CRYPTOLOGY, LAB 2

PGP

NAME: ____

Introduction: A word about PGP and GPG

PGP (Pretty Good Privacy) is a digital data encryption program created by Phil Zimmermann, a special director of Computer Professionals for Social Responsibility (CPSR) from 1997-2000. He created PGP to promote awareness of the privacy issue in a digital age.

As we have seen, protecting one's privacy is nothing new. It has, however, become more urgent today because of the ease with which digital data (information in databases, e-mail, and so forth) can be accessed, intercepted and monitored. It is also not unusual for sensitive information, transmitted or stored in digital form, to accidentally become public knowledge. Once data is in digital form, it easily duplicated and shared. This is why more and more organizations are looking to encrypt ALL their information.

Private individuals should think seriously about doing the same thing. A little paranoia is not a bad thing; it makes sense to take whatever means are available and within reason to protect yourself from people prying into your private affairs.

A word of warning to beginners to encryption, PGP programs may take some getting used to here and there. We are going to use a FREE, OpenSource software program that implements the PGP algorithm: The following series of tutorials aim to help you get over the initial hurdles so you can be up and running using the software without much difficulty.

The features of PGP introduced in this tutorial are all you need to know to use the program to protect your privacy in the normal run of affairs. But bear in mind that to become a power user of PGP--one who takes advantage of the full suite of encryption protections--you will need to invest some time in reading the Manual that accompanies the program

GNU Privacy Guard (aka GPG)

The program that we will be using is GnuPG, a complete and free implementation of the OpenPGP standard. It can be downloaded by going to the GnuPG site:
http://www.gnupg.org/index.en.html
For today's tutorial you will be provided a copy by your instructor (Mr. Meyer).

GnuPG allows users to encrypt and sign their data and communication, features a versatile key management system as well as access modules for all kind of public key directories. GnuPG is Free Software (meaning that it respects your freedom). It can be

freely used, modified and distributed under the terms of the GNU General Public License
.
GnuPG comes in two flavors: 1.4.10 is the well known and portable standalone version, whereas 2.0.15 is the enhanced and somewhat harder to build version.

You will be using a piece of software called Gpg4win which provides a Windows version of GnuPG. It is nicely integrated into an installer and features several front-ends as well:

Novice Tutorials:

The GnuPG manual has an excellent series of tutorials for both novices and advanced users. Unfortunately, the novice tutorials are a little date, and at this time, the advanced tutorials are only available in German…. What's up with that? The complete set of novice tutorials can be found here:
http://gpg4win.de/handbuecher/novices.html

I have selected the following tutorials for you to complete today:
Tutorial: Creating a key pair
(http://gpg4win.de/handbuecher/novices_6.html)
NOTE: Don't create your official key! This is just for practice.

Tutorial: Publishing your key per email
(http://gpg4win.de/handbuecher/novices_7.html)
NOTE: When you send your email, it must be in plain text format!

Tutorial: Sending your key to a keyserver
(http://gpg4win.de/handbuecher/novices_8.html)
NOTE: Don't actually send your key to the keyserver!

Tutorial: Decrypting an email
(http://gpg4win.de/handbuecher/novices_9.html)
NOTE: You won't be using **WinPT** just look for the "Clipboard" in the toolbar at the top of the regular PGP program.
NOTE: If you get an error that says something like "Not valid UTF-8", don't worry about it and just move on.

Tutorial: Attaching a key to your key ring
(http://gpg4win.de/handbuecher/novices_10.html)

Tutorial: Encrypting emails
(http://gpg4win.de/handbuecher/novices_11.html)

Exercises:

Try to complete the following, in class.

Exercise 1: Pick one other person in class. Send them your public key. Have them respond with a message, encrypted with your public key which includes their private key. Then decrypt the message to see what it says and to recover their key.

Exercise 2: Sign the public key that you received from the person, and send it back to them. Try to get as many people in the class as possible, to sign your key.