Dr. R. Beezer Math 133
Practicum #8 Fall 2003
Timestamping with Stamper

Sometimes *when* you say something, or *when* you heard it is just as important (or more so) than *what* you said or *what* you heard.

In this practicum we will practice adding undeniable digital "timestamps" onto our communications, using "Stamper," a free service on the Internet.

Procedure

1. Links to the Stamper WWW site are on the course page, and various pages there will be referenced through these instructions.

2. At `http://www.itconsult.co.uk/stamper/stampinf.htm` (follow the link listed as "Instructions for Stamper service") you will find the public key for the Stamper service (about two-thirds the way down). Much of Stamper works by signing messages (with its private key) so you will need this key on your keyring. Add it to your key ring by doing a cut/paste job into Notepad, saving the new file (with a `.asc` suffix), and then sucking it into PGPkeys for addition to your key ring.

3. Send me a time-stamped message as follows. On the instruction page listed above, you will find instructions for "Using the service in post mode." Read these, and use them to send me some pithy quote. At the end of this section of instructions is a link to an "example of use." This should prove helpful also as you construct your message.

   So you will be constructing a new message, addressed to `post@stamper.itconsult.co.uk`, with a first line in the body of the message (starting in column 1) that says `X-Stamper-To: beezer@ups.edu`. This will be followed by the content of the message.

   Stamper will

   (a) Sign your message (the most important part).
   (b) Give the message a sequential serial number.
   (c) Note the time of day (GMT = Greenwich Mean Time, roughly the time in England, about 6 hours ahead of us).
   (d) Forward the message on to me with the time and sequential serial number.
   (e) Send back a copy to you with the same time and the same sequential serial number.
   (f) Retain a copy of the signature it just made, but not the contents of the message (a "detached" signature).
   (g) Add this detached signature to a daily list, in serial number order.
   (h) Wait until midnight, sign the list of all the day's signatures and post the list (plus the daily signature) on its WWW site (more on this later).
   (i) Wait until the end of the week, form a list of all seven daily signatures, sign it, and post it to a USENET newsgroup for duplication around the world.

Whew! We'll see in a bit what this is all about. When you get a message back, you'll know Stamper has sent a message to me, with the time, serial number and contents of your message (the pithy quote).

4. Do we trust Matthew Richardson, the "owner" of Stamper? Of course not! He seems like a nice chap, and he's been at this for a long time, but I don't trust him. Maybe for the right amount of money, he'd jigger Stamper and let me post-date an old message, or pre-date another. Everybody has their price. And why would I send a message through his computer anyway? Maybe he's saving all of these critical messages, like your pithy quote you sent me, to use later for blackmail, or to send on to one of our enemies. Or maybe he'd just have fun embarrassing us by posting our private communications. What's even funnier, I'm sure Matthew Richardson wouldn't be offended by anything I just said. He knows we shouldn't trust him, too. So what good is a timestamp from Stamper if we are so distrustful and paranoid? Read on.

5. I am going to send you another piece of classic literature. Only I don't want Matthew Richardson to read it. So I'll encrypt it with your public key, so its for your eyes only. I'll probably sign it too, but this is optional. I also want it timestamped by Stamper though, to validate just what time I sent it. So I'll be sending the signed, encrypted form to Stamper at `text@stamper.itconsult.co.uk`. Matthew can't read it (since it was encrypted for only you to see). Stamper will send it back to me with a signature from the Stamper service, and the serial number (but not the timestamp). Stamper will send it back to *me* since I'm not sure I even want Matthew to know that I was talking to you, so I'm not going to tell him who the message is meant for (so Stamper can't know where to send it on to). Even better, the message I get back does not have the actual timestamp on it, so you are going to have to figure that out.

6. I now have this file, encrypted to you, signed by me, signed by Stamper, with its assigned serial number. I will now email this to you. What do you do with it?

7. With a cut and paste job, save the file with a `.asc` suffix (say as `email.asc`).

8. Double-click on the file you just saved (`email.asc`), and you will be prompted to save it as file with a `.msg` suffix. Change the name entirely, using a `.asc` suffix, (say as `converted.asc`). Remember the name of this file, you will need it later.

9. Double-click on the file from the previous step (`converted.asc`). This is the decryption step, so you will provide your passphrase now. Save the output as a file with a suffix of `.txt` (say as `message.txt`). Open this file with Notepad to view the passage I sent you. You should also be verifying my signature in this step so you know one of your classmates isn't fooling with you.

10. Now we want to verify Stamper's signature (you now have Stamper's public key on your keyring from an earlier step, remember?) to make certain that the sender (or Mallet) hasn't tried to impersonate Stamper. Now that you know the message, and its sender, are legitimate, note the serial number.

11. Go back to Stamper's WWW pages and follow the link for "Signature & Summary Files." Go to "Signatures by date" for this year. Use the list of final serial numbers for each day of the year to determine what day I had Stamper sign the message you have received.

12. Back up and then choose the link "Detached Signatures by Date." Find the correct file for the date of the message I sent you, the filename format is `YYYYMMDD.txt`, listing the year, then the month, then the day. Save this file (using a right-click and a "Save Target As...", or a cut/paste job with Notepad). For a filename, be sure to change the suffix to `.asc`, so the filename will have the format `YYYYMMDD.asc` now.

13. Double-click on the file you just created. The following is known to work with version 7.0.3. If you have chosen to disregard previous instructions about which version to install, you may be on your own now. You will be asked to save the file with a "screwy" looking filename. Go ahead and accept that name, but you may want to write it down.

14. Now double-click the file with the screwy name. It will again prompt you for a name, it will look like `YYYYMMDD.zip`. Be sure to go ahead and use this name.

15. You now have decoded a zip archive of all the detached signatures for the day in question. Double-click on the `YYYYMMDD.zip` archive, to unpack the archive (this will work in Th 120 if you don't have WinZip installed on your personal machine). Locate the signature that corresponds to the serial number for the message I sent you, and extract it, making sure you know where it will land. It will be named with the serial number and a suffix of `.sig`.

    Within the zip archive, notice the time the signature was created, and notice the times of the signatures just before and just after yours. Once we finish verifying everything, these two "bracketing" messages will provide some confidence about the time the message you received was actually signed.

16. Finally, we can compare the detached signature file we got from the Stamper WWW site with the signature on the file I gave you. Double-click on the `.sig` file. When it asks for the "signed file" give it the file I asked you to remember above (the one provided as input for the decryption step, `converted.asc`). You should get a message or log entry that says the signature is good.

17. So what have we accomplished? It will be very hard for Matthew Richardson (or anybody else) to adjust any of the summaries or logs, since they are all interdependent and have been signed in so many ways and made publicly available. And it would be hard for an adversary (or you yourself) to adjust anything after the fact. It could be that Stamper imposes significant delays in actually doing the timestamps, but if it were that unreliable, no one would continue to use it. So we know roughly (within a few minutes) when the message was signed by Stamper.

18. Hopefully, the signature file we got from Stamper's logs matches the signature embedded in the email you got. Since signatures involve hashes and private keys, it would

be near impossible to forge this matchup. We have reasonable confidence about when this signature was created, since it is buried down inside Stamper's logs, which have several layers of signatures around them and are therefore difficult to jiggle.

19. So we can get a semi-authorative timestamp on our electronic communications, free-of-charge, all through the properties of public-key encryption and digital signatures. Wow!

Grading

Full credit once I receive

1. A pithy quote from you, delivered with a timestamp via the message posting sevice of Stamper.
2. The author of the passage I sent you.
3. The time, to the minute, when my message to you was timestamped, plus the times of the messages just before yours, and just after yours.

Notes

1. We have learned not to trust the alleged identity of authors of electronic communications without digital signatures. We should now not trust the time given in electronic messages.
2. This exercise should further convince you of the utility of digital signatures for establishing trust in electronic communications.
3. From now on, all of your email to me that requires it be submitted by a deadline (discussion groups, subsequent practicums) must contain a timestamp prior to the deadline (use the posting service as in the first part of this exercise when you sent me a timestamped message), since I will no longer trust the times on your emails that you have provided personally.
4. If you ever have a secret, and you would like to prove to someone that you knew it at a certain time, you could write it down, encrypt it, have Stamper sign it and return it to you. Then you can reveal it at a future time of your own choice. For example, you might be an influential Wall Street analyst who wants to predict the direction of the stock market, but you are afraid people will accuse you of influencing the market in the same direction by making a public prediction. Make the prediction (encrypted, of course), have Stamper sign it, then post a copy on your Web page. Later, verify Stamper's timestamp (proving when you made the pediction), and decrypt the message with your private key, revealing the prediction (which hopefully was accurate).