Lab: Private Communication

Daniel Bosk and Lennart Franked

Department of Information and Communication Systems
Mid Sweden University, SE-851 70 Sundsvall

1 Introduction

This laboratory assignment will cover how public and private keys are used in practice, how to use these to encrypt and decrypt as well as sign and verify a message or a file, also some basic key distribution. We will also cover alternative ways to try to evade prying eyes from finding your message. We will use the open source programs GNU Privacy Guard (GPG) and OpenPuff.

2 Aim

After completion of this assignment you will

- Have an understanding of how to use a public–private key-pair.
- Know how to use implementations of asymmetric ciphers.
- Be able to distribute your own key and retrieve other public keys using publicly available key servers.
- Be able to use steganography as a way to hide messages.

3 Reading instructions

Before starting this assignment you should have read chapters 5 and 23.4.4–5 in *Security Engineering* [1]. You should also read the paper "Exploring steganography: Seeing the unseen" [3] to fully understand how steganography works in practice. (Other recommended papers are "On the limits of steganography" [2] and "Hide and seek: An introduction to steganography" [7].)

During this assignment you should consult the documentation [4–6, 8] for instructions on how to use the specific softwares.

4 Tasks

This assignment is divided into two parts. The first part will cover cryptography—to hide the data—using email as a usage example. The second part will cover steganography—to hide the presence of data.

4.1 Email security

In this part you will work with email security. You will start by creating your own key-pair after which you will upload your public key to one of the public-key servers. Once you have done that you will send an encrypted email to one of your classmates which will be your lab partner. You should be able to find his or her public key on the key servers.

Start by downloading and installing GPG,[1] select the appropriate version of GPG depending on what operating system you use. Once you have GPG installed on your system, generate a key-pair and *make sure to make an active choice for what cipher and key size to use.* When finished, export your public key to the following key server:

`pgp.mit.edu`

Next you shall import both your lab partner's key and your tutor's key, they should be available on the same key server. When you and your lab partner have each other's public keys, send an encrypted email to each other and *confirm that the other party is able to decrypt your email.*

Next, find a way to communicate with your partner, such that you can confirm that they are who they say they are, and ask them to repeat the fingerprint of their public key. Once verified you can publicly sign their public key with an approprite trust level based on the type of verification you did. Give this signed version of the key back to your lab partner so he or she can import it and resend it to the key server.

When you have completed these steps you must send an encrypted email to the tutor and await an encrypted response. The tutor will not accept your email unless your key on the key server is signed by at least one person.

Social engineering using spoofing (optional) Try to trick a different classmate that you are their lab partner. Do this by spoofing an email or any other form of communication. Will you be able trick them into believing that your fingerprint is their partners? What measures must be taken in order not to be tricked?

4.2 Steganography

In this part you will work with steganography. By using either Outguess or OpenPuff you will get a practical understanding of how steganography works.

Outguess is already installed in the computer lab. But it should be available in most BSD and Linux repositories. You can find OpenPuff at the following URL:

`http://embeddedsw.net/OpenPuff_Steganography_Home.html`

[1] If you are in a computer lab on campus, this is most probably already done.

Once you have either program installed, write a message in a text file and hide it in a picture. You will then post this picture in the course forum, where your partner can access your picture and retrieve the hidden message. You can use a password with the steganographic software, encrypt this password for your lab partner's and the tutor's key using GPG. Post this encrypted secret in the course forum with the picture. Now only your lab partner and the tutor will be able to decrypt the secret and retrieve the message from the picture.

Retrieving other groups' messages (Optional) Try to retrieve a hidden message posted by another group. If retrieved, post the hidden message as a reply to that picture. You will then be eligible for the grade *Pass with Distinction* on this assignment.

5 Examination

You must hand in a report including the following:

- A short summary of your key-generation process, including what cipher you chose and the key length. *You must motivate this, "Because it was default" is not a valid motivation.*
- An explanation detailing how you verified that your partners public key was the correct one. Motivate why you chose the method you did, and based on this, how certain you are that this person actually is the same person that owns the public key.
- A copy of the secret message you received from your tutor by email.
- A 250-words detailed description of how steganography works.
- Any proof of that you solved the optional assignments.

References

[1] Ross J. Anderson. *Security Engineering. A guide to building dependable distributed systems*. 2nd ed. Indianapolis, IN: Wiley, 2008. ISBN: 978-0-470-06852-6 (hbk.) URL: http://www.cl.cam.ac.uk/~rja14/book.html.

[2] Ross J Anderson and Fabien AP Petitcolas. "On the limits of steganography". In: *Selected Areas in Communications, IEEE Journal on* 16.4 (1998), pp. 474–481. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=668971.

[3] Neil F Johnson and Sushil Jajodia. "Exploring steganography: Seeing the unseen". In: *Computer* 31.2 (1998), pp. 26–34. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4655281.

[4] Werner Koch. *Using the GNU Privacy Guard*. Mar. 2012. URL: http://www.gnupg.org/documentation/manuals/gnupg.pdf.

[5] Eng. Cosimo Oliboni. *OpenPuff v4.00 Steganography & and Watermarking*. July 2012. URL: http://embeddedsw.net/doc/OpenPuff_Help_EN.pdf.

[6] Niels Provos. *outguess - universal steganographic tool*. URL: http://manpages.ubuntu.com/manpages/utopic/man1/outguess.1.html.

[7] Niels Provos and Peter Honeyman. "Hide and seek: An introduction to steganography". In: *Security & Privacy, IEEE* 1.3 (2003), pp. 32–44. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1203220.

[8] The Gpg4win Initiative. *The Gpg4win Compendium*. Aug. 2010. URL: http://wald.intevation.org/frs/download.php/775/gpg4win-compendium-en-3.0.0-beta1.pdf.