
Hub d'Innovation Epitech

# Cahier des charges : Projet de "Recode" d'applications de télécommunications.

## I. PROBLEMATIQUE

De nos jours, les télécommunications ont pris une importance démesurée dans notre quotidien. Nombreux sont ceux qui, inconsciemment, transmettent en clair des informations sensibles, susceptibles d'être interceptées...
En effet, si les cartes SIM des opérateurs téléphoniques proposent un système de chiffrement, ce dernier peut être désactivé selon les circonstances (antenne vétuste, mauvaise réception...) et faire transiter les données entièrement en clair. Il est également possible de forcer la transmission en clair via une antenne tierce et ainsi récupérer la totalité d'une communication.

## II. OBJECTIF

A la lumière de cette problématique, notre équipe se propose de concevoir des applications de téléphonie par défaut d'Androïd (SMS/MMS et téléphone), qui chiffreraient les contenus transmis par le réseau mobile.

Les utilisateurs de notre nouvelle application de SMS enverraient les contenus de leurs messages cryptés via PGP sur un serveur. L'URL correspondante au message serait envoyée via le réseau mobile (en clair) au destinataire. Ledit destinataire récupère le contenu sur le serveur et le déchiffre au moyen de la clé privée liée à la clé publique échangée au préalable.

## III. SPÉCIFICATIONS TECHNIQUES

### 1. Chiffrement et déchiffrement des données en PGP (via l'**API Bouncy Castle**)

Le PGP est un algorithme de chiffrement à clés asymétriques où chaque utilisateur dispose d'une clé privée A (qu'il conserve) et d'une clé publique B (qui permet à n'importe qui de chiffrer un message qui ne sera déchiffrable que par quelqu'un ayant accès à la clé A). Le plus gros souci serait de trouver un moyen de transférer les clés publiques sans qu'elles ne soient altérées (volontairement ou non) ainsi que d'empêcher toute fuite de clé privée. Aussi, l'utilisateur peut possiblement signer son message pour prouver son identité au destinataire.

### 2. Transfert des clés de chiffrement publiques, possible

- via **NFC**, au contact
- En déposant sa clé publique sur le serveur d'échange, avec son numéro haché
  Dans ce cas, les échanges ne peuvent être faits qu'avec les membres enregistrés sur un même serveur. Pour reconnaître le propriétaire d'une clé, un identifiant (correspondant à son numéro de téléphone haché) y est associé. (optionnel)
- En proposant au lancement de l'application de chercher sur les serveur déjà existants les clés asssociées aux adresses mail des contacts.
  mesage chiffré pour le destinataire + 1 backup pour soi (avec clé privée)
- En déposant sa clé publique sur les serveurs PGP (MIT)
- En envoyant sa clé publique par SMS formaté (plutôt **l'identifiant / le serveur**)

L'application doit pouvoir régénérer une clé PGP une fois qu'un certain nombre de messages ont été envoyés au moyen de cette clé.
L'application pourrait aussi générer des sub-keys au lieu de générer une nouvelle clé

### 3. Hébergement des messages (via l'**API**)

Une API de serveur sera créée dans le but de répondre aux besoins de chaque utilisation.

Chaque utilisateur peut envoyer ses clés publiques sur un serveur de transfert défini. De plus, les messages envoyés par les utilisateurs de l'application sont stockés chiffrés sur ce même serveur.

Le chiffrement se fait au niveau du téléphone et non pas dans le serveur.

Un utilisateur peut spécifier un serveur autre que celui par défaut du service (et par exemple, héberger son réseau de communication sécurisée).

Lorsqu'un message a bien été délivré le serveur fait un feedback à l'app envoyeur pour le lui signaler. Si sur une certaine durée de temps l'application n'a pas recu le feedback du serveur, le message est tout simplement considéré comme non-reçu.

## IV. BACKLOG (SMS/MMS)

- En tant qu'utilisateur je dois pouvoir créer un nouveau message à partir de mes contacts existants dans mon téléphone.

- En tant qu'utilisateur, je dois pouvoir désactiver le chiffrement pour envoyer des messages a une personne en particulier.

- En tant qu'utilisateur je dois pouvoir voir les différents messages d'une conversation et différencier s'ils sont chiffrés ou non (code couleur?)

- En tant qu'utilisateur je peux changer la couleur de l'interface

- En tant qu'utilisateur, je peux échanger ma clé publique de différentes façons : NFC, serveur PGP ou serveur d'application, ou encore par SMS

- En tant qu'utilisateur, je peux accéder via la conversation à une fiche de contact locale à l'app, qui permet d'appeler (redirection app appel sécu?), bloquer le contact ou désactiver le chiffrement.

- En tant qu'utilisateur, je dois être informé lorsque l'un de mes messages chiffré ne peut être lu par le destinataire.

- En tant qu'utilisateur je dois pouvoir choisir le serveur par lequel mes données transitent. Je dois aussi pouvoir paramétrer mon propre serveur de communication au moyen de l'API du service.

## V. BACKLOG (API server)

- En tant qu'utilisateur conscient de ma sécurité, je dois pouvoir me créer un serveur d'envoi/réception de messages.

- En tant qu'utilisateur, le paramétrage d'un serveur ne doit pas être fastidieux.

- En tant qu'utilisateur, je dois pouvoir renseigner plusieurs serveurs et choisir via lequel/lesquels mes messages sont envoyés.

- En tant qu'utilisateur, je dois pouvoir être averti par mon application lorsque mon/mes serveurs ne sont plus joignables.

- En tant qu'utilisateur, je dois pouvoir consulter les status de mes messages : envoyé au serveur, récupéré par le destinataire/en attente, supprimé car non transmis au bout d'une certaine période, etc...

- En tant qu'utilisateur, je dois pouvoir configurer mon serveur à tout moment : durée à partir de laquelle un message non récupéré est supprimé, nombre maximal de messages en attente, numéros blacklistés/whitelistés...