SCRIPT for encryption part I

On the demonstration machine (assumed to be different from the normal machine for common use)

1. Set up Thunderbird on andrew not to delete messages from servers, and only get mail when asked to do so. Edit → Account settings → Server settings for the account; then uncheck "Check for new messages at startup", "Check for new messages every N minutes" and "Automatically download new messages"; and then check the box "Leave messages on server".
2. Create a new e-mail address: onlifa.testing@googlemail.com by going to http://gmail.google.com and create an account. Then go to Settings → Forwarding and POP/ IMAP; then Disable forwarding, Enable POP, and Disable IMAP.
3. Create a new user: Onlifa Testing (login: onlifa).
4. Ensure user privileges for onlifa allow connection to networks.
5. Login to onlifa
6. To enable easy switching between users, right click on the user's name and select Preferences, and uncheck Lock screen. This should be done for all users you wish to switch between.
7. Fire up Thunderbird and set it up to download by POP from googlemail, use password protection for login, and set up servers and ports and SSL connections appropriately. Information on how to do this is on the googlemail site under account settings. Note the change required to the port number (995 for server settings) and that SSL must be ticked under both Server settings and Outgoing server. User names are the full e-mail address from Google.
8. Install Enigmail from repository (as an administrator, System → Administration → Synaptic, and quick search for Enigmail, tick the box by the name → Mark for installation, then Mark any additional packages required, and then hit Apply and wait for install to finish), or as a download add-on to Thunderbird (Tools → Add ons → Get extensions and Download it and install). For Linux users, it is strongly recommended only to use the version packaged by the distro in use. Enigmail has versions for Windows and Mac OS/X as well. The statistics for this add-on show it has been downloaded over 1.3 million times, and that number will not include the Linux packaged version in distros.
9. Create keys for onlifa. Go to OpenPGP → Key Management. If it hasn't ben used before, it will present you with a wizard. There are several reason why you may not wish to do this. If you want to avoid it, click Cancel, and go to step 12.
10. If you use the wizard to get you started, click Next → Create per-recipient rules → Put in a pass phrase, and it will generate a key valid for 5 years and adjust all the settings for Thunderbird as requested. It will then ask whether you want a revocation certificate. This is a good idea and you might as well do it now, and read the text in dialog boxes.
11. **Important:** When generating keys, it is essential that the machine can create large amounts of random data, and to help it do this, start new applications, do massive file searches, etc, to create as much disk traffic as possible as that is where it will collect random data from.

12. The default settings create an El Gamal type key and there are some reported issues with older versions of PGP, but not GnuPG, when using these keys. If you do not like this, then create a new key from the Key Management window through Generate → Key pair, and go to the Advanced tab. There you can change the algorithm to RSA. Doing it this way also enables you to set a comment associated with the key, and as you will see later, this comment can be made visible to everyone else. In that way, you can identify which key you will use for what purpose.
13. Create a revocation certification and keep, it safe. Take it out of your machine, lock it away, but do not lose it. You could even print it out although typing in the long string of text would be a real chore put conceivable if you really had to. This certificate provides the only way you can rescind a key after it has gone public. In the Key Management window, highlight the key you wish to create a revocation certificate for, then Generate → Revocation Certificate. You will notice that you can create such a certificate only for those keys you know the secret key for.
14. If your key has **never been used**, then simply deleting it from your own Key Management window will safely erase it.
15. Preparing an identity. Before you can sign or encrypt a message, the identity must be prepared by enabling OpenPGP support. In Thunderbid, go to Edit → Account Settings, highlight the relevant account name, and click on Manage Identities. This gives a list of your current identities. To create a new one with the appropriate settings, hit the Add button and fill in the names etc as normal, but in order that you can recognise them easily it is a good idea to put in brackets after the name an indication of what it is. So in this example, you can set the name to read “Onlifa Testing (signed)” to be used when sending signed messages. Then go to the OpenPGP tab and check Enable OpenPGP support. Then you can set up any defaults that are suitable. Initially, this could be to check “Use e-mail address of this identity” to select the key and only switch to a specific one when more than one exists. Leave everything else unchecked for the moment.
16. To send a signed message, open the Write window, and go to OpenPGP menu (or icon) to select the signing option. Then write your message as usual and when it is sent, it will be signed by the key asked for. There may be a dialog to select the key to be used.
17. That is all there is to it for sending a signed message
18. To set up the identity to sign by default, Edit → Account settings, select the account name → Manage identities, select the identity and under the OpenPGP tab, set the default to sign composed messages.
19. To check whether it has arrived, switch to the other user and download the mail (Get Mail icon). You will find that it detects a signed message, but cannot verify the signature because the recipient has not seen the key it was signed with. Thunderbird reports these facts in a bar above the preview of the message with a yellow background.
20. To let the recipient know about your key, switch back to Onlifa, and export the public key from the Key Management window, select the key, File → Export keys and create a public key file. When it asks whether you want the secret key included, answer NO, otherwise you will be giving away all your secrets! Save the file somewhere convenient – you may want to re-use it several times.

21. Send this exported key by e-mail as an attachment in the usual way. When you hit Send, you will be asked whether to sign each part separately. The default of signing (encrypting) every part separately is the right one in almost all cases, and is accepted by more e-mail clients than other methods.
22. Switch to other user and get mail again.
23. To incorporate the key that has been sent, save the attachment to somewhere useful, and open the Key Management window. File → Import keys will give you a dialog to navigate to the file; select it, hit OK and you are done.
24. Go back to the first message and you will see the background of the line above will have changed colour to blue. Thunderbird can now find the key, but it will indicate that it is untrusted.
25. Trust of keys is important, and comes in various flavours. First, you want to say whether you trust the sender, and Thunderbird gives you some options about that. In The Key Management window go to Edit → Set Owner Trust and set the degree to which you know you can trust the sender. Note that is not about trusting the *key* but about how much you trust the *sender*. Has their e-mail address been commandeered?
26. **Important:** this does not mean that you know that the key in fact is associated with that owner as you know them. How do you know at this stage whether they really did send that key to you? This is where the notion of "signing" a key comes in, and means a face-to-face encounter where you present a print-out of the public key as received by you, and ask the sender to verify by physically signing the page that it really is theirs. Or they might have pre-signed print-outs of their own keys to give away.
27. When you are sure that the key really does belong to the sender, then you can *sign* their key on your keyring to show the *key* is trusted. In the Key Management window, go to Edit → Sign Key for the selected key. When you now reload the messages, the key line background will have changed to green. You have now started your own "Web of Trust".
28. Switch users to Onlifa, again.
29. Upload the public key from the Key Management window, Edit → Upload Public Keys to a selected public key server (say, pgp.mit.edu). This makes it available to anyone who receives a signed message.
30. Switch back to the other account.
31. Having signed the onlifa's key, then you can upload the signed version to the public key server, too, to enlarge the web of trust for that key.
32. End of presentation.