CONTENTS

COLOUR PLATES

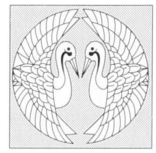
Front cover:

Zulaykhā's maidens struck by the beauty of Yūsuf, a miniature from the St. Petersburg Branch of the Institute of Oriental Studies manuscript *Yūsuf wa Zulaykhā* by Jāmī (call number B 2325), fol. 102b, 7.7 × 7.8 cm (see pp. 62—64).

Back cover:

Plate 1. Merchants rescuing Yūsuf on their way to Miṣr with a caravan, a miniature from the same manuscript, fol. 61a, 8.2 × 7.8 cm.
Plate 2. Yūsuf shepherding Zulaykhā's flock of sheep, a miniature from the same manuscript, fol. 72a, 8.8 × 7.8 cm.
Plate 3. Zulaykhā bringing Yūsuf to her Seventh Palace where he rejects her courting, a miniature from the same manuscript, fol. 90b, 8.9 × 7.8 cm.
Plate 4. Obeying heavenly command Yūsuf who marries Zulaykhā after her adopting Islam, a miniature from the same manuscript, fol. 132a, 7.7 × 7.8 cm.

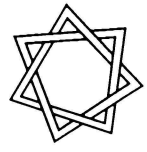
RUSSIAN ACADEMY OF SCIENCES
THE INSTITUTE OF ORIENTAL STUDIES
ST. PETERSBURG BRANCH



Manuscripta Orientalia

International Journal for Oriental Manuscript Research

Vol. 2 No. 4 December 1996



TSESA
St. Petersburg-Helsinki

ORIENTAL MANUSCRIPTS AND NEW INFORMATION TECHNOLOGIES

Correspondence Round Table

ITISALAT DISCUSSION OF CD-ROM PROTECTION/PIRACY PROBLEM

In *Manuscripta Orientalia*, vol. 2, No. 2 (1996), the article "A robust and versatile solution for the digital publication of manuscript material" by Hansje Braam and Marc Vandamme (Utrecht University) was published. The discussion of some ideological and technical problems concerning CD-ROM editions of the manuscript heritage, which are being realized now in several libraries and research centers, was proposed there. The information about the similar program of the Vatican library (IBM technical and software support) shows that the idea of using digital technologies for this purpose has a good future.

Nearly simultaneously with the publication in *Manuscripta Orientalia*, the subscribers of ITISALAT (The Internet Forum on Arabic Computing) started discussion on the protection of CD software against piracy. The discussion seems to be a real success of ITISALAT. It is so interesting that we decided to propose correspondence on the problem to the attention of the *Manuscripta Orientalia* audience which is much wider than that of ITISALAT.

Among the most active participants of the discussion we have to mention George N. Hallak, ARAMedia Group, Arabization & Software Center, who initialized the discussion; Adrian Brockett, Quest MultiMedia; Abdel-hadi, Systems Engineer/Nike; Andrew Freeman, Department of Near Eastern Studies, University of Michigan; Alec McAllister, Arts Computing Development Officer, Computing Service, University of Leeds.

You may remember that issue No. 3 of vol. 1 (1996) of *Manuscripta Orientalia* also contained the material connected with the ITISALAT (it was Arabic/Farsi OCR that was discussed). Proposing the present discussion, we would like to wish all *Manuscripta Orientalia* and ITISALAT subscribers an outright victory over piracy and pirates, who sometimes intend not only to steal our work but also to destroy our private life.

A bas les piratos!

Efim Rezvan,
Manuscripta Orientalia Deputy Editor-in-Chief

***Paul Roochnik**, ITISALAT Moderator*

We have been working very hard on our upcoming release of a CD-ROM in English, French, and Arabic for Windows and Mac. Our developers are utilizing Adobe's Acrobat 2.01 (.pdf) technology.

We are writing to you requesting your help and recommendations for means of protection and security of our work on the CD-ROM against piracy.

We kindly ask you to advise us if you know of such a product. If so, please post or e-mail the source of such product or company that works best with Acrobat, preserving the integrity of our work.

In case this is the wrong place to request such information, please advise of a good source to contact for such information.

Thank you,

George N. Hallak

We protect our CDs with "Laserlock" technology which is proving very effective. The games industry is using it more and more. It works on both hardware and software levels... You can't copy to another CD. You can copy to a hard disk BUT you have to have the original disc in the CD drive to run your program. We'll be happy to forward any further information.

Adrian Brockett

---

* To subscribe, send the command (A) to address (B): (A) — subscribe ITISALAT your-1st-name your-last-name (B) — listserv@listserv.georgetown.edu (for more information, contact: Dr. Paul Roochnik, Moderator, ITISALAT e-mail: roochnik@ios.com).

The Middle East Market is infested with pirates, when my principals called me from Beirut, they had that in mind.

Pirates are very creative, and I do not have a direct answer to your question, not understanding the difference.

If you have two separate solutions, please post or e-mail to me... Who knows what the publisher, in Beirut, is thinking... They are looking into all kinds of solutions.

One, consumer friendly, way is lower the price of the CD to under $40.00 (?), deeming it invaluable for the pirates to copy, master, or whatever. Please, keep the information flowing... Thank you.

George N. Hallak

There's isn't much you can do about this. Hackers/crackers are more creative than copy protection guys. If you want a simple cheap copy protection idea from me for a CD, fill the CD with 650 meg of files. Some files belong to the program and some are useless. If the pirate wants to copy the CD, he better had a lot of hard drive space and tons of floppies. Forget modem transfers. If he's clever, let him figure out which files are genuine!

You don't need that high tech of Laserlock. You can't write to a CD, therefore, what you can do is to try to write to the CD, if it failed then you have an original CD.

Abdel-hadi

It's not been so simple (or cheap) in our experience.

Firstly, with the software we need (to cut hybrid discs) your idea means that we would have to have at least a further 650 meg on our hard disk (in addition to the 650 you propose) every time we would want to cut a new alpha or beta.

Secondly, you would have to give the junk similar names to the real programs and mix them right in with the real programs, or else any copier will immediately see which are the programs to copy and which are not. If you mix in junk, however, it causes you endless headaches updating and backing up, and even worse, it is more than likely to slow down your CD at final run-time.

Moreover, if you do have a really desirable program that's going to bring pirates good money, then a full CD won't stop them. It might stop the casual home user, but not the professional — and those are the people we really want to stop.

I still think Laserlock is the best protection around at the moment. It's not particularly expensive either.

Keeping the price of the CD low, as George Hallak says, is also a good policy, but not on its own (without protection) unless you're prepared to sell for next to nothing, because if you have a killer product that will sell in large numbers, the profit for the pirates will still be attractive, even if they sell it for 10 dollars. I heard the other day of a CD for sale in Saudi apparently for as little as 15 Riyals and it contained Sakhr's Al-Qari' al-Ali, plus a whole bunch of other Sakhr products and some of our earlier KC programs (before we started protecting them). The Al-Qari' al-Ali is apparently fully functional! As far as KC is concerned, we're looking on it is a compliment and some free advertising for our later products!

A bas les piratos!

Adrian Brockett

That sounds like the best idea I've heard so far in this forum concerning this issue.

The other thing that the entire US software industry seems to have opted for is to make your application difficult enough to use that it requires a complete copy of the manual in order to use it effectively. If they copy the software they still can't use it until they get a good copy of the manual.

Printing pirate manuals is always a lot easier to trace (or so it seems) than furtive diskette/CD-ROM copying.

Piracy is still an issue in the US. When everybody ended up with a hard drive, in the mid-80s, it seemed like the pirates were going to win, and us, programmers, were going to have to write our software as a free public service from now on. Here are some of the techniques in use that make piracy more difficult:

1) put a hardware device on the printer/serial/mouse port with an ID string at a specific hardware location addressable on that device. If the device is not attached to the proper port, with the proper ID byte, with the correct value in that byte, the software prints out a nasty message and refuses to run;

2) embed a user's ID function into the program. If the user has not typed in his user's ID that was printed on the packaging the software won't run;

3) no registered user's ID, no technical support. PERIOD;

4) make user's pay for technical support. They need proof of purchase and a user's ID string to apply for technical support;

5) educate folks on the dangers of SW viruses and teach them that part of practicing "safe hard drive data" is to only use software that has been properly purchased (or only pirate software which you have seen come out of the shrink wrap);

6) encourage your (fellow) employees to only use properly purchased software;

7) encourage your local law enforcement folks to enforce the copyright laws;

8) don't use pirated software.

If you want more specific advice than that or me to actually put some real time and energy into it, well ..... what I can I say that doesn't sound horribly mercenary.

Andrew Freeman

1. You mean a dongle. I don't know, but they seem pretty useless. There are companies out there, whose sole purpose is selling software that crack dongles. Their ads are right next to the dongle guys like Everlock, Rainbow, etc. You won't believe how simple this can be. There was once the famous one byte crack for AutoCad's 3D Studio. Many companies abandoned them because they caused too many problems.

2. You mean something like a serial number. The same idea. Copies are distributed with the original serial number in a text file. There are even serial number generators that spit out dozens of serial numbers that work.

3. Hackers have their own networks of support.

4. The same as above. Plus you can post questions in Usenet, Compuserve, etc., and you will get support from fellow users and teach support. Nobody asks you if you're a legitimate user.

5. Hackers are usually technically very versatile. They know more about viruses than we do. In fact, they are the ones that create them. With virus checkers, users would feel safe.

6. Illegal copies are everywhere.

7. They are too busy fighting violent crimes. Plus it's too difficult to get a warrant to search a place.

8. Don't use pirated software.

Abdel-hadi

How can a pirate crack the "dongle" as you call it, without being forced to manufacture a duplicate device?

My point on piracy with regards to self-discipline and encouraging self-discipline in my peers is, I think, the solution which has born the most fruit in the US. I don't have any pirated software on my hard disk, I currently don't know anybody else with pirated software on their disk... The last three places I have worked, using pirated software was reason for dismissal.

These are just thoughts. Just another though, pirated software cannot thrive without an environment which tolerates it.

Andrew Freeman

You change the program so that the program doesn't use the dongle anymore, rendering it useless.

I don't know anyone who does NOT use any kind of pirated software. Whether they are copies of some commercial software or unregistered shareware.

Abdel-hadi

In the UK, software theft is seen as no different from any other sort of theft: people can go to prison for using pirated software.

Three or four years ago, a member of staff at a UK Higher Education establishment was sentenced to (if I remember correctly) 6 months in prison for doing so. Since then, not one of my colleagues has been found to use pirated software. Even the real beginners in computing know that it is simply not an option. Students are routinely warned that software theft will lead to expulsion from the university.

Quite apart from legal punishments, it is in the interest of universities themselves to be like Caesar's wife: not only innocent, but beyond all possible suspicion. If suspicion ever arises we will lose all the excellent deals which allow us to use legal software at educational discount prices.

Alec McAllister

Pirates in the Middle East are the ones who sell your work. If there are fools who are willing to use up 650 MB of their hard disk, to get a recipe, they are welcome to it...

We did use "garbage files" as a filler with our 3 CD titles and lowered the retail price...

There are so many good tricks that I learnt from posting my question. Utilizing Adobe's Acrobat technology in the upcoming release, (sorry guys, it is called "Attabkh El-Arabi") we are able to, and are looking at what Adobe has to offer. In Adobe's downloadable 3.0 release, there is a section about Encryption and such, that our technical guys are evaluating, as we speak.

Thanks to all the people who contributed... Please, carry on with the debate...

Cheers,

George N. Hallak

Hard drive space is cheap these days. A 2.1G drive costs $300. So 650 Megs is less than $100. Plus you can reuse that space.

The main drawback of my idea is that the CD might be slow searching for the correct file. Are you loading many files? Updating and doing other maintenance is not a big problem. You're using an installer.. right? Let it do the hard work for you.

Tell me exactly, who are you targeting? The home user, who has a limited hard drive space and who will think that copying a 500 Meg+ is not worth it, or the professional hacker, who will crack your software no matter what you use. There's no bulletproof copy protection. I can personally give your CD to teenage crackers who will crack it just for the fun. If you look around, you will probably find crackers who can handle Laserlock. What does it do exactly? Fingerprints the CD?

Does Laserlock require that the CD is in the drive? Your CD has a volume name. Your program reads that volume to verify that the correct CD is in the drive. Try to write to it and check to see if it has been written the way you intended to. If it hasn't, you have a CD. You can't write to a CD.

Abdel-hadi

Dear Abdel-hadi,

I'll take up your challenge! Send me your address and I'll send you a CD and please send me back a copy as soon as your hackers have got one working ... 1998?

If by fingerprint you mean some special unique identifier, then no. They call their disfigurement of the disc a "watermark". It is in non-standard form, so the disc cannot be copied in its entirety on to another disc.

Laserlock DLLs work on a lower level than volume names.

I'm beginning to sound like a Laserlock salesman! Forgive me, I just want to make sure I'm getting what I've paid for.

Al-Mutanabbi was said to have been an inveterate copier and so was Beethoven, apparently, so it's nothing new.

Adrian Brockett

I am sure that the major factor why Arabic (and other language) desktop publishing software has not kept pace with that of English language software is primarily due to the lack of enforcement in these countries.

Adapting USA and International Copyright Laws, honoring Licensing Agreements are the first steps to curbing this piracy... Without a "bite" in the laws, who will take software developer rights seriously?

Countries that do not adapt these laws should face actions from individual (host software) countries and international community. By protecting the rights of the developer, EVERYONE will benefit with less costly and more abundant software options...

Mark

U.A.E, Saudi Arabia and Egypt have cracked down software pirates. Closed down some shops and probably prosecuted them. So they are in the right tracks. Although there will always be underground activity, for the most part they can't do what they used to do in public.

No more computers sold and load with pirated software. No more selling copies publicly.

Abdel-hadi

Companies develop software to make profit. With profit, comes the part where a big company has an obligation towards the part of the world community it is serving.

People of the Middle East and other countries, like in South America, they brag, publicly, about pirating an expensive software. In Beirut, the Microsoft representative is selling both versions (pirated and original) of Microsoft products on shelves next to each other!

It is not only the responsibility of the Lebanese government, alone, nor the honorable volunteering of not using Pirated software; it is also the responsibility of companies like Microsoft and others to help those talented pirates to their side, by subsidizing products going to an area of the world, where the per capita income is far less than the originating country of the software. The average PC user in USA pays $39.99 for Windows 95 (after the rebate, currently at Staples). While the Arabic user will have to shed $174.00 (?). What sort of logic is that?

The law is on the books, the resources to enforce it are not there. It may not be practically feasible for big Companies to subsidize such ventures. That what is called, "lack of vision"... If there is not an instant profit, they will not do it, such companies have no one to blame but themselves.

On the other hand, piracy can break a small developer like Arabization and Software Center. I hope the above will not be misconstrued as "pro- pirate", on the contrary; and please, remember who started this thread, in the first place.

Virtually,

George N. Hallak

I didn't know it's $39.99... $50.00 rebate? They can afford to go that low because they sell millions of it. They sell a few hundreds or a few thousands of the Arabic version. I am not sure if even Microsoft makes any money out of the Arabic version.

Abdel-hadi

Speaking as a practicing Software Engineer, I have three observations to make:

1) developing software is hard work, requiring an enormous amount of training, skill, perseverance, effort, blood, sweat, tears, knowledge, patience, connections in the industry, and some amount of luck;

2) I don't think that there really is any method of keeping software out of the hands of a dedicated "Piracy Industry". Anything that a developer can do to make it difficult to copy, a "cracker" can unravel given enough time and resources, not to mention the occasional disgruntled employee from the target company. A mega-giant like Microsoft can take the hit, by jacking up their prices, a first time start-up company will not survive under these circumstances, unless they can somehow buy off the pirates or sell enough copies before the pirates crack the protection scheme to cover most of their expenses to keep the investors happy or... well, I can't imagine what;

3) most of the software which I have written has ended up on a ROM, so, really, piracy was not much of an issue, although, occasionally for certain markets, we would do things like scramble the data pins going into the ROM. Keeping track of that kind of stuff just made the task of producing quality software that much more difficult.

I guess what I'm aiming at is that the current situation does not make it very inviting for small-time investors and developers to set up shop in the Middle East. I even would want to argue that it is helping contribute to the "brain-drain". I have lost any thought of wanting to live and work in the Middle East supporting myself as a software engineer since coming to a full understanding (in the last week) of the nature of the SW piracy industry in the Middle East. I guess I will now be forced to support myself on Fullbright grants and do any linguistic studies mostly as an outsider.

There is no easy solution, but this here "piracy industry" is not helping things in the long run.

Yours,
Andrew Freeman

That is not censorship... It is a much needed protection for the developers. ASC stopped using dongles because customers did not like them, and can cost to the software.

George N. Hallak

Hello fellow netizens,

I sort of dread re-opening this particular thread but the idea came to me in the shower, and it has been stewing in my head now for about a week. So, here it is.

The latest and humblest software protection scheme:

1) the software is only available over the WEB;

2) the software does not come complete ready to run, the user has to dial into a BBS/WEB/INTERNET site and use his license ID number to download a small piece in order to activate the software. Only the first one to call in with that ID number gets activated, every other caller is simply "Shoot! Out of Luck" hereinafter referred to as SOL;

3) the application will only run from the hard-drive, and keeps a few counters laying around which keep track of how many times each menu option has been accessed. The application writes itself and these variables cleverly imbedded in cryptic assembly language code back out to disk every time it runs. These data structures are CRC, or some other integrity checking scheme, tagged for evidence of tampering. Also stored is the track and cylinder information for where on the disk the application has been stored;

4) at load time the application figures out where it is on the disk, if this does not match the stored information, it writes zeroes over the image stored on disk. It also attempts to find any other copies of the application on any other disk in the system and tries to write zeroes over them as well;

5) the user needs to dial in about once a month to get his menu counters reset to whatever we decide is reasonable. When any menu counter gets decremented down to zero the software refuses to run;

6) whenever the software writes itself back out to disk it also writes out the date encrypted. If the boot date is earlier than the stored date the software refuses to run;

7) to discourage folks from tampering with this scheme, some the values written out to disk are an encrypted CRC of the entire binary image. If the stored CRC doesn't match the calculated CRC the software refuses to run;

8) the first thing the software does, when it loads itself, before it responds to any keyboard input, it trashes all the menu counters, dates and CRCs which are stored in the disk image. This is to discourage folks from rebooting the machine rather than exiting by saying "quit", in a nice orderly fashion. This basically means that, if the machine crashes while the software is running, the user needs to dial into the vendor with his license/ID number and repair his disk image;

9) at regular intervals the software will force the user to re-download a significant piece of the application, which of course has had features added and been re-linked in a different object file order;

10) if the time elapsed since the last dial-up refresh of menu entry counters is more than 30 days the software will disable all menu options except for the "Call Home and Update Software" option;

11) one of the pieces of data stored, when it writes itself back to disk, is the old CRC for the entire image — if the new CRC ever matches the old CRC it refuses to run;

12) sort of related to #9, if the last run date is more than 30 days older than the current machine date the software will disable all menu options except for the "Call Home and Update Software" option.

I think there might be hole in the system having to do with saving the first-time image and always using that one, but then you need to write a utility to make sure it gets stored in the same location every time, which I foresee as a serious headache on any busy system, and you need to run with a bogus date always.

If this really becomes a problem, then

13) The software needs to call into the vendor every time that it runs, if anything has been tampered with the software refuses to run, i. e. menu counters are not being updated. If the software is being used more than 480 hours in a month, it gets shut off.

I think this is basically an elaborate but workable way to establish control over the application's executable image. I hate it, because it means taking a chance with writing non-portable code, tailored to the different disk controllers which can exist on whatever platform your code runs on. It boils down to at least one hardware specific disk controller object code library for each disk controller for each operating system that doesn't directly supply disk location data. My experience has been that, in the long run, non-portable code is non-professional code.

My guess is that since Norton's Utilities can do what it does for any disk out there for both the MAC machines and the PC/DOS machines, it only means writing one operating system specific library per operating system and not a proliferation of one-library-per-Disk-Controller/per operating system horde of libraries.

One big drawback is that in places like Cairo where the average consumer can't make long-distance phone calls from his/her flat, you have to set up a local distributor with a BBS, who, of necessity, will have all the control over the local licensing and updating schemes. This is a potential leak in the system.

The other option would be to route a private T1 line out of the country through an International trunk, which I'm not sure is possible or legal or economically feasible ... I am giving up this idea into the public domain as a public service.

Anyway, it's obviously a lot of work, but I would be interested in hearing:

a) if anyone has tried anything like this;

b) if anybody can think of a way around it;

c) if anybody is going to try it.

Andrew Freeman

Humblest software protection scheme! It's an overwhelming way, and people will hate it.

So most users (also your customers) are out. This holds well in the Middle East where few have WEB access.

A disk defragmentor will render your software useless, because it will change the software physical location unless you make it unmovable. With all your schemes you mentioned. something is bound to get wrong and by Murphy's law, it will. Too much headache.

Abdel-hadi
