staff.

• The "authorization" and authentication for many casual pay-per-service users is whether they have money or creditworthiness. Accordingly, some applications will require real-time billing, just as they do in coin-operated environments.

It is not sufficient that the accounting data is available only *after* the job has finished, because the user is, by definition anonymous and may have already left the site with the (unpaid) work. A payphone is an example where the accounting information needs to be made available at or before the time the service (connection) is made.

The payphone is a locked box, so the accounting information to money collecting device is physically protected. The same cannot be always said of printers. Just as phones deny service without coins, the printer needs to deny service if the crucial accounting accounting information is unavailable to the host because the comms have been disconnected.

• In the case of users who are known and have accounts, just as in the case of unknown and anonymous users, we must assume that some users will deliberately attempt to evade the accounting regime. As the chain is only as strong as the weakest link, we must treat both types of users equally as far as the risk posed by comms cable disconnection is concerned, by denying the very advantage the dishonest user is trying to gain at the expense of the site operator.

• The position of this paper, based on experience, is that accounting without this feature ("the dead man's handle") is pointless in practical public access application environments.

• That accounting information - sensitive commercial information - should be protected from packet sniffer attacks..

• Data sent to a printer needs to be able to be encrypted if chosen by the site operator. In other words, the feature needs to be a mandatory MIB object that may or may not ever be taken advantage of, but leaves users the choice.

Appendix A - MIF Draft.

To manage a network printer effectively requires the ability to keep track of the progress of jobs that are being printed including all information that may be necessary for billing the user and the progress of printing.

Information that is needed includes the number of pages and sheets printed (users might be charged differently for double-sided printing), the type of paper used for each page, and any other information that may be relevant. The progress tracking should be capable of informing the host computer of the progress of printing on a per-page basis and allow the host to send commands to the printer to perform such actions as, say, printing ONLY the first 3 pages.

This could be used in situations where the user can only pay for or have enough credit for 10 pages but they have sent to the printer a 100 page document. The printer could print the first 10 pages and then ask the host what to do next. The host, having determined that the user has no money left, could then ask the printer to discard the rest of the print job.

The printer must have security of configuration data and should have security of print jobs. This means that to configure the printer or to print a job you would need to have the right password. Another optional level of security is encryption of data and passwords transferred over the network. This would ideally be configurable on a per job basis.

Auditing and Accounting

To accurately and effectively track printer usage requires that the printer send regular status messages to the host machine through the entire print process.

It would be easier to just tell the computer the result after the completion of the job, but this would not work in all environments. In hostile environments such as universities, libraries, and internet cafes, it is expected that users would attempt to avoid paying for their printing by performing such actions as unplugging network connections or power cables from the printer. The only way to avoid this is to have the computer tell the printer how many pages to print after it has received credit from the user. An obvious way of doing this would be for the computer to refuse printing until after receiving full payment for the print job. However if the job requires hundreds of pages, then the user will be extremely unhappy if something goes wrong and the job gets cancelled after only a few pages, yet they have paid upfront for other pages that did not print.

It then creates a major administrative hassle to remedy the discrepancy.

The only satisfactory solution to this problem is to charge the user for a given number of pages (say 1 to 5 pages as a suggestion) and then tell the printer to print those pages. After those pages have been printed the printer will inform the host of the fact and ask how many more should be printed. The host will then either accept some further credit from the user and send a command to the printer telling it how many more pages to print or telling it to cancel the job if the user has no more credit.

If the printer does not receive any communication from the computer for a certain amount of time (due to a network failure or the computer being rebooted) then it may be desirable to cancel the print job. This should be a configuration option in the printer.

Security

There are two levels of security that are required for the printing process. Firstly there is security against unauthorised actions and secondly there is encryption to prevent snooping on print jobs.

The actions that have to be controlled by the security system include unauthorised configuration changes and unauthorised printing or cancelling of print jobs. This is can be implemented using separate passwords for printing and configuring the printer. If it is desired to have security over the control of a print job then the job should be sent with a password. That print job should not be able to be cancelled or have its status changed without the use of that password. However we believe that support for job passwords is not a high priority, so we have not included any support for it in the attatched MIF draft.

If encryption is to be used when communicating with the printer, then it should be implemented at a level below the levels of job control and be transparent to the higher level software interfaces. This means that all passwords used for controlling the operation of the printer and all data sent to the printer would be encrypted.

The basic encryption method that we propose using is SAFER encryption. This is because the algorithm is in the public domain and can be freely implemented. It has also been extensively tested and analysed by experts. We also propose the optional usage of other algorithms such as RSA to provide better security applications.

Job Identification

Every print job should have a name which is defined by the computer which generates it. The printer should track jobs by the identification of the computer which sent it and the name of the job. This will allow a printer to maintain a number of print jobs in its spool and to allow host computers to manipulate them individually. This offers the added benefit of having a computer acting as a print server and using the same interfaces as an intelligent printer while supporting dumb printers.

DRAFT MIF data for network printer management

This MIF segment is designed to be incorporated in the DMTF PRINTER MIF DEFINITION V.5. There are some rough edges, so we again stress that this is a draft.

```
//////////////////////////////////////////////////////////////////////////
//                                                                      //
//                      JOB SPOOL TABLE                                 //
//                                                                      //
// The job spool table lists all of the jobs that are currently        //
// spooled in the printer.                                              //
//////////////////////////////////////////////////////////////////////////

start group
      name  = "Job Spool Table"
      id    = XX
      class = "Cyberzone|Job Spool Table|1.00"
      description = "This table lists all of the jobs that are "
                    "currently spooled in the printer's buffer."
      key   = 1,2

start attribute
      name  = "Host Name"
      id    = 1
      description = "The name of the host that printed the job."
      access      = READ-ONLY
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute

start attribute
      name  = "Job Name"
      id    = 2
      description = "The name of the print job."
      access      = READ-ONLY
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute

start attribute
      name  = "Printed Pages"
      id    = 3
      description = "The number of pages printed."
```

```
      access      = READ-ONLY
      storage = SPECIFIC
      type  = INTEGER
      value = 0
end attribute

start attribute
      name  = "Total Pages"
      id    = 4
      description = "The number of pages in the job."
      access      = READ-ONLY
      storage = SPECIFIC
      type  = INTEGER
      value = 0
      // Contains the value 0 if unknown.
end attribute

start attribute
      name  = "Pages to Print"
      id    = 5
      description = "The total number of pages to print before "
                    "pausing and sending status to host."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = INTEGER
      value = 0

      // Contains the value 0 if there is no status message to
      // be sent before the job is complete.
      //
      // If at any time "Printed Pages" is greater than "Pages to Print"
      // then the printer will send a status message to the host and
      // wait for a response.  If the host changes the "Pages to Print"
      // value then printing will resume IF it's new value is greater than
      // "Printed Pages".
end attribute

start attribute
      name  = "Status Timer"
      id    = 6
      description = "The maximum amount of time to elapse "
                    "before sending another status message "
                    "to the host."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = INTEGER
      value = 0
      // Contains the value 0 if there is no status message to
      // be sent before the job is complete.
end attribute

start attribute
      name  = "Print Status"
      id    = 7
      description = "The current status of the print job."
      access      = READ-ONLY
      storage = SPECIFIC
      type  = START ENUM
              name = "Printing Status"
```

```
              type = INTEGER
                 1 = "In Queue"
                 2 = "Printing"
                 3 = "On Hold"
                 4 = "Printed specified number of pages, waiting for
host"
              END ENUM
      value = 1
end attribute

end group


//////////////////////////////////////////////////////////////////////////
//                                                                      //
//                      SECURITY TABLE                                  //
//                                                                      //
// The security table lists all security related information.          //
//////////////////////////////////////////////////////////////////////////

start group
      name  = "Security Table"
      id    = XX
      class = "Cyberzone|Security Table|1.00"
      description = "This table lists all security related "
                    "information and allows it to be altered "
                    "where appropriate."
      key   = 1,2

start attribute
      name  = "Security Type"
      id    = 1
      description = "The type of security record."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = START ENUM
              name = "Security Type Val"
              type = INTEGER
                 1 = "Host Password"
                 2 = "Administration Password"
              END ENUM
end attribute

start attribute
      name  = "Name"
      id    = 2
      description = "The name of the host that the "
                    "password applies to.  Null if "
                    "the administration password is "
                    "being referred to."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute

start attribute
      name  = "Current Password"
      id    = 3
```

```
      description = "The current password - the password must "
                    "be written to this attribute to change it."
      access      = WRITE-ONLY
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute

start attribute
      name  = "New Password"
      id    = 4
      description = "The new password.  Will only be "
                    "accepted if Current Password matches."
      access      = WRITE-ONLY
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute

end group

//////////////////////////////////////////////////////////////////////////
//                                                                      //
//                      CONTROL TABLE                                   //
//                                                                      //
// The security table lists all network printer configuration data.    //
//////////////////////////////////////////////////////////////////////////
//
//
start group
      name  = "Control Table"
      id    = XX
      class = "Cyberzone|Control Table|1.00"
      description = "This table stores all data related to "
                    "maintaining control over the printing "
                    "process."
      key   = 1,2

start attribute
      name  = "Host Name"
      id    = 1
      description = "The name of the host that printed the job."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute

start attribute
      name  = "Job Name"
      id    = 2
      description = "The name of the print job.  Null for settings "
                    "which apply to all jobs from the host."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute
```

```
start attribute
      name  = "Current Password"
      id    = 3
      description = "The current password - the password must "
                    "be written to this attribute to allow changes."
      access      = WRITE-ONLY
      storage = SPECIFIC
      type  = STRING (255)
      value = ""
end attribute

start attribute
      name  = "Host Timeout"
      id    = 4
      description = "The time to wait for confirmation from the "
                    "host before cancelling a job.  Measured in "
                    "seconds."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = INTEGER
      value = 0  // 0 means no waiting.
end attribute

start attribute
      name  = "Status Time"
      id    = 5
      description = "The amount of time to elapse between status "
                    "messages sent to the host.  In seconds."
      access      = READ-WRITE
      storage = SPECIFIC
      type  = INTEGER
      value = 0

// 0 means no timed status messages.  Status messages
// will still be sent as a result of "Pages To Print".
end attribute

end group
```

Other discussion points:

1. Consider printer assigned job id No. to prevent likelihood or print servers using same name and or resolving conflicts. Although this goes against the basic SNMP philosophy, it should be available as a optional redundant feature where "smarts", such as each server knowing what the other server is doing, are not always necessarily avaiable. This could be a text field.

2. Page completion advice, defined as when the page is marked and output to the out tray. This should also have an few bits to show paper size, tray selected and whether printed duplex. This redundant information is very useful from the application perspective.

FF (unique job number) plus nibble (page size = 0-F) plus nibble (duplex true = value <0) = 16 bits.

3. Job completion advice including printer assigned ID (FF job id plus TRUE bit)

4. The bit mode is merely suggested. This may be better used as text.

Appendix B
Industry Round Up & the Gossip.

We have only included the languages we were able to dig up. Some of these are separate control and page description languages, whilst others are combined. This list will undoubtedly be immediately assailed as being too limited or the observations as being unfair etc.

Bear in mind, before complaining, that if you work for a company you feel has been maligned, that you have all sorts of special information access and knowledge. It is *your* perspective that is peculiar, not that of the end-user or developer. This round-up reflects the actual life experience that "outsiders" face; hopefully it will shed light on the hurdles buyers and developers face getting information and the marketing consequences that follow. As it is, printer manufacturs still have the same "closed architecture" mentality that used to plague the PABX and key telephone system industry.

It is hoped that this paper, by creating awareness of what is out there and how it has fared, will encourage the best features to be adopted as mandatory objects into the PWG's job submission, security, monitoring and accounting MIF, and that this will become a lasting and useful legacy of the PWG's work.

DOC*IT - Created by OKI. Includes page-by-page completion notification as well as password protection on job submission. OKI suffers from the usual industry illness whereby it is impossible to get information easily on which of their products this is implemented. In addition, it is understood that DOC*IT is being phased out in favour of another unspecified standard. OKI was able to make an API manual and disk of code available for the cost of a phone call. Getting to that point took a great deal of persistent digging on the internet and toll calls. Somehow the DOC*IT message never made it to OKI's resellers. It was only discovered by chance thanks to OKI's web page.

Imaging Device Protocol (IDP) - created by Apple Computers. The details of what this does, which printers it has been incorporated on and so on are are still being formulated.

Intelligent Printer Data Streaming (IPDS) - creation of IBM circa 1984 that did not get hived off to Lexmark. IPDS has all the page-by-page security, monitoring and so on one could ever hope for. Its problem is that it was always aimed at mainframes, and more recently OS/2 Advanced Server LANS and AS400. Exactly what constitutes IPDS compliance is vague and frequently shifting. Commercially, IPDS has not broken out of its niche and is sufficiently complex that there is no possibility of mapping anything other than a tiny fraction of its features to an SNMP type MIB. Full details of the object architecture can be found in IBM publication No. S544-3417-03.

PostScript Level II - Adobe. This is so well-known I need not say more other than it is widely in use and also has emulations of it by Japanese companies. It is very well documented and has several newsgroups on the internet devoted to it. To add confusion, it is possible to enscapulate PostScript jobs inside the HP PJL (q.v) environment so as to effectively import limited page accounting and security. However, as documented at

Auckland and Kinko's, people can and have edited postcript files by putting in negative values as a means of page accounting evasion.

Printer Job Language (PJL) - Hewlett Packard. Closely associated with HP's Printer Control Language (PCL) currently at Level 5 (enhanced) (also called PCL 5e). It differs from PCL in that PJL is a control language whereas PCL is a Page Description Language (PDL). This gives page-by-page completion and job completion as unsolicited data via the USTATUS command.

To understand the applications, it must be remembered that the HP printer JetDirect cards have TCP and telnet capabilities, the most significant from the perspective of this paper being 15 byte password protection built into the JetDirect Flash ROM version A.03.06 or higher that has been around since about April 1995.

Network Printer Alliance Protocol (NPAP) - Technically, NPAP can make SNMP calls to take advantage of what is on offer in the DMTF MIF. Since the DMTF MIF follows the SNMP common printer MIB, variable for variable, NPAP can be used to query the SNMP variable for a DMTF request. The latest version of the Lexmark NPAP offers page complete alerts, job status alerts as well as querying active or queued jobs. See the "Register for Lexmark Extension Alerts" in the Lexmark NPAP's API specification.

Appendix C.

The following is suggested as a royalty free, secure, fast and relatively processor non-intensive encryption algorithm. The PWG could later expand this to include RSA if it chooses, but, we consider that all factors taken into account, this is the SAFER option ! To kick this along, Cyberzone is willing to do the work needed on this to port, test and troubleshoot the the the code to specific assembly languages.

```
/* -- SAFER.H -- */

/* The number of rounds for the different SAFER variants.  The maximum of
13 rounds is set by the requirements of the key schedule operation */

#define SAFER_K64_ROUNDS      6

#define SAFER_K128_ROUNDS     10

#define SAFER_SK64_ROUNDS     8

#define SAFER_SK128_ROUNDS    10

#define SAFER_MAX_ROUNDS      13

/* The SAFER cipher block and key size */

#define SAFER_BLOCKSIZE       8           /* 64 bits */

#define SAFER_KEYSIZE   16          /* 128 bits */

/* The SAFER internal key length */

#define SAFER_KEYLEN     ( 1 + SAFER_BLOCKSIZE * ( 1 + 2 *
SAFER_MAX_ROUNDS ) )
```

```
/* -- SAFER.C -- */

/* Code to implement the improved SAFER-SK form of the SAFER cipher,
originally published as "SAFER K-64: A Byte-Oriented Block-Ciphering
Algorithm", James L. Massey, "Fast Software Encryption", Lecture Notes in
Computer Science No. 809, Springer-Verlag 1994, p.1. This code implements
the 128-bit key extension designed by the Special Projects Team of the
Ministry of Home Affairs, Singapore and published as "SAFER K-64: One
Year Later", James L.Massey, presented at the K. U. Leuven Workshop on
Algorithms, Leuven, Belgium, 14-16 December, 1994, to appear in "Fast
Software Encryption II", Lecture Notes in Computer Science, Springer-
Verlag 1995, along with Lars Knudsen's strengthened key schedule,
presented in "A Key-Schedule Weakness in SAFER K-64," Lars Knudsen,
presented at Crypto '95 in Santa Barbara, California. All parts of SAFER-
SK are non-proprietary and freely available for anyone to use as they see
fit. This version improved by Geoff Slater & Peter Gutmann */


#include <string.h>

#include "defs.h"


/* The size of each half key used in the key schedule */



#define SAFER_KEYHALF_SIZE         ( SAFER_KEYSIZE / 2 )


/* The size of each half key plus the parity byte used in the key
schedule */


#define SAFER_KEYHALF_P_SIZE ( SAFER_KEYHALF_SIZE + 1 )


/* Define for byte rotates (C isn't quite a high-level assembly language)
*/


#define ROL(x,n)  ( ( ( x << n ) | ( x >> ( 8 - n ) ) ) & 0xFF )


/* Exponents and logs, evaluated via the pre-calculated lookup tables */


#define EXP(x)    expTable[ x ]

#define LOG(x)    logTable[ x ]


/* The two-point Pseudo-Hadamard Transform

      b1 = 2a1 + a2

      b2 =  a1 + a2
```

```
and inverse two-point Pseudo-Hadamard Transform

      a1 =  b1 -  b2

      a2 = -b1 + 2b2

which are used to create a three-dimensional PHT (ie independant two-
point PHT's in each of three dimensions, which is why there are 2^3 = 8
bytes in the input and output of the PHT) through a decimation- by-
two/fanning-out-by-two network.  The PHT provides guaranteed complete
diffusion within one linear layer */

#define PHT(x,y)  { y += x; x += y; }

#define IPHT(x,y)       { x -= y; y -= x; }



/* The lookup table for logs and exponents.  These contain the powers of
the primitive element 45 of GF( 257 ) (ie values of 45^n mod 257) in
"expTable" with the corresponding logs base 45 stored in "logTable".

They may be calculated as follows:

      exponent = 1;

      for( i = 0; i < 256; i++ )
            {

            int exp = exponent & 0xFF;

            expTable[ i ] = exp;

            logTable[ exp ] = i;

            exponent = ( exponent * 45 ) % 257;

            } */

static BYTE expTable[] = {

      0x01, 0x2D, 0xE2, 0x93, 0xBE, 0x45, 0x15, 0xAE,

      0x78, 0x03, 0x87, 0xA4, 0xB8, 0x38, 0xCF, 0x3F,

      0x08, 0x67, 0x09, 0x94, 0xEB, 0x26, 0xA8, 0x6B,

      0xBD, 0x18, 0x34, 0x1B, 0xBB, 0xBF, 0x72, 0xF7,

      0x40, 0x35, 0x48, 0x9C, 0x51, 0x2F, 0x3B, 0x55,

      0xE3, 0xC0, 0x9F, 0xD8, 0xD3, 0xF3, 0x8D, 0xB1,

      0xFF, 0xA7, 0x3E, 0xDC, 0x86, 0x77, 0xD7, 0xA6,

      0x11, 0xFB, 0xF4, 0xBA, 0x92, 0x91, 0x64, 0x83,

      0xF1, 0x33, 0xEF, 0xDA, 0x2C, 0xB5, 0xB2, 0x2B,
```

```
        0x88, 0xD1, 0x99, 0xCB, 0x8C, 0x84, 0x1D, 0x14,
        0x81, 0x97, 0x71, 0xCA, 0x5F, 0xA3, 0x8B, 0x57,
        0x3C, 0x82, 0xC4, 0x52, 0x5C, 0x1C, 0xE8, 0xA0,
        0x04, 0xB4, 0x85, 0x4A, 0xF6, 0x13, 0x54, 0xB6,
        0xDF, 0x0C, 0x1A, 0x8E, 0xDE, 0xE0, 0x39, 0xFC,
        0x20, 0x9B, 0x24, 0x4E, 0xA9, 0x98, 0x9E, 0xAB,
        0xF2, 0x60, 0xD0, 0x6C, 0xEA, 0xFA, 0xC7, 0xD9,
        0x00, 0xD4, 0x1F, 0x6E, 0x43, 0xBC, 0xEC, 0x53,
        0x89, 0xFE, 0x7A, 0x5D, 0x49, 0xC9, 0x32, 0xC2,
        0xF9, 0x9A, 0xF8, 0x6D, 0x16, 0xDB, 0x59, 0x96,
        0x44, 0xE9, 0xCD, 0xE6, 0x46, 0x42, 0x8F, 0x0A,
        0xC1, 0xCC, 0xB9, 0x65, 0xB0, 0xD2, 0xC6, 0xAC,
        0x1E, 0x41, 0x62, 0x29, 0x2E, 0x0E, 0x74, 0x50,
        0x02, 0x5A, 0xC3, 0x25, 0x7B, 0x8A, 0x2A, 0x5B,
        0xF0, 0x06, 0x0D, 0x47, 0x6F, 0x70, 0x9D, 0x7E,
        0x10, 0xCE, 0x12, 0x27, 0xD5, 0x4C, 0x4F, 0xD6,
        0x79, 0x30, 0x68, 0x36, 0x75, 0x7D, 0xE4, 0xED,
        0x80, 0x6A, 0x90, 0x37, 0xA2, 0x5E, 0x76, 0xAA,
        0xC5, 0x7F, 0x3D, 0xAF, 0xA5, 0xE5, 0x19, 0x61,
        0xFD, 0x4D, 0x7C, 0xB7, 0x0B, 0xEE, 0xAD, 0x4B,
        0x22, 0xF5, 0xE7, 0x73, 0x23, 0x21, 0xC8, 0x05,
        0xE1, 0x66, 0xDD, 0xB3, 0x58, 0x69, 0x63, 0x56,
        0x0F, 0xA1, 0x31, 0x95, 0x17, 0x07, 0x3A, 0x28
        };

static BYTE logTable[] = {
        0x80, 0x00, 0xB0, 0x09, 0x60, 0xEF, 0xB9, 0xFD,
        0x10, 0x12, 0x9F, 0xE4, 0x69, 0xBA, 0xAD, 0xF8,
        0xC0, 0x38, 0xC2, 0x65, 0x4F, 0x06, 0x94, 0xFC,
        0x19, 0xDE, 0x6A, 0x1B, 0x5D, 0x4E, 0xA8, 0x82,
```

```
    0x70, 0xED, 0xE8, 0xEC, 0x72, 0xB3, 0x15, 0xC3,
    0xFF, 0xAB, 0xB6, 0x47, 0x44, 0x01, 0xAC, 0x25,
    0xC9, 0xFA, 0x8E, 0x41, 0x1A, 0x21, 0xCB, 0xD3,
    0x0D, 0x6E, 0xFE, 0x26, 0x58, 0xDA, 0x32, 0x0F,
    0x20, 0xA9, 0x9D, 0x84, 0x98, 0x05, 0x9C, 0xBB,
    0x22, 0x8C, 0x63, 0xE7, 0xC5, 0xE1, 0x73, 0xC6,
    0xAF, 0x24, 0x5B, 0x87, 0x66, 0x27, 0xF7, 0x57,
    0xF4, 0x96, 0xB1, 0xB7, 0x5C, 0x8B, 0xD5, 0x54,
    0x79, 0xDF, 0xAA, 0xF6, 0x3E, 0xA3, 0xF1, 0x11,
    0xCA, 0xF5, 0xD1, 0x17, 0x7B, 0x93, 0x83, 0xBC,
    0xBD, 0x52, 0x1E, 0xEB, 0xAE, 0xCC, 0xD6, 0x35,
    0x08, 0xC8, 0x8A, 0xB4, 0xE2, 0xCD, 0xBF, 0xD9,
    0xD0, 0x50, 0x59, 0x3F, 0x4D, 0x62, 0x34, 0x0A,
    0x48, 0x88, 0xB5, 0x56, 0x4C, 0x2E, 0x6B, 0x9E,
    0xD2, 0x3D, 0x3C, 0x03, 0x13, 0xFB, 0x97, 0x51,
    0x75, 0x4A, 0x91, 0x71, 0x23, 0xBE, 0x76, 0x2A,
    0x5F, 0xF9, 0xD4, 0x55, 0x0B, 0xDC, 0x37, 0x31,
    0x16, 0x74, 0xD7, 0x77, 0xA7, 0xE6, 0x07, 0xDB,
    0xA4, 0x2F, 0x46, 0xF3, 0x61, 0x45, 0x67, 0xE3,
    0x0C, 0xA2, 0x3B, 0x1C, 0x85, 0x18, 0x04, 0x1D,
    0x29, 0xA0, 0x8F, 0xB2, 0x5A, 0xD8, 0xA6, 0x7E,
    0xEE, 0x8D, 0x53, 0x4B, 0xA1, 0x9A, 0xC1, 0x0E,
    0x7A, 0x49, 0xA5, 0x2C, 0x81, 0xC4, 0xC7, 0x36,
    0x2B, 0x7F, 0x43, 0x95, 0x33, 0xF2, 0x6C, 0x68,
    0x6D, 0xF0, 0x02, 0x28, 0xCE, 0xDD, 0x9B, 0xEA,
    0x5E, 0x99, 0x7C, 0x14, 0x86, 0xCF, 0xE5, 0x42,
    0xB8, 0x40, 0x78, 0x2D, 0x3A, 0xE9, 0x64, 0x1F,
    0x92, 0x90, 0x7D, 0x39, 0x6F, 0xE0, 0x89, 0x30
    };

/* Perform a SAFER key schedule */
```

```
void saferExpandKey( BYTE *key, BYTE *userKey, int noRounds, int
useSaferSK )

      {

      int round, i;

      BYTE *keyLow = userKey, *keyHigh = userKey + SAFER_KEYHALF_SIZE;

      BYTE ka[ SAFER_KEYHALF_P_SIZE ], kb[ SAFER_KEYHALF_P_SIZE ];

      /* Save the number of rounds as part of the key */

      if( noRounds > SAFER_MAX_ROUNDS )

            noRounds = SAFER_MAX_ROUNDS;

      *key++ = ( BYTE ) noRounds;


      /* Copy the user key halves to Ka and Kb */

      for( i = 0; i < SAFER_KEYHALF_SIZE; i++ )

            {

            ka[ i ] = keyLow[ i ];

            kb[ i ] = keyHigh[ i ];

            }

      /* Append a parity byte to keys Ka and Kb */

      ka[ SAFER_KEYHALF_SIZE ] = kb[ SAFER_KEYHALF_SIZE ] = 0;

      for( i = 0; i < SAFER_KEYHALF_SIZE; i++ )

            {

            ka[ SAFER_KEYHALF_SIZE ] ^= ka[ i ];

            kb[ SAFER_KEYHALF_SIZE ] ^= kb[ i ];

            }


      /* K1 = Kb */

      for( i = 0; i < SAFER_KEYHALF_SIZE; i++ )

            *key++ = kb[ i ];


      /* Rotate each byte of Ka right by 3 */
```

```
      for( i = 0; i < SAFER_KEYHALF_SIZE + 1; i++ )

            ka[ i ] = ROL( ka[ i ], 5 );


/* Perform the key schedule needed to derive the remaining keys
K2,..K2r+1 from the 128 bit input key Ka+Kb */

      for( round = 1; round <= noRounds; round++)

            {

            /* Left rotate each byte of Ka and Kb by 6 */

            for( i = 0; i < SAFER_KEYHALF_P_SIZE; i++ )

                  {

                  ka[ i ] = ROL( ka[ i ], 6 );

                  kb[ i ] = ROL( kb[ i ], 6 );

                  }


/* Add the key biases to give K2i-1 and K2i.  The original algorithm
specification is:


k[ 2 * i, j ] = ka[ ( ( j + 2 * i - 2 ) % 9 ) + 1 ] + \ expTable[
expTable[ 18 * i + j ] ];

k[ 2 * i + 1, j ] = kb[ ( ( j + 2 * i - 1 ) % 9 ) + 1 ] + \expTable[
expTable[ 18 * i + 9 + j ] ];


however we rearrange this to calculate K2i-1 and K2i seperatelyto
eliminate the need to repeatedly evaluate 2i on the LHS */

            for( i = 0; i < SAFER_KEYHALF_SIZE; i++ )

                  if( useSaferSK )

                        *key++ = ( ka[ ( i + 2 * round - 1 ) %
SAFER_KEYHALF_P_SIZE ] + \

                                               expTable[ expTable[ (
SAFER_KEYHALF_P_SIZE * 2 ) * \

                                                  round + i + 1 ] ] ) & 0xFF;

            else

                        *key++ = ( ka[ i ] + \

                                               expTable[ expTable[ (
SAFER_KEYHALF_P_SIZE * 2 ) * \

                                                  round + i + 1 ] ] ) & 0xFF;
```

```
            for( i = 0; i < SAFER_KEYHALF_SIZE; i++ )

                  if( useSaferSK )

                        *key++ = ( kb[ ( i + 2 * round ) %
SAFER_KEYHALF_P_SIZE ] + \

                                       expTable[ expTable[ (
SAFER_KEYHALF_P_SIZE * 2 ) * \

                                          round + i + (
SAFER_KEYHALF_P_SIZE + 1 ) ] ] ) & 0xFF;

            else

                        *key++ = ( kb[ i ] + \

                                       expTable[ expTable[ (
SAFER_KEYHALF_P_SIZE * 2 ) * \

                                          round + i + (
SAFER_KEYHALF_P_SIZE + 1 ) ] ] ) & 0xFF;

            }


      /* Clean up */

      memset( ka, 0, SAFER_BLOCKSIZE );

      memset( kb, 0, SAFER_BLOCKSIZE );

      }


/* Encrypt a block of data with SAFER */


void saferEncryptBlock( BYTE *data, BYTE *key )

      {

      BYTE a, b, c, d, e, f, g, h, t;

      int rounds = *key++;


      /* Copy the input block to local variables */

      a = data[ 0 ];

      b = data[ 1 ];

      c = data[ 2 ];
```

```
      d = data[ 3 ];
      e = data[ 4 ];
      f = data[ 5 ];
      g = data[ 6 ];
      h = data[ 7 ];

      while( rounds-- )
            {
            /* Perform the mixed xor/byte addition of the round input
with the
               subkey K2i-1, combined with the first level of the
nonlinear
               layer, either 45^n mod 257 or log45n, and the mixed
xor/byte
               addition with the subkey K2i */
            a = EXP( ( a ^ key[ 0 ] ) & 0xFF ) + key[  8 ];
            b = LOG( ( b + key[ 1 ] ) & 0xFF ) ^ key[  9 ];
            c = LOG( ( c + key[ 2 ] ) & 0xFF ) ^ key[ 10 ];
            d = EXP( ( d ^ key[ 3 ] ) & 0xFF ) + key[ 11 ];
            e = EXP( ( e ^ key[ 4 ] ) & 0xFF ) + key[ 12 ];
            f = LOG( ( f + key[ 5 ] ) & 0xFF ) ^ key[ 13 ];
            g = LOG( ( g + key[ 6 ] ) & 0xFF ) ^ key[ 14 ];
            h = EXP( ( h ^ key[ 7 ] ) & 0xFF ) + key[ 15 ];


            /* Perform the Pseudo-Hadamard Trasform of the round output
*/
            PHT( a, b );
            PHT( c, d );
            PHT( e, f );
            PHT( g, h );
            PHT( a, c );
            PHT( e, g );
```

```
            PHT( b, d );
            PHT( f, h );
            PHT( a, e );
            PHT( b, f );
            PHT( c, g );
            PHT( d, h );
            t = b; b = e; e = c; c = t; t = d; d = f; f = g; g = t;
            key += 16;
            }

        /* Perform the final mixed xor/byte addition output transformation
           using K2r + 1*/
        data[ 0 ] = a ^ key[ 0 ];
        data[ 1 ] = b + key[ 1 ];
        data[ 2 ] = c + key[ 2 ];
        data[ 3 ] = d ^ key[ 3 ];
        data[ 4 ] = e ^ key[ 4 ];
        data[ 5 ] = f + key[ 5 ];
        data[ 6 ] = g + key[ 6 ];
        data[ 7 ] = h ^ key[ 7 ];
        }

/* Decrypt a block of data with SAFER */

void saferDecryptBlock( BYTE *data, BYTE *key )
        {
        BYTE a, b, c, d, e, f, g, h, t;
        int rounds = *key;
```

```
	/* Since we're now running throught the algorithm backwards, we
move to

	   the end of the key and start from there */

	key += SAFER_BLOCKSIZE * ( 1 + 2 * rounds );


	/* Perform the initial mixed xor/byte addition input transformation

	   using K2r+1 */

	a = data[ 0 ] ^ key[ -7 ];

	b = data[ 1 ] - key[ -6 ];

	c = data[ 2 ] - key[ -5 ];

	d = data[ 3 ] ^ key[ -4 ];

	e = data[ 4 ] ^ key[ -3 ];

	f = data[ 5 ] - key[ -2 ];

	g = data[ 6 ] - key[ -1 ];

	h = data[ 7 ] ^ key[ 0 ];

	key -= 8;


	while( rounds-- )

		{

		t = e; e = b; b = c; c = t; t = f; f = d; d = g; g = t;


		/* Perform the inverse Pseudo-Hadamard Trasform of the round
input */

		IPHT( a, e );

		IPHT( b, f );

		IPHT( c, g );

		IPHT( d, h );

		IPHT( a, c );

		IPHT( e, g );

		IPHT( b, d );

		IPHT( f, h );
```

```
            IPHT( a, b );

            IPHT( c, d );

            IPHT( e, f );

            IPHT( g, h );


            /* Perform the mixed xor/byte addition of the inverse PHT
output with

               the subkey K2r+2-2i, combined with the second level of the

               nonlinear layer, either 45^n mod 257 or log45n, and
finally the

               mixed xor/byte addition of the round output with K2r+1-2i
*/

            h = LOG( ( h - key[  0 ] ) & 0xFF ) ^ key[  -8 ];

            g = EXP( ( g ^ key[ -1 ] ) & 0xFF ) - key[  -9 ];

            f = EXP( ( f ^ key[ -2 ] ) & 0xFF ) - key[ -10 ];

            e = LOG( ( e - key[ -3 ] ) & 0xFF ) ^ key[ -11 ];

            d = LOG( ( d - key[ -4 ] ) & 0xFF ) ^ key[ -12 ];

            c = EXP( ( c ^ key[ -5 ] ) & 0xFF ) - key[ -13 ];

            b = EXP( ( b ^ key[ -6 ] ) & 0xFF ) - key[ -14 ];

            a = LOG( ( a - key[ -7 ] ) & 0xFF ) ^ key[ -15 ];

            key -= 16;

            }


      data[ 0 ] = a;

      data[ 1 ] = b;

      data[ 2 ] = c;

      data[ 3 ] = d;

      data[ 4 ] = e;

      data[ 5 ] = f;

      data[ 6 ] = g;

      data[ 7 ] = h;
```

```
        }


/* A Rough and Ready Test Function */

#include <stdio.h>

void main( void )

        {

        BYTE key1[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,

                                        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x01 };

        BYTE plainText1[] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08 };

        BYTE cipherText1[] = { 0x41, 0x4C, 0x54, 0x5A, 0xB6, 0x99, 0x4A,
0xF7 };

        BYTE key2[] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08,

                                        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00 };

        BYTE plainText2[] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08 };

        BYTE cipherText2[] = { 0xFF, 0x78, 0x11, 0xE4, 0xB3, 0xA7, 0x2E,
0x71 };

        BYTE key3[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

                                        0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08 };

        BYTE plainText3[] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
0x08 };

        BYTE cipherText3[] = { 0x49, 0xC9, 0x9D, 0x98, 0xA5, 0xBC, 0x59,
0x08 };

        BYTE data[ SAFER_BLOCKSIZE ], key[ SAFER_KEYLEN ];


        saferExpandKey( key, key1, SAFER_SK128_ROUNDS, TRUE );

        memcpy( data, plainText1, SAFER_BLOCKSIZE );

        saferEncryptBlock( data, key );

        if( memcmp( data, cipherText1, SAFER_BLOCKSIZE ) )

        puts( "Bang" );
```

```
saferDecryptBlock( data, key );
saferExpandKey( key, key2, SAFER_SK128_ROUNDS, TRUE );
memcpy( data, plainText2, SAFER_BLOCKSIZE );
saferEncryptBlock( data, key );
if( memcmp( data, cipherText2, SAFER_BLOCKSIZE ) )
      puts( "Bang" );
saferDecryptBlock( data, key );
saferExpandKey( key, key3, SAFER_SK128_ROUNDS, TRUE );
memcpy( data, plainText3, SAFER_BLOCKSIZE );
saferEncryptBlock( data, key );
if( memcmp( data, cipherText3, SAFER_BLOCKSIZE ) )
      puts( "Bang" );
saferDecryptBlock( data, key );
}
```

Appendix D

Sources.

Hewlett Packard, *Printer Job Language Technical Reference Manual* , HP, 1995.

Pfleeger, C, *Security in Computing*, 1989, Prentice Hall, New Jersey, p.459 ff.

Reinhardt , A, "Smarter Copiers, Printers and Fax Devices Are Coming", *Byte*, November 1994 at p.82.

Smith, N, *Getting the Most From Your HP LaserJet* , Wordware, Texas, 1995, p.65.

Unknown, "Laser Printers: The Price is Right", *Byte*, February 1995 p. 131.

And other selected articles in numerous magazines, internet news groups and a great deal of personal experience of the authors.

The authors

Russell Coker - programmer, tester & GUI designer as well the proprietor of Virtual Café, Melbourne, Australia. Fan of OS/2 and IMAP. (rjc@virtual.net.au).

Geoffrey Slater - R & D co-ordinator & programmer for Cyberzone and contractor to numerous US based computer chains, libraries & cyber cafés. (ask@cyberzone.com.au)

Peter Gutmann - part of the Pretty Good Privacy (PGP) and Viacrypt team along with Phil Zimmerman and Colin Plumb; inventor of MDC. Peter is a cryptologist, computer security specialist & PhD candidate in compression mathematics. He also programmes for Cyberzone. (pgut01@cs.auckland.ac.nz).

Common problems in cost-recovery or shared printer situations:-

1. Direct connection of unauthorized laptop for free printing.
2. Postscript dumps directly to printer to bypass cost-recovery driver software
3. Pulling out cable after buffering to stop accounting information from printer to host
4. Packet sniffers used to covertly learn the contents of documents sent for printing
5. Packet sniffers to get passwords - the seemingly innocuous printer server password is often the same as passwords that allow hackers into servers & so on.
6. Poorly written cost-recovery drivers causing 50 pages of raw postscript code to print
7. Each driver is printer specific and maintainence a problem
8. Printing more than available credit
9. Cost of administration and human staff to monitor printer.
10. Cost of proprietory UNIX/Server box and network connectivity.

Common user complaints for refund requests:-

• print jobs sent to the wrong printer
• print jobs that the user fails to pick up from the printer
• user sending multiple copies of a print job to a printer
• user printing the "wrong" document
• print jobs delayed several minutes due to heavy user load