Exercise set: 6
Tommaso Carraro
student number: 015147592

Introduction to IoT: sixth set of exercises

First exercise

For this exercise I have chosen the first application:

1. A privacy risk is the collection of personal data by the sensors in the car: the time when the car has been driven, the route that the driver has taken, the number of people in the car and finally the driver profile. This is a risk because these information are periodically sent to the back-end of the manufacturer, so they can be intercepted by an attacker that could use them for malicious purposes. In fact, the attacker could know if the all family is out of the home, where the family has been (learn family habits) and finally everything about the driver personal data included in the driver profile. This is the data surveillance attack.
2. A way I would use to prevent the data surveillance attack in this case is the pseudonymization of these information using asymmetric cryptography. In this way the car will use the public key of the manufacturer to encrypt these data before sending them, while the manufacturer will use its own private key to decrypt these information before doing its analisys. In this way the attacker has to know the private key of the manufacturer to be able to access to these personal information.

Second exercise

1. One of the methods to perform pseudonymisation is by using encryption to create the pseudonym by making the mapping irreversible without access to the encryption key. In particular, I decided to use AES to encrypt the string “identity+occupation”. In this way I can encrypt two information in one time. The only way the attacker has to access these information is to know the key that has been used for the encryption. Another way to make this process more robust was to create a pseudonym for the identity and a pseudonym for the occupation in such a way that the attacker has to find two keys to access to the entire information. But I assume that if the attacker is able to find one key, he will be able to find the other one too, so I decided to encrypt the two information simultaneously. To encypt these information I used online AES encryption (https://aesencryption.net/) using a block size of 128 bit and the following key: **a=d&E&kE5$d$)e**. I provide only one example and for the rest of the table the same procedure has to be applied:
   Bruce Wayne+Businessman → h3JwxK8SwUPSSFXEc2QMIzDCCHOPZr3nGuipPjqjvzU=
   The pseudonym that has been computed has to be put on the table in place of the two field that have to be hidden.
2. To perform the 2 and 3-anonymity I used generalization of the weight coloumn of the table. To save a little of bit of space and time I put the two anonymity in the same table together with the identity and the weight of the superheros.

Identity	Weight	2-anonymity	3-anonymity
Bruce Wayne	95	92-101	80-95
Barry Allen	88	85-92	80-95
Ollie Scott	80	75-85	70-80
Alan Scott	91	85-92	80-95
Clark Kent	107	101-115	95-107
Diana of Themyscira	75	65-75	70-80
Kate Kane	66	65-75	55-70
Arthur Curry	147	115-150	107-150
Peter Parker	76	75-85	70-80
Steve Rogers	100	92-101	95-107
Tony Stark	102	101-115	95-107
James Howlett	136	115-150	107-150
Carol Danvers	56	55-65	55-70
Natalia Romanova	57	55-65	55-70

Looking at this table it is possible to see that for the 2-anonymity the attribute values can be indistinctly matched to at least 2 individuals, while for the 3-anonymity the attribute values can be indistinctly matched to at least 3 individuals. Another thing I want to say is that using bigger ranges for the generalization should be better to avoid the attacker to guess at which indivual a record belongs to. This table is to show the construction of my 2 and 3-anonymity, so in the name field only the first letter should be shown and the weight one should be removed in the real scenario. The final table is presented below.

Identity	2-anonymity	3-anonymity
B***	92-101	80-95
B***	85-92	80-95
O***	75-85	70-80
A***	85-92	80-95
C***	101-115	95-107
D***	65-75	70-80

K***	65-75	55-70
A***	115-150	107-150
P***	75-85	70-80
S***	92-101	95-107
T***	101-115	95-107
J***	115-150	107-150
C***	55-65	55-70
N***	55-65	55-70

For the examples where the de-identification can re-identify the name these are the data that the attacker knows.

Identity	Weight
Bruce Wayne	93+
Barry Allen	90+
Ollie Scott	77+
Alan Scott	87+
Clark Kent	104+
Diana of Themyscira	71+
Kate Kane	67+
Arthur Curry	116+
Peter Parker	77+
Steve Rogers	100+
Tony Stark	100+
James Howlett	118+
Carol Danvers	60+
Natalia Romanova	61+

Example 2-anonymity: since we know that Bruce Wayne is heavier than 93 kg and Barry Allen is heavier than 90 kg and their names begin with the same letter (B), they should be in the first two rows of the table. Since Bruce Wayne is 93+ he should stay in the first row because in the second one the range is 85-92 kg.

Example 3-anonymity: we know that the only three superheros that are in the range 80-95 are Bruce Wayne, Barry Allen and Alan Scott because they weigh 93+, 90+ and 87+. By exclusion we can identify Alan Scott because he's the only person in this range which the name begins for A.

Third exercise

I have found this small article about an IoT vulnerability being exploited:
"IoT devices have tremendous potential in the field of medicine. However, the stakes are very high as far as security is concerned. This was starkly illustrated by an incident in 2017 when the FDA announced that they had discovered a serious vulnerability in implantable pacemakers made by St. Jude Medical. Anyone who has watched the Homeland will be familiar with this attack.
In this case, the vulnerability laid in the transmitter that pacemakers used to communicate with external services. These pacemakers relayed information about the patient's conditions to their physicians, which made monitoring of each patient much easier. Once attackers gained access to pacemaker's transmitter, they were able to alter its functioning, deplete the battery, and even administer potentially fatal shocks.".

1. The attack that was performed in this case is the hardware backdooring because the attacker has bypassed the security measures of the transmitter to get access to the pacemaker. A possible attack in terms of privacy could be the data surveillance because the attacker could monitor the conditions of the patient using the data generated by the pacemaker.
2. In terms of privacy the only sensitive information that are recorded by the pacemaker are the patient's conditions. A way to prevent the data surveillance is to perform the pseudonymization using cryptography techniques. Since with an hardware backdoor the attacker is able to access the code of the pacemaker the pseudonymization couldn't be enough. The best way to prevent the data surveillance in case of hardware backdooring is to use data anomyzation and in particular the swapping. I have chosen the swapping because the data generated by the pacemaker can't be suppressed because they are critical for the doctors to know the conditions of the patient and since the attacker has access to the system and the code an hash function could be easily infered from the source code. Using the swapping it is less probable that the hacker will find the correspondences.
3. A way to avoid hardware backdooring is by ensuring firmware integrity and confidentiality. It is possible to use a secure boot with asymmetric cryptography: the hash of the firmware code is signed with manufacturers private key, while the public key is stored on the device. During runtime the firmware hash is compared against the public key. Another way to avoid the hardware backdooring is by forcing the changing of the default passwords in the devices that have been sold.
4. This attack involves three types of attacks:

   a. hardware attack because the attacker has installed a custom firmware to obtain the access to the system;
   b. network attack because the attacker has used the network to intercept the messagges sent by the pacemaker's transmitter and then to find it;
   c. sensing attack because the attacker has used the device's sensors to administer potentially fatal shocks to the patients.

Bonus one

This is the IoT data privacy checklist for the first application of the first exercise:

1. This application is periodically collecting the time that the car has been driven, the route that has been taken, the number of people inside the vehicle and finally the driver profile.
2. These data are sent to the back-end of the manufacturer that is probably on a server cloud.
3. These data are used to evaluate the performance of the components in the vehicle, to assess they could be improved in upcoming models, and to inform the driver of potential faults in the car. Additionally, the driver can access these data locally to know the current driving conditions including speed, location and traffic.
4. All these information can be used to perform these tasks, in fact:
   a. the number of people in the vehicle gives information about how much the vehicle has been loaded;
   b. the route that has been taken gives information about the state of the route: if the route if full of holes this is probably the cause of a problem to the suspensions of the vehicle;
   c. the time when the car has been driven gives information about how much the car has been used;
   d. the driver profile gives information about its driving style: there are driving styles that are subject to an increasing number of faults in the vehicle.
5. These data could be intercepted by an attacker during their way to the back-end of the manufacturer. Additionally, the server cloud is probably managed by a third company that can have access to these data.
6. The data is ultimately stored in the back-end of the manufacturer that is probably in a server cloud managed by a third company.
7. The data will stay in the server for the time that is necessary to perform the inference of the machine learning models that have to predict the faults in the car and suggest improvements for the upcoming models. Additionally, these data can be added to the dataset to perform the training of these models, so they will probably used for a long time.
8. It is useful to use this technique to prevent the data to be accessed, in fact more the data stay in the storage of the server cloud and more it is probable that these data will be intercepted by someone.
9. It depends on the techniques used to prevent the data to be accessed. As I said in the first exercise the best way to protect these data it is probably to use a data encryption technique using an asymmetric cryptography algorithm.

10. If a data breach has been detected by the company the safest way to inform the client about that is using the mail service, because an e-mail could be intercepted by the attacker if the e-mail address of the user is included in the data that have been violated.

Bonus two

To perform this task I used the following online tool for the RSA encryption (https://www.devglan.com/online-tools/rsa-encryption-decryption). I have used the provided public key and the RSA cypher type. This is the encrypted text the tool has returned:
HLrXmhFIdlwNwRFfiXghpvRucFMponE3gLSIRl3epauTGMYZu8NN5rSIFVG3a95Il0GEgrMahgFgIuWcirb5CgR2cpPahxY6LvVkI14+o9VFCDCH/7FrFoQIitTMOSx3HqNnzJBO2OS8jx6foDbpCy2+SyJkRu1YO1+d+DsgB48=

Bonus three

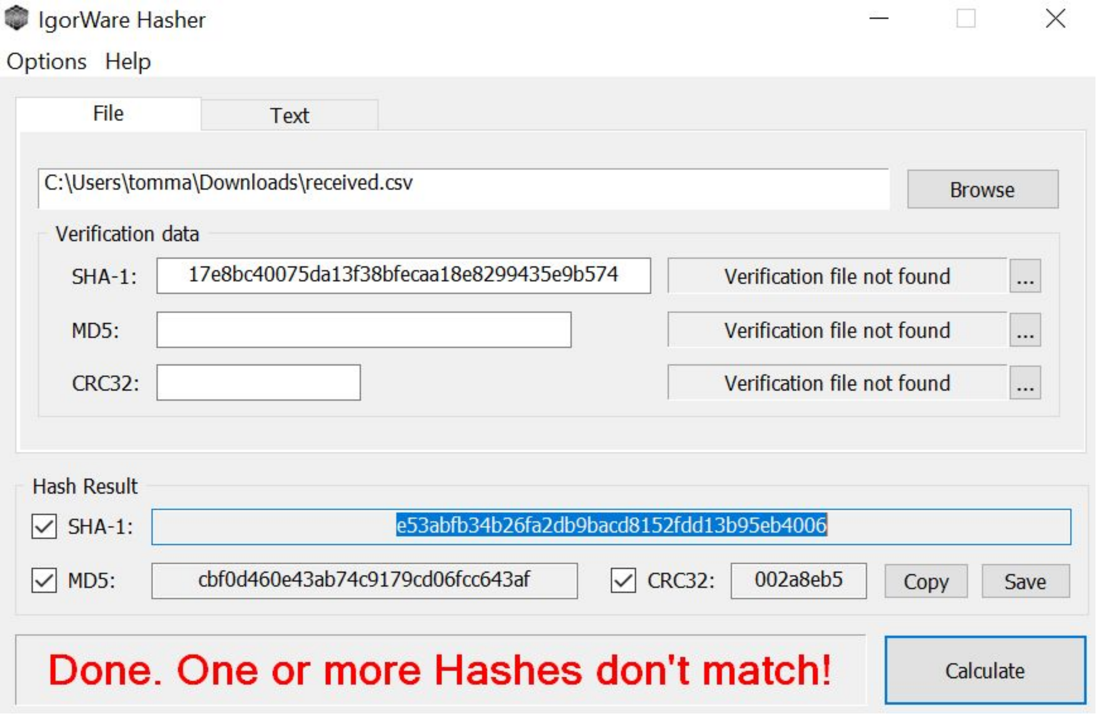
To perform the integrity check I used the following procedure:

1. I have applied the SHA-1 cryptographic hash function at the sent document;
2. I have applied the SHA-1 cryptographic hash function at the received document;
3. I have compared the two hashes values returned by the SHA-1.

These are the results I have obtained:

1. Sent document: 17e8bc40075da13f38bfecaa18e8299435e9b574
2. Received document: e53abfb34b26fa2db9bacd8152fdd13b95eb4006
3. The two hashes values are different and this means that the file may have been infected by malicious software.

To do this exercise I used the following tool: **IgorWare Hasher**. The following image is a screenshot of the application.



C:\Users\tomma\Downloads\received.csv

SHA-1: 17e8bc40075da13f38bfecaa18e8299435e9b574 — Verification file not found

MD5: — Verification file not found

CRC32: — Verification file not found

☑ SHA-1: e53abfb34b26fa2db9bacd8152fdd13b95eb4006

☑ MD5: cbf0d460e43ab74c9179cd06fcc643af ☑ CRC32: 002a8eb5

Done. One or more Hashes don't match!