Encrypto

Eine verschlüsselte Messenger-Applikation die Simplizität und Sicherheit zusammen ermöglicht.

Anforderungen

- Eine moderne Linux oder BSD distro mit Vernetzung
- Xorg server mit GTK+
- Libgcrypt
- Pthread

Installation

Um die Applikation zu installieren braucht man die source:

```
git clone https://github.com/leventeBajczi/encrypto-c.git
cd encrypto-c
make clean
```

Es gibt mehrere Möglichkeiten von hier an:

- Nur in diesem Folder die binäre Dateien herstellen:

  ```
  make encrypto
  ```

  Nach der Kompilierung kann eine neue Datei heißt `encrypto` in der `build` Folder gefunden werden.
- Nur in diesem Folder die binäre Dateien herstellen, und eine Server-Einheit laufen lassen:

  ```
  make server
  ```

  Nach der Kompilierung kann eine neue Datei heißt `encrypto-server` in der `build` Folder gefunden werden, der auch laufen gelassen wird.
- Binäre Datei auch in /usr/bin/ herstellen, damit die Applikation systemweit benutzbar wird:

  ```
  make install
  ```

  Nach der Kompilierung kann eine neue Datei heißt `encrypto` in der `build` Folder gefunden werden, der ins /usr/bin/ auch kopiert wird.

Synopsys

Die Applikation kann durch CLI-Paramatern gestartet werden, welche je eine standarde Wert haben (falls nicht eingegeben):

```
encrypto [-m mode] [-c client] [-k keys] [-d directory] [-s server] [-n name] [-p port]
```

Schaltern

- **Mode**
  - Entweder `server` oder `client` (default).

- In `server` Modus startet die Applikation sich als eine Server - Clients können sich zu dieser Einheit verbinden, Daten dadurch senden, und jeder, der sich über die Sicherheit des Programmes interessiert, kann die Logs checken. Nach der Anfang schreibt der Server zwei Zeile mit jede einer Portnummer aus: diese sind die Verbindungsporten, wozu Clients sich verbinden können. Zum Beispiel:

  ```
  Info available on port 45317
  Listening on port: 47709
  ```

  Diese beduetet, dass Clients können zu Port 47709 sich verbinden, und dass die 'Administration-Page' kann beim http://serverip:45317 gefunden.

  - Jeder Chat-Room braucht eine Server-Einheit. Diese sollte an inem solchen Maschine laufen gelassen werden, deren IP-Adresse bekannt und sichtbar (für die Clients) ist. Wenn ma dafür eine WAN benutzen möchtet, dann sollte er die Problem lösen, dass normalerweise die lokale IP-s sind nicht von außer sichtbar. In Allgeimein es beduetet, dass eine Weise Port-forwarding wird benötigt werden.

- In `client` Modus verbindet sich die Applikation zu einem Server, und fängt an Messages zu senden. Vorsicht - dafür braucht man einen Partner, wegen der Verschlüsselung!

- **Client**
  - Entweder `gui` oder `cli` (default).
  - In `gui` Modus hat man eine auf GTK basierte graphische Oberfläche. Es verhindert den Benutz des Applikations nicht - jeder Operation kann auch von dem GUI durchgeführt werden.

- **Keys**
  - Die Folder spezifieren, wo die Keyfiles gespeichert werden können.
  - Die standarde Folder dafür ist `keys/` in dem aktuellen Folder.
  - Vorsicht - diese Folder muss existieren!

- **Directory**
  - Die Folder spezifieren, wo die Dateien gespeichert werden können.
  - Die standarde Folder dafür ist `messages/` in dem aktuellen Folder.
  - Vorsicht - diese Folder muss existieren!

- **Server**
  - Die IP-Addresse des Servers spezifieren.
  - Die standarde Addresse ist `localhost` (`127.0.0.1`)

- **Name**
  - Die Benutzername, die den Benutzer kennzeichnet.
  - Die standarde name ist `Anon`

- **Port**
  - Die Port spezifieren, die die Applikation benutzen sollten.
  - Diese Port muss leer sein!
  - Die standarde Port ist `0`, was bedeutet, dass die Kernel einen beliebigen Wert zuordnen wird in `server` Modus.

- In `client` Modus muss diesen Wert spezifiert werden!

Beispiel für der Verwendung

Eine Beispiel, der ermöglicht voll verschlüsselte Kommunikation für zwei Leute, die nur verschiedene LANs erreichen.

1. Server

1. Eine der Leuten hat eine LAN, der Port-forwarding erlaubt.
    1. Port 50505 aus dem WAN nach Port 50505 von einem Maschine des LANs forwarden.
    2. An diesem Maschine die folgende Schritte laufen lassen:

        ```
        git clone https://github.com/leventeBajczi/encrypto-c.git
        cd encrypto-c
        make clean
        make install
        mkdir log/
        encrypto -m server -d log -p 50505
        ```

    3. Die Ausgangszeilen interpretieren:

        ```
        Info available on port 45317
        Listening on port: 47709
        ```

    4. Die Programm, den Server und der Router (und Switch, usw...) laufen bleiben lassen.

2. Client an dem gleichen LAN

1. An diesem Maschine die folgende Schritte laufen lassen:

    ```
    git clone https://github.com/leventeBajczi/encrypto-c.git
    cd encrypto-c
    make clean
    make install
    mkdir log
    mkdir keys
    encrypto -c gui -d log/ -p 50505 -k keys/ -n Name1 -s <lokales IP des Servers>
    ```

2. Der Terminal, in denen diese Schritte laufen gelassen wurden, fängt an Verschlüsselungsschlüsseln zu generieren. Für die sichere Speicherung des Private-Keys fragt die Applikation für einen Geheim, dessen Hash als Schlüssel benutzt wird.
3. Nach der Angabe dieser Geheims startet eine GUI, und von diesen Punkt die Verwendung ist ganz trivial - oben kann man die Daten des Programmes sehen, und unten kann neue Messages schicken.

3. Client nicht an dem gleichen LAN

1. An diesem Maschine die folgende Schritte laufen lassen:

    ```
    git clone https://github.com/leventeBajczi/encrypto-c.git
    cd encrypto-c
    make clean
    make install
    mkdir log
    mkdir keys
    encrypto -c gui -d log/ -p 50505 -k keys/ -n Name2 -s <öffene IP des Routers von dem Server>
    ```

2. Der Terminal, in denen diese Schritte laufen gelassen wurden, fängt an Verschlüsselungsschlüsseln zu generieren. Für die sichere Speicherung des Private-Keys fragt die Applikation für einen Geheim, dessen Hash als Schlüssel benutzt wird.

3. Nach der Angabe dieser Geheims startet eine GUI, und von diesen Punkt die Verwendung ist ganz trivial - oben kann man die Daten des Programmes sehen, und unten kann neue Messages schicken.

4. Beide Clients

1. Nach Schlüsselaustausch fangen die Clients an, Messages zu schicken. Diese Messages sind verschlüsselt mit AES256 - die öffene Daten kann man auf der Seite http://serverip:infoport checken - natürlich nur der Client am gleichen LAN sieht diese Information normalerweise - eine neues Port braucht geforwardiert werden, um die Andere Sichtbarkeit zu ermöglichen.