Laboratory Assignment 2: Cryptography

Olof Magnusson and Yu Hsuan Liao

Computer Security EDA263

August 11, 2025

Contents

1 Questions

1. What are your names, lab-group, and civic registration numbers?

   Answer: Olof Magnusson, Group 51, 940121. Yu Hsuan Liao, Group 51, 950118

2. What is the fingerprint of your key?

   Answer: A722 B6F3 E6CA A736 0ABD DB1B DC3C 310F 9545 FC5B.

3. What mail client did you use for the lab?

   Answer: For this lab, we used Thunderbird. The primary reason is that Thunderbird is already integrated with GPG in such way that we can encrypt and decrypt using the Thunderbird's interface. There is much information available how to get started and many plugins which makes it more efficient than doing everything in a command line.

4. Will you use gpgin the future? Please specify your reasons.

   Answer: From our experience with this lab, we find this very beneficial and a smooth way when exchanging critical information. Regarding if we would use gpin in the future, it depends on the standards the company is using for encrypting/decryping sensitive information. For private use, it is pretty hard to deploy since our friends and family are not technically interested, thus it would be hard to convince them to use gpg to decrypt my messages.

5. What is the purpose of signing in terms of the "CIA" (Confidentiality, Integrity,Availability)? Explain how signing works in PGP? Why is a hash function used?

   Answer: The purpose to signing in terms of CIA is that we want to overcome a problem. One of them is the integrity which implies that only authorized individuals should be granted to modify the data. We can see that the message transmitted from recipient have not been tampered with. Thus, a digital signature certifies and timestamps a message, and thus assure the originality in the message. In PGP this works in the follow way [1, 2]:

   1. Import a key into a keyring `(gpg --import name_of_pubkey)`
   2. Verify fingerprint of the key (ask the owner, to be 100%) `(gpg --verify name_email@address.com)`
   3. Sign the UID with gpg `(gpg --sign-key name@address.com)`
   4. Export the signed public key `(gpg --output signed_key --export --armor email@address.com)`
   5. Check if the key is signed `(gpg -list-sig key-id)`
   5. Email the signed key to the recipient mail address.
   6. The receiver imports the key into a keyring (`gpg --import name_of_pubkey`
   7. The receiver checks that the key is correctly imported `(gpg --list-keys)`
   8. The receiver checks if the key is signed `(gpg -check-sigs name@address.com)`

   A hash function is a mathematical summary which used to digitally sign messages [3]. We therefore, use a fixed-length hashing function for the key to assure the originality. The hash function is beneficial in the way that any changes of the file will provide different output. Thus, when we compare the new signature and the digest sent by, for example, Bob they must be the same. On the other hand, if the messages is not the same, we can conclude that the message has been manipulated on transit.

6. What is the purpose of encryption in terms of the "CIA"? Motivate your answer.Explain how encryption and decryption works in PGP? Make sure that your answer includes how PGP combines the best features of both conventional and public key cryptography.

Answer: The purpose of encryption in terms of CIA is to achieve confidentiality in the communication. We do not want any unauthorized to be able to detect the data which is transmitted, e.g we want to be able to communicate through a secret medium. The encryption works in the following way [4]:

1. Generate a random key `(gpg --full-generate key)`
2. Encrypt data using random key
3. Encrypt key using receivers public key
`(gpg --encrypt --armor -r name@email.com name_of_file)`
4. Send data to the receiver

Decryption:
1. Receive encrypted message
2. Import the key to the key ring `(gpg --import name_of_pubkey)`
3. Use receivers private key to decrypt
4. Decrypt the data using key `gpg --decrypt name.asc > name.txt`
5. Verify that decrypting was successfully `(less name.txt.asc)`
5. Check diff (`diff name.txt name.asc`)

A more detailed view of the encryption and decryption process can be viewed in 1.

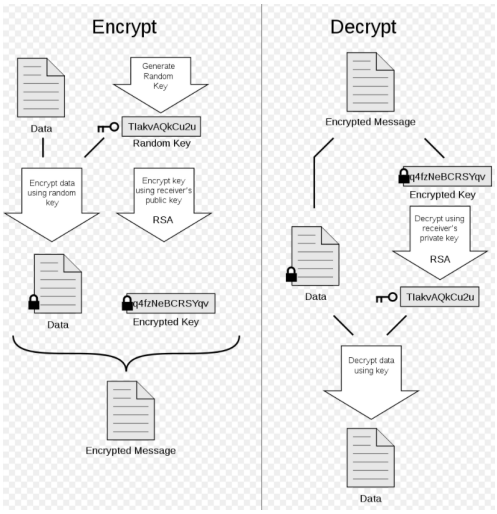

Figure 1: Encryption and decryption in PGP
[4]

Conventional because it is a symmetric algorithm which uses the same key for both the encryption and the decryption. Thus, it is very fast, simple and it comes at a very low cost compared

to public-key encryption. However, public encryption comes with several benefits. One primary is the distribution of keys to multiple recipients which is a problem with symmetric encryption. Compared to symmetric, it is safer since we are using two keys, one for the decryption and one public which is distributed over a medium [5].

7. In the lab you validated the keys by asking the owners verbally for the fingerprint. Explain how the trust system (web of trust) in GnuPG works and how you validate a key belonging to a person you do not know personally. Your answer must distinguish between(owner) trust and(key) validity (sometimes called validity trust). You should also include the conditions for when you start to consider keys from foreigners valid

   Answer: This is based on certain criterias [6].
   1. The key is personally signed.
   2. It has already been signed by one fully trusted key
   3. It has been signed by three marginally trusted keys
   4. The path of signed keys from the k to k-1 is less than six steps.

   So, the level of trusted parameter for a certain individual is a value that we need to set by our self (we can set it either high or low). The validity is different since it is calculated based on web of trust [7].

8. It is common practice during this lab to exchange the keys via e-mail. Is it a trustful way to share keys? What about uploading the keys on Canvas or sharing them on your personal Facebook profile1? Explain how using different communication technologies/instant messaging applications/social networks can change your perception of trust.

   Answer: There are no risks associated with publishing the public key. However, we would say that this is not best practice to share public keys nor uploading the keys on Canvas or sharing them on a personal Facebook profile. The best practice is to distribute the keys on a key server. Some examples are **keyserver.ubuntu.com keyserver.pgp.com** often used associated with SeaHorse to automatize synchronization with the keys. Thus, if you would like to distribute the keys to a user named Alice, the only thing we need to do is to link that particular page of the key server. However, we do encounter some issues with this method. One of the issues is that the key needs to be signed by multiple individuals to have complete trust (my brother trust it, is certainly not enough). Therefore, we need to think about key signing parties to have the full effect of the method [8].

   Regarding how different message communication technologies can change our perception of trust, this depends on what type of PKI the application uses (client-to-client PKI could be assumed more trustworthy than client-to-server PKI) using a server as an middleman of the communication. Nevertheless, we still use Facebook and Snapchat, and even though there has been security breaches and information disclosure, we are still using it. However, we both agree that we do not distribute any sensitive information on that platform based on that.

   (a) You know Dharma, and you now enter that you trust her fully. If you list the key for Dharma, it would now show: f/m (trust: full, validity: marginal).

What would be listed for Elena? Answer using the form: Elena trust: x, validity: x

Answer: Elena trust: unknown, validity unknown

(b) Time passes, and you realize that Chloe is actually somewhat trustworthy. For that reason, you change the owner trust of Chloe to marginal (Chloe: m/f). What would now be listed for Elena? We assume the changes done in (a) are still in effect.

Answer: Elena trust: unknown, validity full

9. What is your public key (in ASCII-format)? Use the menu option Attach My Public Key in Enigmail/OpenPGP to include your key with the email. You can also manually insert the key into the email. First, you need to export a fresh copy of the key (with the armor option) to a file. Second, include the contents of this file into the email. Make sure that there is a blank line between your answers above and the start of the preamble of the key block:

Answer:

———BEGIN PGP PUBLIC KEY BLOCK———

mQMuBGAjj20RCADVboHETYaOtaRaItb8b825lDTowG7E1RLPFmecysC7a6AdwlMt
nO9XYp+tXRpS/NSPPc0ky2tvoFF3Vwz/45h1/nZdfApY/goUHh8ecLMJwrpHqsN9
u3AxjD1CUmNJBw2VJG+hRGLBKbrQPVI0RJ485agYpKSAX2YMr2M3B3ablf9R8WBQ
xNC40uKmXBNaAhYqXmu57PsBOpucoEVZDIIy1hVbsr5UvEervtG9EDtDoOY8UjNJ
7GTs+cDTgkuLvCBCkD5+sEWHrHd69tbpUulVstTEMg9jZKs2KNXpgbrIC8eK1Hkp
VLaoj4kxz6OLRh2h0gRgtvFwPWvVw5YrdYczAQCfTw/hIO7ZZKUrLvvyF8uAG8Zf
aGMzXZ4xUuiI6Pzz1wf9G44+fc5tqrFcXdQ+QSGCoIGQvzKaXrtlmkP3e2vvhsCd
9vZTlqkpl+4Ilf24vzhRYoATbjZ3cfP6y8GayC5TcEejtBLAkqOJSkBwedexwye0
2fUgmS3yGOsNrCbVMwXkT0tfrcfiBtFtkkpsYMVKd59Vy/ZKaiEMoW6rtYRtxhoz
KGyuKJfBcHYsrbxSya/DlfuuRMRWezgPolQ4Vi3qQwZijv5TPFq1vf7GfOEL/cJB
z3fwIj5FnHlnaN3lAAjerdEfkSHDdJxguI6daOW2GwA/Eh4jf9sI0dOJvau/DzDT
Pwrxnhp0IxVpuRcTfU+4+RQ1XN9mqyEBLa+WbjljaQf/QtVy8e511TzsxOJgZexD
JAzEvF8Uwqr8qI2wcExTj99P3FaSj8qUvLnoSj0tJ4/X26ERmnFjRsckKSW9iYWz
DY+X2DGwNLDRk9w5/qt1z738/Gi0eLOLKsNQA3utT1wPfyRwTt75byDiUK3T56Xe
HZ6X+kMlk9k58foul4iHDn3tsSQWULBGWfwuUvaBQyXIqWJAzO1wZA9sDYvm7uuO
oHPfBMNWln6bxvXOXJkdxZOhRxq7wE9DwzaDFcUk8+12I035SD5e9s8lhXM4lAcs
2JXMo+ZaLVcgA0nMG/x8pbIl1unNC3+HcBw1/5dFrQB4OnFwTkWKQhrPvyskmJc+
LLReT2xvZiBNYWdudXNzb24gR3JvdXAgNTEgKEtleSB0byBiZSB1c2VkIGluIHRo
ZSBjb3Vyc2UgRURBMjYzL0RJVDY0MSkgPGd1c21hZ29sZkBzdHVkZW50Lmd1LnNl
PoiWBBMRCAA+FiEEpyK28+bKpzYKvdsb3DwxD5VF/FsFAmAjj20CGwMFCQCRBQAF
CwkIBwIGFQoJCAsCBBYCAwECHgECF4AACgkQ3DwxD5VF/Fu+/wD+N+zUbZEaSjUx
Qe63SqL6F2KY1TWI2/gm4UBPpEAl1hIA+wRgGZ3qKNyDwkVVKHQopSyRTDcjnPf1
7Q5Ie2sp0Dt0tExZdUhzdWFuTGlhbyAoTGFicGFydG5lciBpbiBjb3Vyc2UgRURB
MjYzL0RJVDY0MSkgPGhzdWFubm9iZWxwcml6ZUBnbWFpbC5jb20+iJYEExEIAD4W
IQSnIrbz5sqnNgq92xvcPDEPlUX8WwUCYCOZHgIbAwUJAJEFAAULCQgHAgYVCgkI

CwIEFgIDAQIeAQIXgAAKCRDcPDEPlUX8W2hFAP46qMsC9tP4Cog/0T52JpdAxBMW
0dvVJE/naXI84agScgD+I49Ez6eaQwWtwQvz/00uZBeG2DE8fpKD1oVgUw8sChC5
Ag0EYCOPbRAIANTE72K7YFIa867Ck6kX5958ulLRNCzVfJ03Af3QKwGQlVOIXnBh
p0/flUifs+szukYiGTUoHQNXySXmuaz9A2czQeiAVUVX/iYAL61k/J/1/zP3TDLZ
OGuEWRd20InHBNTFaoKIgPME7TIDDOEHjtUuKtCahIcq8iTK1oOvZjDPfdiB2cFN
w6/UniaBMXLUYz30miIq4Bgx9txbgU37gU7jU4L8jUkjqo+4qktmgKuc/4Nb49Xo
4tQRf/hT/2/rVwHHUawJ8+qxJkjG2A6q+5Dgyjb/wADI2slm7buJjtsciQEhpVh2
yWxIJqfk4ak7xI6jKa/4Ox4Hi9Dbynp7QQsAAwUIAJtUoqmghbZJQ+KQvhcXwK6c
LUIa3jV+mumWrv/sl2cJ9G122hE+XlYaRZiSIeAWe9/+cfqzqK/5/NFwm1S+keM7
Ax4JLT+0HDFECHwK2gImnsaTYtjVJxWOaujwWGUV7WR4p5MYS3UN8lZwjmyf1GkI
YXvHxhkr6uhjNG1WRPXKk8CuFmWdXvKgHlCV4GEdAubJauOZaDAwRDHvV9wgAsIM
sKXqDXC0auNRZ0np92p5yptoT3Dmi38Q0NOWEpmP2vPFd4lVQvMB6/HfA8ZJCYBP
9VXkVu/bLR5pUiELuT6BOLdic+nDQlBvu86uOP4ApcFqEF46NeGpshg8QQx/Iw6I
fgQYEQgAJhYhBKcitvPmyqc2Cr3bG9w8MQ+VRfxbBQJgI49tAhsMBQkAkQUAAAoJ
ENw8MQ+VRfxb5C0A/1rOwgaEt/SwJ/KsFaDRGeEbyOF5fgsYFtSy0NdwHO/BAPwK
6bH+NS+pRf1y9uH9nT3aqvZnFYPchljGEGv3EpSTuw==
=OCZo
———END PGP PUBLIC KEY BLOCK———

2 Conclusion and reflections

The lab has provided a general overview of cryptographic primitives: public-key cryptography, conventional cryptography with its advantages and disadvantages. For this lab, it has been interesting to see how cryptography in a more realistic environment—especially when signing others students keys and to connect the theoretic information to the practice. There is no doubt that this is an essential topic in computer security to create effective and sustainable solutions both for general-purpose computers and embedded systems.

References

[1] "Signing pgp keys."
Online: ttps://carouth.com/articles/signing-pgp-keys/ Accessed: 2021-02-16, carouth.com.

[2] R. Wright, "Pretty good privacy (pgp)."
Online: https://searchsecurity.techtarget.com/definition/Pretty-Good-Privacy Accessed: 2021-02-11, techtarget.com.

[3] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya, "Merkle-damgård revisited: How to construct a hash function," in *Annual International Cryptology Conference*, pp. 430–448, Springer, 2005.

[4] "Pretty good privacy."
Online: https://en.wikipedia.org/wiki/Pretty_Good_Privacy Accessed: 2021-02-15, wikipedia.com.

[5] J. Hoffstein, J. Pipher, J. H. Silverman, and J. H. Silverman, *An introduction to mathematical cryptography*, vol. 1. Springer, 2008.

[6] Gnupg, "Validating other keys on your public keyring."
Online: https://www.gnupg.org/gph/en/manual/x334.html Accessed: 2021-02-15, gnupg.com.

[7] J. M. Ashley, M. Copeland, J. Grahn, and D. A. Wheeler, "The gnu privacy handbook," *The Free Software Foundation*, vol. 34, 1999.

[8] M. Davidson, "How should i distribute my public key?."
Online: https://security.stackexchange.com/questions/406/how-should-i-distribute-my-public-key Accessed: 2021-02-12, techtarget.com.