[Underground Security Paper no. 2]

Encrypting Email Communiques

v 1.0

By: DIzzIE [antikopyright 2007]

This is the second Underground Security Paper designed to further empower you to give yourself some semblance of electronic privacy.

If you haven't done so, go read over USP no. 1: Encrypting your Instant Messaging Conversations: http://forum.rorta.net/showthread.php?t=576

And yes, I'm well aware that there are a few other guides on encrypting emails already out there, but they are either outdated or don't mention all of the shit that I want to mention. So here we go, first we'll go over a few ways to encrypt emails if you have steady access to the same computer (i.e. your laptop or home desktop), and then I'll mention some options for encrypting emails from public terminals using free webmail providers.

Nota Bene: If you're a little fuzzy about this whole 'public key cryptography' thingamajig (don't worry, so am I), it may behoove you to take a quick gander at the respective Wikipedia article on the subject

http://en.wikipedia.org/wiki/Public-key_cryptography

so that shit like 'public/private keypair' will make slightly more sense as you read this guide.

In a nutshell, what you'll be doing is generating a set of two keys, or a key pair, one public and one private. The public key you make public (duh) by giving it out to all of your contacts, posting it on your website, and so on. The private key you keep--you guessed it--private, and protected with a strong and salted passphrase (we'll get to that later in the text). The sender of the email encrypts the email being sent to you using your public key, and only you can then decrypt the email using your private key. Likewise you use your contact's public key to encrypt the emails you send to zir, and z must use zir private key to decrypt the emails that you send to zir. Now then, let's explore your various encryption options...

Option I: Encrypting Emails on a Stationary Computer Part One -- Thunderbird & GnuPG

If you have steady access to a computer on which you can install software, this is the option to use. It offers much more flexibility and security than any of the other options in that it is not tied to any specific email provider or any specific operating system; furthermore, your key management is done locally, not on any sketchy third-party server. If you have some of the tools discussed herein already installed (Thunderbird, GnuPG, Enigmail), you can of course skip over the steps that tell you to install them ;).

1. Install Mozilla Thunderbird (http://www.mozilla.com/thunderbird/), which is a free, open-source, multi-platform email client that you'll be using to fetch, read, and

send your emails in lieu of whatever client, web-based or otherwise, that you currently may be using.

2. Configure Thunderbird to fetch emails with your existing email account. Open Thunderbird, go to File>New>Account, select Email Account, and follow the Account Wizard to completion. You'll need the address of your mail provider's POP/IMAP server to retrieve emails and the address of an SMTP server to send emails. Search through all of the readme files and help documentation of your email provider to find the necessary server addresses to put into Thunderbird.

   If you're using certain crippled webmail accounts like Yahoo or Hotmail, you'll need to run a third party program like YPOPs! for Yahoo Mail (http://ypopsemail.com/) or the Thunderbird WebMail extension (http://webmail.mozdev.org/) for Yahoo, Hotmail, and so on (there are various other options such as FreePOPS (http://www.freepops.org/) floating around as well; try them all out and pick your favourite). Both of these programs have detailed installation/setup instructions on their websites, so I won't bother repeating them here. If you have a Gmail account, instructions for configuring Thunderbird are here (http://mail.google.com/support/bin/a...y?answer=38343).

   If this seems like too much of a headache, you may find it easier to try the browser plugins mentioned below in Option II (though they may be significantly less secure, thus I strongly recommend you stick with Option I).

   Nota Bene: Be sure to give Thunderbird and any of the other related programs you install the ability to access the Internet in your firewall and/or setup your router to forward all necessary ports (default POP3 TCP port is 110, Secure POP3 (POP3S) is 995, and default SMTP port is 25, though your mileage may vary).

3. Install GnuPG (http://www.gnupg.org/download/). This is the free, open source encryption suite that will provide the backbone for encrypting/decrypting your emails. You can either grab the source code and compile it yourself, or download a precompiled binary for your OS (for instance, Windows users will probably go with ftp://ftp.gnupg.org/gcrypt/binary/gn...2cli-1.4.7.exe - the latest compiled binary for Windows at the time of this writing).

4. Install Enigmail (http://enigmail.mozdev.org/download.html). Enigmail is an extension for Thunderbird that will allow you to easily use GnuPG to encrypt/decrypt your emails within Thunderbird. (If you have *.xpi files set to usually open with Firefox, save the xpi file, open Thunderbird, go to Tools>Add Ons>Install, and go to the directory where you saved the Enigmail xpi file.) Be sure to install GnuPG prior to installing Enigmail.

5. You'll now need to put the finishing touches on Enigmail by generating a key-pair and tweaking a few settings (don't worry, you're almost done :)). Reopen Thunderbird and you should now see an OpenPGP menu at the top. Click on OpenPGP>Key Management and the OpenPGP Wizard dialogue should pop up.

   a. Hit 'No' to exit the wizard (we'll be generating a key-pair that is stronger than the one generated automatically by the wizard, so we'll have to do this manually).

b. In the OpenPGP Key Management window click on Generate>New Key Pair. Select the Account you want to generate the key-pair for in the drop-down dialogue, and leave 'Use generated key for the selected identity' selected.

c. This step is **the most important step** as it creates the passphrase for accessing your private key. Aside from common sense rules like making sure your passphrase is unique (i.e. not something you also use to sign into your instant messengers or your webmail or forum accounts) and contains mixed-characters including capitals, numbers, and other <|-|/-\r@(73r5, you should also make sure that your passphrase is of sufficient length (30+ characters is a good start, or in other words the bare minimum).

Furthermore, be sure to sufficiently salt your passphrase (if you picked a particular quote that you like, don't put in the quote verbatim but mix up the spelling a bit so as to deviate from the standard accepted spelling by adding a few random characters here and there and so on. For instance, instead of mysecretpassphrase try ||\/||ysss3|<r377p@55frraeyz).

Devote ten minutes a day for a week to commit your passphrase to memory, and then destroy any existent recorded evidence of the passphrase. Rehearse your passphrase periodically in your head so that you don't forget it. If you nonetheless fear that you will forget the passphrase store a copy of it in a clandestine location away from your home, being sure that it has no identifiable markings that would allow it to be traced back to you should it be found at a later date by an undesirable third party.

d. In the 'Key expiry' area below the Passphrase field, set the duration you want your key to last. Remember that while 5 years (the default) may seem like a lot of time, when your key expires you'll have to go through the trouble of redistributing your public key to everyone all over again. On the other hand, if you are communicating with a small number of individuals to whom you can regularly give out public keys, it would behoove all parties involved to generate new keys on a regular monthly or even weekly basis.

In other words, change your keys often for added security, but also weigh the hassle involved in redistributing your keys and picking a new passphrase for each key.

e. Click the Advanced tab and change the key size from the default 2048 to 4096 bits. Change key type from DSA & El Gamal to RSA. You can find a variety of information online espousing the virtues of RSA versus DSA and vice versa. I prefer RSA due to the simple fact that DSA was designed by a then-NSA employee, while RSA was designed by three professors. Though academia is of course deeply intertwined with government interests, RSA may still be the so-called lesser of two evils due to that one notch of separation from the government. At any rate, irrespective of whatever encryption algorithm you select, **be sure to bump the key size up to 4096 bits**. Recent news (http://arstechnica.com/news.ars/post...4-bit-rsa.html) shows that 1024 bit keys are ever-closer to being cracked.

f. Now open a movie or two in a video player (re: the little message saying 'actively browsing or performing disk-intensive operations during key generation will replenish the 'randomness pool' and speed-up the process'), and then hit Generate key and wait for your key-pair to generate.

g. After the key pair has successfully generated you will be asked if you want to create a revocation certificate, hit Yes and save the certificate to an external medium (**not** your hard drive) that you can then store in a secure remote location (**not** your home and **not** where you're keeping your passphrase). **If you have doubts about your ability to securely and remotely store the revocation certificate, do not create one**. Remember that if anyone gets a hold of your certificate they can then make your key invalid, forcing you to have to explain to your contacts why ?you? are suddenly using a new key, which will in turn cast a shadow of doubt over your supposed identity.

h. Exit out of the Key Generation window to get back to the Key Management window and, right-clicking on your newly generated key, select Export Keys to File. **Be absolutely sure to click 'No' in response to the question that pops up asking you if you want to include your secret key**. Open the resulting .asc file in a text editor and double check to make sure that only our public key is included. You can now send this public key to all of your contacts, as well as posting it on your own website and/or on that of a public key server (in case of the latter, right click on the key again and select 'Upload Public Keys to Keyserver', but I strongly advise against doing so).

   Nota Bene: I don't particularly recommend uploading your key to a public keyserver as I don't like the idea of even my public key bouncing around on some third-party server, **not to mention that it may be possible to determine relationships between people by comparing all signatures tied to a key by performing a Verbose Index search for keys on a given key server. This is a significant blow to your privacy.**

i. Exit out of the Key Management window and, clicking on OpenPGP again, this time go to Preferences and check 'Display expert settings'). In the Sending tab, be sure to check 'Add my own key to the recipients list', 'Re-wrap signed HTML text before sending' and 'Always trust people's keys.' The other options are optional, and you can mouse over them to get a little more information about each one. Alternatively, more information on Enigmail configuration is available here: http://enigmail.mozdev.org/configure.html. Come to think of it, the first three options are also entirely optional, but will make Enigmail/Thunderbird run a little more smoothly.

j. Under the Advanced tab, be sure to **uncheck** 'Add Enigmail comment in OpenPGP signature' as this comment field tends to sometimes interfere with successful decryption of the message in Enigmail/Thunderbird when included in the sender's message. The other features are optional. Hit OK to save your modified preferences and get back to the main Thunderbird window.

6. Now that you've finally set all the shit up (phew!) it's time to take it for a test run by sending yourself an encrypted email.

a. Hit the Write icon (or press Ctrl-N) to open a new email window, and enter your email address in the To field. Enter a sample message and press the little triangle next to the OpenPGP icon to select 'Encrypt Message' (or press Ctrl-Shift-P) (Sign Message should already be checked, though do check it if it isn't).

b. Hit Send (or press Ctrl-Return), enter your passphrase, and hit OK.

c. In the main Thunderbird window, click on Get Mail (or press Ctrl-Shift-T) and locate the message that you just sent yourself (sort the emails by date or by sender or search for the subject to make it easier to find).

d. Double click on the message and if you have 'Automatically Decrypt/Verify Messages' selected under the OpenPGP menu, you should now be prompted for your passphrase. Otherwise, select the email and hit the Decrypt button, and enter your passphrase. You should now see your decrypted message in plaintext.

7. Now let's try sending someone else an encrypted email. Get a contact to generate a key pair and obtain zir public key. Z can either send you the public key manually, upload it to a website, or upload it to one or more of the public keyservers (see end of Step 5.h -- uploading to public key servers is not advised). Yet another alternative would be for your contact to send zir public key as an email attachment to you along with the initial message so that you will be able to encrypt your response.

   a. In case your contact uploaded the key to a keyserver (which if you recall, may not be a good idea) go to OpenPGP>Key Management>Key Server>Search for Keys>select the Keyserver to which your contact uploaded zir key, and enter the contact's Key ID prefaced by 0x (for example 0xSC4TL0V3 wherein SC4TL0V3 is the Key ID, which your contact can find by going to OpenPGP>Key Management). Hit OK and then OK again once the public key has been found to import it into your keyring.

   b. If, on the other hand, you're being smart and safe and importing the public key manually, go to OpenPGP>KeyManagement>File>Import Keys from File>and find the key. If the key has a .asc extension, you should be able to find it using the default file type 'GnuPG Files'; however, if you saved the file as .txt or what have you, be sure to select 'All files' in the 'files of type' drop-down area or you won't be able to see the key file. Once you've found your contact's public key, hit open and OK to import the public key into your keyring.

   c. Now that you have your contact's public key imported, go ahead and send zir an encrypted email. Repeat Step 6.a to compose your test message, but this time when you hit Send, select the Recipient(s) for Reception in the window that pops up by placing a check next to your contact's key and hitting OK. Put in your passphrase and wait for Thunderbird to say 'message successfully delivered.'

d. Your contact should now check zir inbox to find your encrypted message and then click the Decrypt button, input the passphrase to zir private key, and then successfully view the email you just sent. Now get your contact to send you an email encrypted with your public key so that you can practice decrypting it using your private key. If you have no immediate contacts to test encryption with, make another email account and send encrypted emails between your two accounts.

Nota Bene: Encrypted email is also a great way to send encrypted file attachments. When composing your message, click the Attach button and select your file(s), then when clicking Send simply select 'encrypt each attachment separately', and your entire attachment will now be encrypted (and can be decrypted by selecting the attachment, right-clicking and selecting either 'Decrypt and Open' or 'Decrypt and Save As'). Though do bear in mind that the name of your attachment will not be encrypted, so My_Sisters_Snuff_Reels.avi may not be the best idea for a filename ;).

And there you have it! You should now be able to send and receive encrypted emails and everything that entails (generate strong key pairs, import/export keys, generate revocation certificates, and so on and so forth).

Option II: Encrypting Emails on a Stationary Computer Part Two -- Web/Broswer-Based Options

In case Option I seems way too overwhelming or you just can't seem to get one of the necessary add-ons to work, there are a couple other web/browser-based options that you can employ to encrypt your email. **I neither trust nor recommend any of them** (though I haven't tried them out either), and am only listing them here in case you need to send encrypted email urgently, and don't have the time to go through the elaborate setup of Option I (or you can't get I to work). Though it will probably take you just as long to setup these options (and get your recipient to do the same) as it would for you to setup Thunderbird/GnuPG in Option I in the first place. Thus, if you can't get Option I to work, you're better off trying the webmail options presented in Option III below.

~ **Freenigma** (http://www.freenigma.com/) is a Firefox extension that integrates into popular web-based email options like Hotmail and then allows you to generate your keypair/encrypt your email within the browser using the ordinary web-based Hotmail/Yahoo/whatever page. Freenigma currently doesn't work with anyone who doesn't also have a Freenigma account and doesn't encrypt attachments. All key-management is furthermore done server-side which means you apparently can't import/generate keys on your own. You can find further setup information here: http://www.simplehelp.net/2006/08/26...ing-freenigma/. Not recommended.

~ **Gmail Encrypt** (http://www.langenhoven.com/code/emai...ailencrypt.php) is a Greasemonkey script that adds encryption functionality to Gmail accounts. Both the sender and the recipient will apparently have to be Gmail users. Again, not recommended unless none of the other options are feasible in your situation.

Option III: Encrypting Emails on Public Terminals (Using Free Webmail Providers)

Setting up Thunderbird/GnuPG is great assuming that you have a computer of your own to set everything up on (or access to a computer that has enough permissions enabled to be able to install software on it). But what to do if you don't? Until you jack a passed out college kid's laptop at the local college library, you can use a couple webmail options that have encryption capabilities.

~ **Hushmail** (http://www.hushmail.com/) provides a free encrypted email service with various limits (for instance you're given only 2 Mb of storage, and are required to log into your account every three weeks or lose the account). The nice thing about Hushmail is that it allows you to export your private/public key pair so that others can send you encrypted emails using, say, Thunderbird (or another webmail option like Mailvault, see below), and that you can use your private key to read encrypted emails using other clients as well. To export your keys, log into your Hushmail account, click on Preferences and then Export Encryption Keys.

Public keys can also be imported by uploading them to Hushmail's own keyserver. Instructions for doing so appear at the bottom of this page, https://www.hushmail.com/help.php?subloc=pgp&l=454, under 'How can a Hushmail user send secure email to a PGP user?' What this means is that you can send encrypted email to contacts who do not have Hushmail accounts but are using encryption with another client (like Thunderbird).

Finally, Hushmail also lets you set a security question/answer in the case that you need to send an encrypted email to someone who has neither a Hushmail account (when you send emails to another Hushmail user the emails are automatically encrypted) nor a PGP key pair. In this case, your intended recipient will have to provide the correct answer to your security question in order to be able to view your email. To set the question/answer, click on Compose, and then go to Message Options.

Keep in mind that at least in one case (US v. Tyler Stumbo - http://static.bakersfield.com/smedia...filiate.25.pdf) the pigs were able to obtain Hushmail email records.

~ **Mailvault** (http://www.mailvault.com) is a service that is similar to Hushmail, and which likewise allows you to import and export keys thus enabling you to send encrypted communiques to those who aren't using Mailvault (and likewise allows folks who don't use Mailvault to send you emails as well). However, one of the disadvantages of Mailvault is that its mail servers seem to be a tad erratic, in that mail sent to Mailvault accounts at times gets bounced back as undeliverable.

~ There are various other pseudo-secure web-based email options out there that you can explore by doing a web search for a query along the lines of 'free encrypted email', though do keep in mind that all of these services are only to be used if, for whatever reason, Option I is not feasible in your situation.

A Few Parting Tips and Reiterations (READ THIS SHIT!)

~ Always generate the largest keys the programs allow you to generate, which is currently 4096 bits. Don't settle for the default 2048 bit key lengths.

~ Don't upload your public keys to public key servers unless absolutely necessary (in other words, never). As mentioned in Step 5.h, when you perform a Verbose Index search for a key ID on a given key server (for instance by going to http://pgp.mit.edu/, entering your target's Key ID and conducting a Verbose Index search), you may then be able to see all the signatories tied to that key and may then be able to deduce who has likely communicated with the owner of the given key, thus being able to map an individual's potential contacts.

~ The first time you send a contact an encrypted email, it would be a good idea to attach a copy of your public key along with the email so that the recipient can likewise send you encrypted email in return (unless of course you have already provided the recipient with your public key at an earlier juncture via another distribution channel).

~ Remember to pick a strong, salted (meaning using non-standard vernacular) passphrase with a minimum length of 30 characters. Commit your passphrase to memory or store it in a remote location with no identifiable information that would allow anyone to trace it back to you). Store your revocation certificate (if you chose to make one) at another location (with the intention being that if the passphrase location is compromised or forgotten, you'll still have the second location as a fail-safe to be able to revoke your now-insecure key pair).

~ **Delete your fucking emails!** (and don't keep any logs). Don't be one of those jackasses that archives all of your emails from the past ten years, especially in an unencrypted format. There's nothing the pigs love more than a nice trough-full of aeon-old potentially incriminating evidence to gorge upon as they plot your untimely demise.

~ And finally, remember that **encryption is not the same thing as anonymity**. When you send/retrieve emails your IP address is recorded by your email provider, not to mention probably your ISP, and (depending on the mail server) often times passed on to the recipient as well (Hushmail and Gmail are two email providers that don't pass along your IP to the recipient, though this doesn't mean that they don't store it on their servers (if the double negative was confusing, it means that they *do* keep IP logs of their own that they'll be all too glad to hand over to either the pigs or anyone who pretends to be a pig and sends them an intimidating letter). Use anonymity tools such as Tor (http://tor.eff.org/) or the portable Xerobank (http://xerobank.com/xB_browser.html) when on a public terminal to help hide your IP address, along with piggybacking on a wifi connection if at all possible (You can find various guides on using Tor online, or expect it to be covered in a future issue of the Underground Security Paper). Do not go to the same place at the same times to send/read emails and be mindful of CCTV surveillance.

As usual, this text has gone on for way longer than I expected, so time to end this shit.

Email me at xcon0 @t. yahoo d.t com (Now that I've written the guide, I can finally say that I will no longer respond to any unencrypted emails. My public key (0xF370BFBF) is available here: http://www.dizzy.ws/x1pub.asc).

Visit www.rorta.net & www.dizzy.ws for more knowledge

Yeah, I don't see why it would hurt to make Enigmail prompt you for your passphrase prior to exporting the key. This might be something you may want to suggest on their Feature Requests forum. (On the other hand, if someone gains access to your computer they can also just navigate via the command line to your gpg directory, type something along the lines of 'gpg --armor --export-secret-key 0xYourKeyID > d:\myusbdrive\yourstolenprivatekey.asc' and they'd have your private key anyway, or they could copy your secring.gpg file, which has all of your private keys as well).

The upside is that they would also need your passphrase to be able to use your private key to decrypt your messages, which is all the more reason to make sure you have a strong passphrase.

If you have serious concerns about your machine being compromised you should store your secret keys on a removable medium and then delete the secret key from Enigmail. Of course, this would mean that you would have to import the key each time you need to decrypt a message and then delete it again when you're done, but at least it wouldn't be sitting on your computer waiting to be compromised. You would also have to find a fairly secure location for your storage media that has the key on it as well, which could quickly become a hassle if you need access to it every day. Remember that if it's easy for you to find it's easy for the pigs or whoever else may be looking for it as well.

Going a step further, if you're worried about your physical area being compromised and that someone will be able to gain access to the physical copy of your private key you can do what I think is known as 'blinking', wherein you take a photo of your room prior to leaving, and upon coming back take a photo from the exact same location. Turn the two photos into an animated GIF and you should be able to spot any minute differences between the conditions of the room before/after you left.

Under the pleasant norms of Parisian life, beneath the veneer of culture and civilisation, one of the bitterest and most sadistic underground wars of modern history was fought out.

Last edited by DIzzIE; 7th November 2007 at 09:44 PM.

Posts: 470 Windows users may find gpg4win a nice alternative to the package suggested by Diz. It comes packaged with some nifty tools that make life easier on Windows.

Also, you might try looking here for a list of free POP3/IMAP email providers instead of trying to configure your crappy yahoo accounts.

http://www.emailaddresses.com

Here's a somewhat recent Wired article which links to the same court affidavit that I briefly mentioned in the guide, and goes on to talk about how Hushmail basically admits to having the capability of sniffing your passphrase/private keys and to actively doing so at the behest of the authorities (in other words:

don't fucking use hushmail, or any other third-party provider which claims to encrypt your emails for you).

Also, a curious service a comrade recently pointed out is www.spammimic.com. You enter a (preferably short) message, and it gets transformed into either a spam look-alike message or a fake PGP-encoded look-alike text. Someone then pastes the encoded text into the decoder on the site and it spits out the plaintext message. Obviously if you're using this for anything besides the novelty value you'll want to use the password feature to make sure that not just anyone can decode the message, though the site still rightly warns that "This is weak encryption - it's better than no encryption at all but not much. Not recommended for secret agents or even corporate spies."

Under the pleasant norms of Parisian life, beneath the veneer of culture and civilisation, one of the bitterest and most sadistic underground wars of modern history was fought out.