Majeure Informatique
Security, Trust & privacy Course final report.

Due January 14, 2021

This work is a individual work. If you want or if you need you can exchange on the subject with your classmate, but at the end, the report must be written with your own words.
Any documents, information from any sources is allowed.
You must provide this work as a single pdf file and deposit it on campus before January 14, 2021.

Covid-19 pandemic and proximity tracing.

Introduction

COVID-19 apps are mobile software applications for digital contact tracing during the COVID-19 pandemic, i.e. the process of identifying persons ("contacts") who may have been in contact with an infected individual.

Numerous applications have been developed or proposed, with official government support in some territories and jurisdictions. Several frameworks for building contact tracing apps have been developed. Privacy concerns have been raised, especially about systems that are based on tracking the geographical location of app users.

Less intrusive alternatives include the use of Bluetooth signals to log a user's proximity to other cellphones. On 10 April 2020, Google and Apple jointly announced that they would integrate functionality to support such Bluetooth-based apps directly into their Android and iOS operating systems.

Source: `https://en.wikipedia.org/wiki/COVID-19_apps`

The subject

The goal of this exam is to study the two main proposals in Europe for *contact tracing* protocols: Robert and DP-3T. For this purpose, you will find at the end of this documents some links presenting these protocols and related informations. Your first job will be to browse these documents to understand what is *contact tracing* and what are the different security and privacy concerns. After this first phase, you will have to provide answers to the questions below and produce a report (with your own words as indicated in the header) and post this report on the campus' page of the Security course.

Question 1

- Synthesize and draw all exchanges describe in protocol specification for the two proposal (Robert and DP-3T) using the BAN syntax (see annexe) in the three main use cases:
  - proximity contact recording;
  - exposure verification;
  - infection declaration.

Question 2

- Explain why people say DP-3T is decentralised protocol an not Robert?

Question 3

- What are main privacy goals expressed for each protocol?
- How these two protocols fulfills these goals?
- Point out the pros and the cons for each protocol?

Question 4

Imagine you have to implement a contract tracing system using one of the two protocol.

- Describe the global infrastructure (with required equipments placed) of your solution (take into account all different users populations: general population, administration, medical teams...)
- What are the main security points you should take into account in your solution and explain how you solve it.

Question 5

In the two protocols, a server is required at exposure verification and infection declaration.
Imagine you are the owner (administrator) of the server and be able to observe all communications with your server at any time.
Imagine you are a telcom operator and be able to to observe all communications, you are also able to localize cell phone.

- in these two situations, explain how the flows could be exploited, what the attacker could learn and what protocols' properties cannot be respected.

Question 6

Robert an DP-3T protocols use implicitly an access control mechanism.

- What is the implicit access control mechanism used?
- Explain your answer.

Links to contact tracing protocols

- Wikipedia
  `https://en.wikipedia.org/wiki/Digital_contact_tracing`
- Apple and Google
  `https://www.google.com/covid19/exposurenotifications/`
  `https://covid19.apple.com/contacttracing`
- Inria and Robert
  `https://gitlab.inria.fr/stopcovid19/accueil`
  `https://github.com/ROBERT-proximity-tracing/documents`
- DP-3T
  `https://github.com/DP-3T/documents`

BAN syntax

The specification protocol syntax proposed here is very closed from the one used by M. Burrows, M. Abadi et R. Needham in there parper intituled "A logic of authentication" (1989). It was subsequently adopted to this publication in many works related to cryptographic protocols and is a reference today.

This syntax is defined by the following grammar where `(exp)*` is 0 or more time `exp`, `(exp)+` is à least one time `exp` and `(exp)?` is 0 or 1 time `exp`:

```
messages   := (message (action)? )+
message    := label'.' id '->' id ':' tuple
label      := id
id         := non empty sequence of alphanumeric characters
tuple      := atom (',' atom)*
atom       := cipher | clearterm
clearterm  := id | apply | '(' (tuple)? ')'
cipher     := { tuple }clearterm
apply      := id'('tuple')'
action     := arbitrary text
```

The identifiers (n.t. `id`) are protocol variables of a primitive types or functions, like one-way (hash) functions or functions which associate keys to principal names for instance. A primitive is proposed for encryption, with the usual notation using brackets. However, special identifier can also be used for encryption, when more details are needed.

The identifiers can be declared before the protocol messages. The form of the type declaration of primitive or functional identifiers is free. In particular, in case of public key cryptography, a particular notation can be used to associate the public and private part of a keypair.

Exemple : Needham Schroeder protocol

```
A,B :       principal
Na,Nb :     nonce
Kpa,Kpb :   key
```

1. `A -> B :` $\{\texttt{Na,A}\}_{\texttt{Kpb}}$
2. `B -> A :` $\{\texttt{Na,Nb}\}_{\texttt{Kpa}}$
3. `A -> B :` $\{\texttt{Nb}\}_{\texttt{Kpb}}$

The protocole provided as exemple can be read as follow:

Declarations:

- `A` et `B` are the actors of the system;
- `Na` and `Nb` are *nonces* (numbers never used in the system);
- `Kpa` and `Kpb` are, respectively, `A` and `B` public keys. We assume that each actor has the corresponding private key.

Exchanges:

1. `A` send to `B` the `Na` nonce and its name (`A`) encrypted whit the `B`'s public key (`Kpb`);
2. `B` send to `A` the `Na` nonce she received and the `Nb` nonce encrypted whit the `A`'s public key (`Kpa`);
3. `A` send to `B` the `Nb` nonce she received , encrypted whit the `B`'s public key.