Berk Emre Mert
bmert24@student.oulu.fi

Task 1: Private and authentic messaging

Task 1A) Signing a message 1/4p
Create a gpg keypair, use the private key with a passphrase to sign a message and return both keys here.

gpg: key 4FD5829C1D20FB5A marked as ultimately trusted
gpg: directory '/home/ubuntu/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/ubuntu/.gnupg/openpgp-revocs.d/2D6D4B36E9F69F45EA6744A84FD5829C1D20FB5A.rev'
public and secret key created and signed.

pub rsa4096 2025-03-11 [SC] [expires: 2025-03-17]
2D6D4B36E9F69F45EA6744A84FD5829C1D20FB5A
uid Berk Emre (This is my message for the Privacy and Social Engineering course in Oulu.) <berkemremert@gmail.com>
sub rsa4096 2025-03-11 [E] [expires: 2025-03-17]

My private key: private-key.asc

Task 1B) Encrypting the message 2/4p
Encrypt the message using the provided public key in the "files" folder. Send the encrypted message to email address for the task, the subject should be your full name - this matters for grading the task. It does not matter what address you send the email from. Mark this task done.

I sent the encrypted message to taskmailaddress@proton.me but I am not sure I truly sent it.

I sent it with mutt. The command:

```
mutt -s "Berk Emre Mert" taskmailaddress@proton.me -a message.txt.asc < /dev/null
```

Task 1C) Verifying a message 3/4p
Download the message in the "files" folder and verify the signature on it with the public key in the same folder. Answer this part with the name of the owner of said keys.

Click here to see the message file

Task 1D) Questions 4/4p

What can be found out about the email you sent, by one who intercepted it in transit?

If someone intercepts the encrypted email **in transit**, they can learn:
Metadata (unencrypted):
-The sender's and recipient's email addresses.
-The subject line (if not encrypted separately).
-Timestamps and mail server details.
-Email size.
Content (encrypted):
-The body of the email is encrypted, so the interceptor **cannot** read it.
-The encrypted text looks like random gibberish unless they have the recipient's private key.
-However, if an attacker has access to both the **encrypted message** and a compromised **private key**, they could decrypt and read the contents.

Does verifying the the message guarantee the senders identity?

Not necessarily. **Verifying a GPG signature** ensures that the message **wasn't tampered with** and that it came from someone who holds the **private key corresponding to the signed public key**. However, it does **not** guarantee the sender's **real identity** unless you trust the key. To improve trust:
-Check the key fingerprint and confirm it with the sender via a trusted channel.
-Use a **web of trust** where multiple trusted people sign the sender's key.
If an attacker generated their own key with a fake name and you blindly trust it, they could impersonate someone.

Is the process of sending an email this way end-to-end-encrypted(E2EE)?

Yes ☺ because:
-Only the **recipient** (who holds the private key) can decrypt the message.
-Even email providers **cannot** read the message content.
-If the email is intercepted, only encrypted data is visible.
However:
-Metadata is not encrypted
-If the recipient's private key is compromised, encryption is useless.
-If you store unencrypted drafts or plaintext messages, encryption doesn't protect you.
For **full E2EE protection**, both sender and recipient should:
-Use **encrypted subject lines**.
-Store emails in **encrypted form** after receiving them.
-Use **secure email services** that support PGP (like ProtonMail).

Task 2: Metadata and messaging

Task 2A) Compare messaging platforms

Messaging apps collect different types of user data and offer varying levels of security. **WhatsApp** gathers phone numbers, contacts, and device info, and shares data with Meta outside the EU. **Signal** collects almost nothing except a phone number, making it the most privacy-friendly. **Telegram** stores contacts and user IDs and recently announced it may share user data with governments. **Messenger** collects extensive personal information, including names, emails, and financial details.

Encryption also differs. **Signal encrypts everything by default, including metadata**, making it the most secure. **WhatsApp has end-to-end encryption (E2EE) by default but leaves backups unencrypted** unless manually enabled. **Telegram and Messenger require users to enable E2EE manually**, meaning most chats are not fully secure.

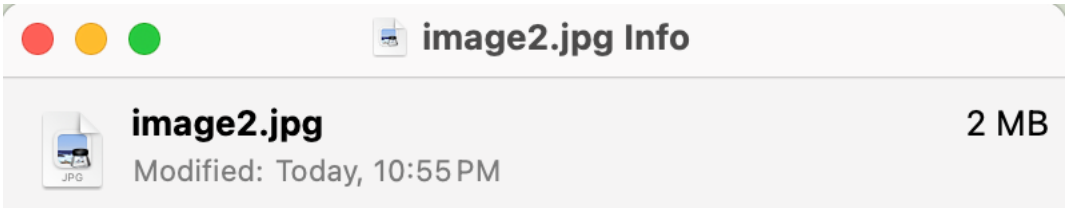
Metadata, even without message content, can reveal user habits and relationships. If User A always messages User B at the same time each week, patterns emerge that can be tracked. **Signal is one of the few apps that encrypts metadata**, while others leave it exposed.

Using the same phone number across platforms also raises privacy concerns. **WhatsApp's data-sharing with Meta** allows for detailed user profiling and behavior prediction.

Finally, features like **"last seen" and "online status" can be exploited**. If someone tracks when contacts are active, they could monitor habits, which can be a privacy risk.

In conclusion, **Signal is the best choice for privacy**, while WhatsApp, Telegram, and Messenger have varying weaknesses. Users should carefully check app settings and data policies rather than assuming an app is private just because it claims to be.

Task 2B) Image metadata



More Info:

Where from:	https://raw.githubusercontent.com/ouspg/PrivacyAndSocialEngineering/refs/heads/main/2.Messaging_and_mobile_privacy/images/image2.jpg (2)
Dimensions:	2736×3648
Device make:	HUAWEI
Device model:	CLT-L29
Color space:	RGB
Color profile:	sRGB IEC61966-2.1
Focal length:	5,58 mm
Alpha channel:	No
Red eye:	No
Metering mode:	Pattern
F number:	f/1,8
Exposure program:	Normal
Exposure time:	0,03
Latitude:	65° 3' 27,614" N
Longitude:	25° 28' 7,116" E

Task 3: Application permissions and trackers

I chose Spotify for this Excersize.
https://reports.exodus-privacy.eu.org/en/reports/com.spotify.music/latest/

1. Trackers and Permissions
Trackers: The Spotify Android app reportedly includes 13 trackers that monitor user behavior and share data with third-party companies.
Permissions: While the exact number of permissions requested by Spotify can vary depending on the device and app version, it typically requests access to:
-Storage
-Microphone
-Contacts
-Location
-Phone status
-Network connections

2. Dangerous and Special Permissions
According to Google's classification, "dangerous" permissions require user approval at runtime due to their potential impact on user privacy. For Spotify, these include:
Microphone Access: Allows the app to record audio.
Contacts Access: Permits the app to read user contacts.
Location Access: Enables the app to determine the device's location.

3. Permissions for Monetary Gain
Certain permissions and trackers in Spotify can be utilized for monetary purposes:
Trackers: The 13 trackers present in the app collect user behavior data, which can be used for targeted advertising and partnerships with third-party companies.
Permissions:
-Location Access: Allows Spotify to offer location-based advertisements or promotions.
-Microphone Access: Could be used to analyze ambient sounds for contextual advertising, though there's no evidence Spotify uses it this way.
-Contacts Access: May facilitate social features or network-based advertising strategies.

4. Potential Attack Vectors
If an attacker gains access to the Spotify application or its database, the following attack vectors could be exploited:
Microphone Access: An attacker could eavesdrop on users, capturing sensitive conversations or ambient sounds.
Contacts Access: Compromising this permission could lead to unauthorized access to a user's contact list, enabling phishing attacks or the spread of malware.

5. Android and iOS Privacy Labels Comparison
Privacy labels on app stores are intended to inform users about data collection practices. However, discrepancies can exist between platforms:
-Android: Google Play's Data Safety section relies on developer-provided information, which may not always be verified.
-iOS: Apple's App Store requires developers to self-report their data practices for privacy labels, but these are also based on the honor system.

Task 4: Application SDKs, code signatures, Tags and Pixels (bonus)

Task 4A) Legacy technologies

Apple’s App Tracking Transparency or ATT was a big change in how apps track people. Before this, apps could follow users across different apps and websites using something called IDFA without asking. But after ATT, apps had to ask for permission first. Most people, like 96 percent in the US, said no. This made it way harder for companies to track people and show targeted ads.

Big companies like Facebook (now Meta) got hit hard because they rely a lot on ads that follow users around. Since they couldn’t track users the same way anymore, their ads became less effective, and they lost a lot of money. Some companies started using other ways to track people, like device fingerprinting, which collects info about your phone to guess who you are. But this is kind of shady and might not last because of privacy laws. Now companies are looking at different ways to show ads. One method is contextual advertising, where ads are based on what you're looking at instead of your past behavior.

So instead of tracking you across apps, they just show ads that fit whatever website or app you're using at the moment. This is better for privacy but might not be as effective for advertisers. In the end, ATT changed a lot about mobile ads, and companies are still trying to figure out what to do next.

Task 4B) The use of the advanced technologies

Tracking users online has become more advanced, especially with things like SDKs and tracking pixels. SDKs are bits of code that companies put inside apps to collect data about users, like what they do in the app and when they open it. Big companies like TikTok and AppsFlyer work together to track app installs and user activity, so advertisers know if their ads are working.

Pixels or tags work in a similar way but are mostly used on websites. They are tiny pieces of code that track what people do after seeing an ad. Many companies like Google, Meta, and TikTok have their own pixels to help businesses see if their ads are making people visit their websites. These trackers are everywhere, and you might not even notice them unless you use a special tool to check.

When I looked at Finnish websites like motonet.fi, masku.com, wolt.com, and others, I found that many of them use multiple tracking pixels. Companies use these to collect data on how people move between different pages and whether they buy something after clicking an ad. The same thing happens in apps, but there it's hidden inside SDKs instead of website pixels.

Another thing that helps tracking is using "Login with Google" or "Login with Facebook" buttons. It makes signing into websites easy, but it also lets those companies collect more data about you. They can see which apps or websites you use and connect that data to your profile. This helps them show you ads based on what they learn about you.

Overall, tracking is everywhere, and it's getting smarter. Companies want to know what people do online to improve their ads and make more money. But at the same time, it raises big privacy concerns because users don't always realize how much of their data is being collected.

Sources for task 4b:

-AppsFlyer & TikTok integration guide
-Browser tools for detecting tracking pixels on websites