Home

Jump to bottom

jim-b edited this page on 1 May 2015 · 45 revisions

Mikey-Sakke KMS

#1. Project overview This work was part of a personal project to try and understand how the cryptography, often referred to as Mikey Sakke, described in RFCs 6507, 6508 and (parts of) 6509 worked. I had a few personal goals for this development:

- Produce a simple (as possible), easy to build, 'C' implementation using OpenSSL for the maths.
- To make it as clear as possible, by reference out to the RFCs (6507-6509), where specific calculations were described and expected results were indicated, so that these could be checked i.e. A learning exercise/ tool, useful to others trying to understand what is going on internally.

As part of my ECCSI-SAKKE Mikey-Sakke investigations I also needed to be able to create my own key material, rather than rely on the RFC values alone. That work is this project with a (incredibly simple) text driven menu on top.

This menu code is **not robust** and can be broken quite easily. It is just there as a simple example, to show that the underlying Key generation works.

NOTE! To use the Mikey-Sakke Key Material generated by this project, please refer to my other project ECCSI-SAKKE and specifically es-demo-2.c and it's wiki.

1.1. Restrictions on use

None that I know of; You can use this code for free following the Apache 2.0 License, including the *kudos clause* provided by the NOTICE text file.

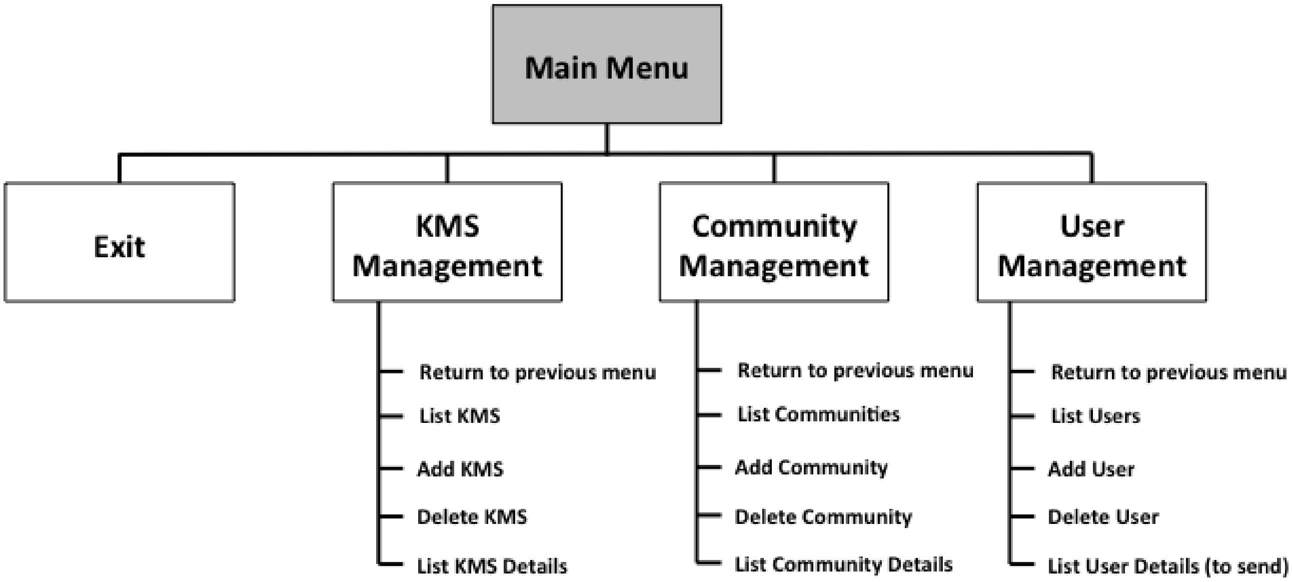
I'm not of entirely up to speed on licensing issues, if something's wrong or I've stepped on anyone else's toes, let me know and I'll put it right.

1.2. Mikey-Sakke Overview

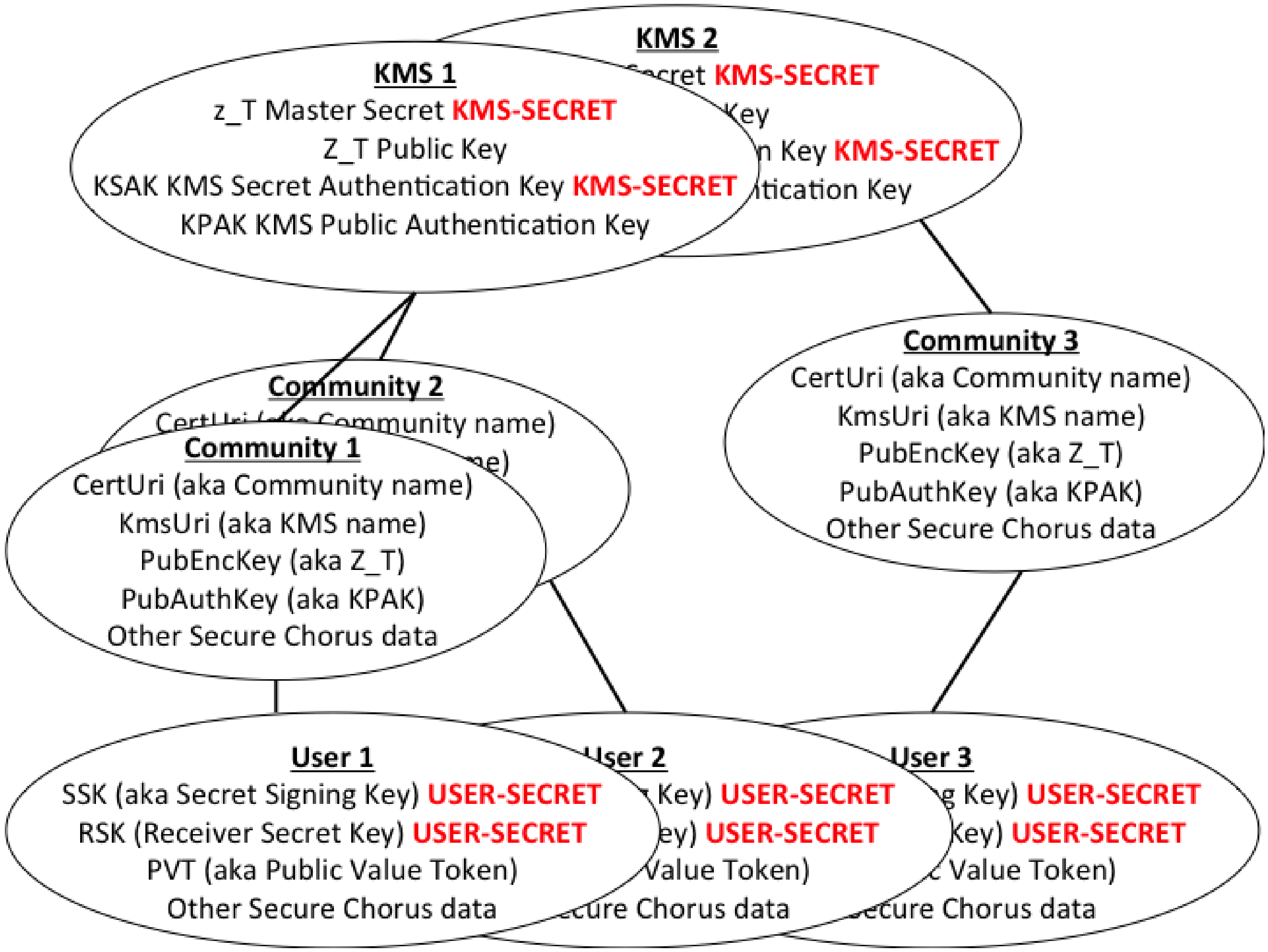
As this project **requires** the ECCSI-SAKKE project to work and this has a wiki covering Mikey-Sakke and how that project is constructed and works, I would suggest a quick read of that here.

2. KMS Menu Structure

The KMS provides a simple text based menu system as shown below:



The code has the ability to store multiple KMS, Community and Users. Note! This implementation allows each KMS to have multiple Communities and these communities to have (obviously) multiple Users as shown here:



3. Building the project

Building on Linux. Development machine was CentOS 6.

This project makes use of the ECCSI-SAKKE crypto library files that can be cloned by:

```
git clone https://github.com/jim-b/ECCSI-SAKKE.git
```

and then made following the instructions here.

Once that (ECCSI-SAKKE) project has been made we revert to making this (KMS) project.

We need to tell the make script where it can find the ECCSI-SAKKE crypto library files. To do this you modify the **ECCSI_SAKKE_DIR** attribute in the *make-kms* file, for example:

```
ECCSI_SAKKE_DIR=/home/_myname_/ECCSI-SAKKE
```

Next, we need to make the *make-kms* script executable, so you need to:

```
chmod 775 make-kms
```

Then to make, run the simple make script:

```
./make-kms
```

Finally, in order to run the KMS program you will need to make the libraries location accessible. The easiest way to do this on linux systems is to add the libraries location to LD_LIBRARY_PATH as follows (using the same data as above):

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/_myname_/ECCSI-SAKKE/lib
```

To run the program:

```
./kms
```

3.1. Display options

The ECCSI-SAKKE crypto library by default displays lots of DEBUG information, that links out to the relevant RFCs. I **strongly** suggest you turn this debug off, unless you want to see the detailed calculation described. To turn of the debug you should comment out the following line:

```
#define ES_OUTPUT_DEBUG
```

from:

```
src/utils/log.h
```

in **the ECCSI-SAKKE project** and rebuild it. Then rebuild this (KMS) project.

As is indicated in the previous Directory/ File Layout section, *community*, *user* and *kms* information is stored in a directory named *storage*. If you want to have these files stored in another directory at some point in the future, you should modify:

```
STORAGE_ROOT
```

in

```
inc/globals.h
```

in **the ECCSI-SAKKE project** and rebuild it. Then rebuild this (KMS) project again.

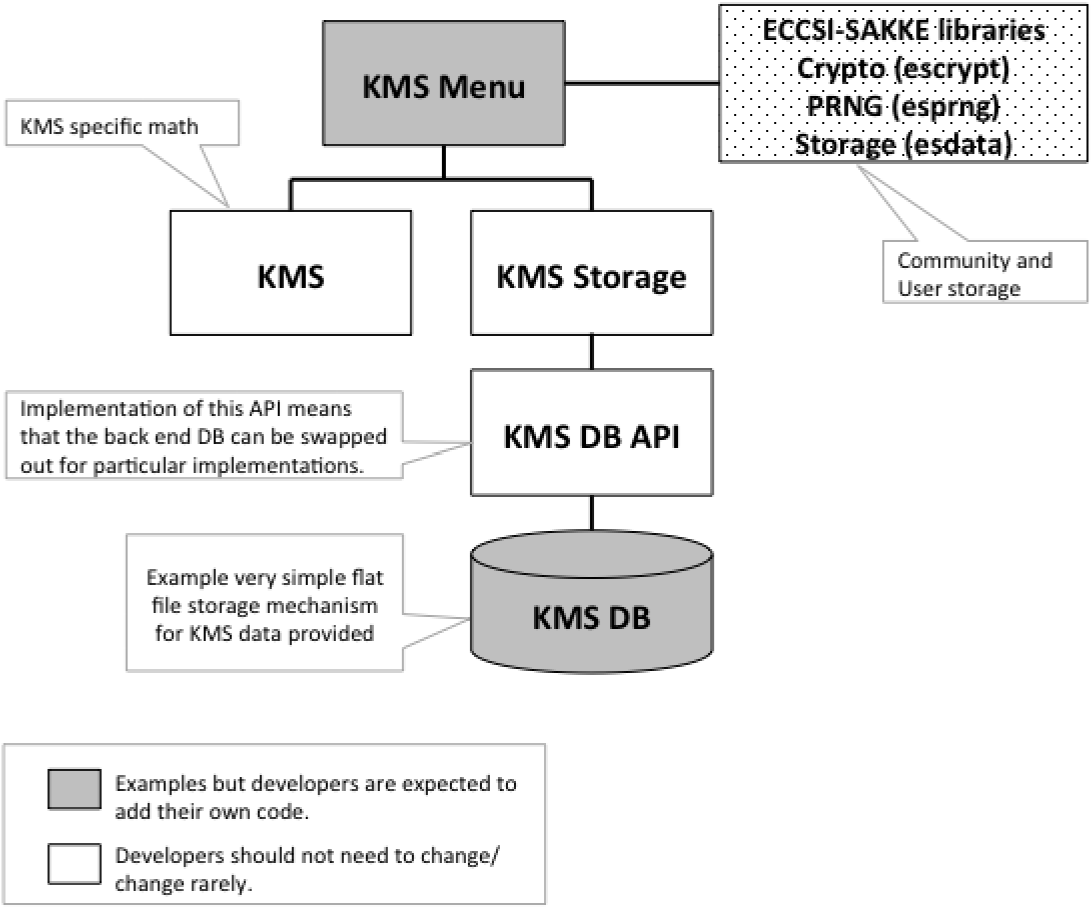
4. Architecture

This project uses the ECCSI-SAKKE project, not only for some required cryptographic functions, but also:

- *Community* and *User* data storage.
- Pseudo Random Number Generation.
- Log handling.

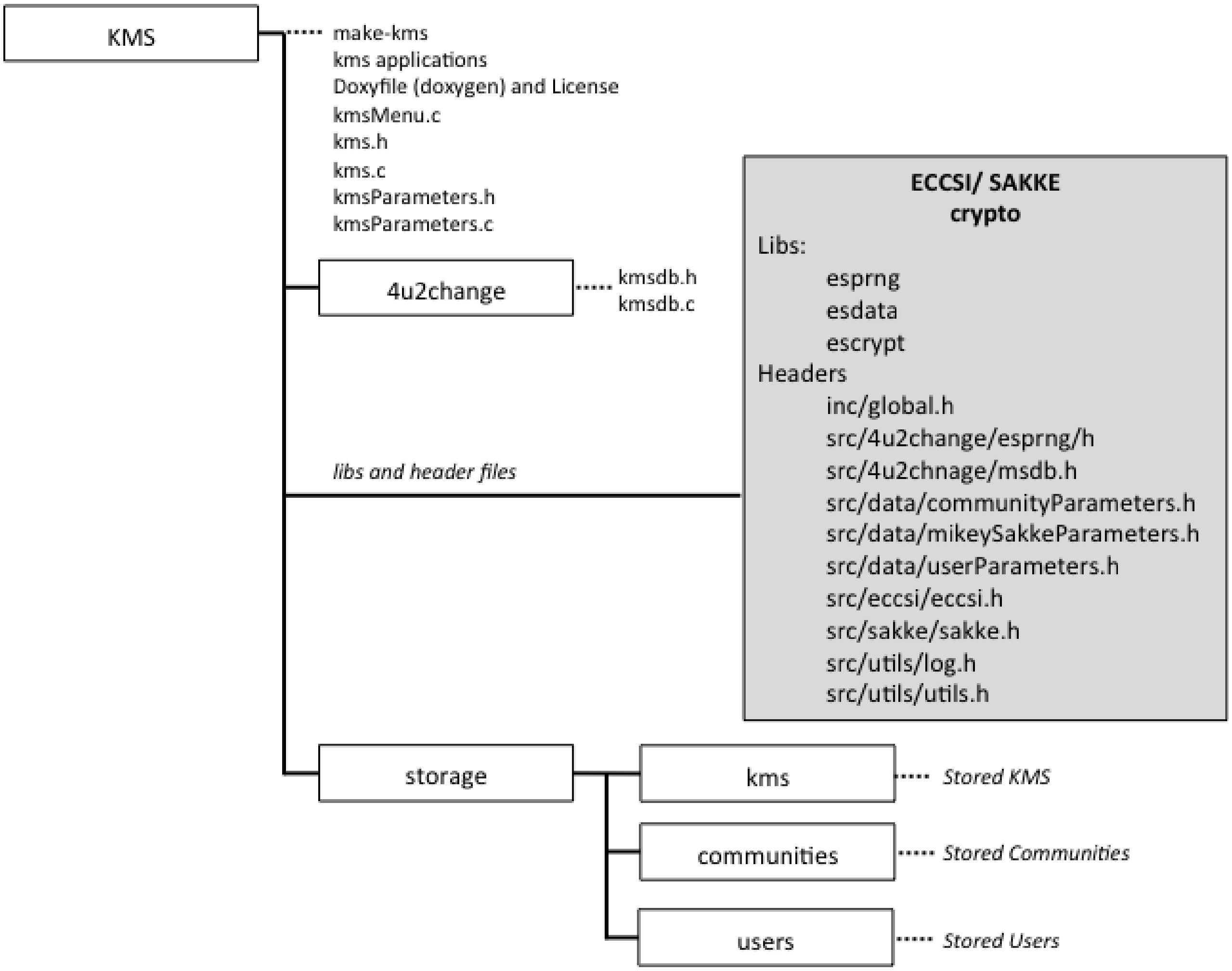
4.1. Software structure

The software structure follows a similar (though simpler) model to that implemented for the ECSSI/ SAKKE project and is shown below.


KMS Menu
ECCSI-SAKKE libraries
Crypto (escrypt)
PRNG (esprng)
Storage (esdata)
KMS specific math
KMS
KMS Storage
Community and User storage
Implementation of this API means that the back end DB can be swapped out for particular implementations.
KMS DB API
Example very simple flat file storage mechanism for KMS data provided
KMS DB
Examples but developers are expected to add their own code.
Developers should not need to change/ change rarely.

4.2. Directory/ File Layout

The following diagram shows the directory/ file layout structure with links out to the ECCSI-SAKKE libraries.



5. Using the KMS

This KMS project implements a simple text driven menu to show how the calls needed to create KMS, Community and User data. You can use the kmsMenu.c code to see how calls are made to the underlying KMS code to generate and manage Key Material (list-items, add, delete, list-details-for). Specifically, for:

- kms
- community
- user

5.1. Creating a KMS

Note! z_T and KSAK data in the generated KMS file **MUST** remain secure on the KMS otherwise security is compromised!

```
Main Menu
=========
    0 Exit
    1 KMS Management
    2 Community Management
    3 User Management
Selection: 1

KMS Menu
--------
    0 - Return to previous menu
    1 - List KMS
    2 - Add KMS
    3 - Delete KMS
    4 - List KMS Details
Selection: 2
    Enter name of new KMS: kms.mikey-sakke.org
    Enter a version for these KMS data (optional): 1.0
    Enter an owner for these KMS data (optional): Jim Buller
    Do you want to use RFC values, rather than random? (y/n): n
KMS save successful

KMS Menu
--------
    0 - Return to previous menu
    1 - List KMS
    2 - Add KMS
    3 - Delete KMS
    4 - List KMS Details
Selection: 0
```

Note! The KMS data is stored in file:

```
<your-storage-directory>/kms/kms.mikey-sakke.org
```

5.2. Creating a Community

```
Community Menu
--------------
    0 - Return to previous menu
    1 - List Communities
    2 - Add Community
    3 - Delete Community
    4 - List Community Details
Selection: 2
```

```
    Name of new community     : community.mikey-sakke.org
    Version (optional)        : 1.0
    Select KmsUri             :
        1 - kms.mikey-sakke.org
        2 - aliceandbob.co.uk
    Selection: 1
    Issuer (optional)         :
    Valid From (optional)     :
    Valid To (optional)       :
    Revoked (y|n default 'N') :
    User ID Format (optional) :
    KMS domain list (optional):
Community save successful

Community Menu
--------------
    0 - Return to previous menu
    1 - List Communities
    2 - Add Community
    3 - Delete Community
    4 - List Community Details
Selection: 0
```

Note! The Community data is stored in file:

```
<your-storage-directory>/community/community.mikey-sakke.org
```

5.3. Creating a User

```
Main Menu
=========
    0 Exit
    1 KMS Management
    2 Community Management
    3 User Management
Selection: 3

UserMenu
--------
    0 - Return to previous menu
    1 - List Users
    2 - Add User
    3 - Delete User
    4 - List User Details (to send)
Selection: 2
    Note! entries are not validated.
    Enter date validity for new User (YYYY-MM): 2015-04
    Enter identify of new User (e.g. tel:+number): tel:+441111111111
    Which community is the new user part of:
        1 - community.mikey-sakke.org
        2 - aliceandbob.co.uk
    Selection: 1
    Do you want to use RFC values, rather than random? (y/n): n
User save successful

Community Menu
--------------
    0 - Return to previous menu
    1 - List Communities
    2 - Add Community
    3 - Delete Community
    4 - List Community Details
Selection: 0
```

Note! The User data is stored in file:

```
<your-storage-directory>/users/2015-04 tel:+441111111111 community.mikey-sakke.org
```

5.4. Display Generated Data

At this point we have generated *KMS*, *Community* and *User* Key Material and created files for these data. The contents of the *Community* and *User* files need to be relayed **securely** to the Client in order to perform Mikey-Sakke crypto.

You *could* simply just copy these files to the (*storage/community* and *storage/user*) storage directories, if using the ECCSI-SAKKE project.

Instead, if you want to see the data listed you can do the following from the KMS menu.

```
UserMenu
--------
    0 - Return to previous menu
    1 - List Users
    2 - Add User
    3 - Delete User
    4 - List User Details (to send)
Selection: 4

User to list details for:
-------------------------
        1 - 2015-04 tek:+442222222222 community.mikey-sakke.org
        2 - 2015-04 tel:+441111111111 community.mikey-sakke.org
        3 - 2011-02 tel:+447700900123 aliceandbob.co.uk
    Selection: 2

++++++++++++++++++++++++++++++++++++++++++++++++++
The following data needs to be relayed (SECURELY!)
to the client (perhaps using Secure Chorus message
structures), but it's up to you to pick your
preferred protocol.

If you are just using this, and the ECCSI-SAKKE
code, you can either modify the USER and COMMUNITY
data creation examples provided in the ECCSI-SAKKE
project 'es-demo-nnn.c' file, or, just create the
file and add the content as indicated below.

NOTE! If you following the 'modify es-demo-nnn.c'
route, for adding this data to the client(s) demo,
you will NOT need to include HASH (i.e. only add
SSK, RSK and PVT) for the user data. This is
because the HASH will calculated and added to the
file when it is created.

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Create User filename:
    <your-storage-dir>/storage/users/2015-04 tel:+441111111111 community.mikey-
sakke.org
With the following content:

SSK:
    EF78C601 CE73934C BFA1D578 E34E4E3E
    3FCBAA8F DC8EE0B4 36DD34E9 71DD72D7

RSK:
    041BB681 A2008151 EE7A788A ADBFA170
    8745FE05 94987AA2 D4111410 A98DDA8A
    78DA6767 4AD22533 ADC20490 1D72DBF2
    0A01384F FB7799A3 18E4160A 34352A1B
    66575EB7 53F14C2D 2292608A 38344650
    AA252745 2CC29A1A C66027D7 19714C38
    AE5601E3 035B4B93 7C3B8B8E 1C1FDEAB
    6D466FE1 5E3A65E0 572F0912 D0E9CD44
    B60D812D 1EAA9B30 370394ED 0791EDEE
    34B151C8 92802B6E 65750F28 DEFE0028
    46B2DA19 D052B6CC F4F3C6ED C24722A4
    FF4F1F36 59ECED52 BE3592E8 18A04A99
    598CCE1A EA56694B 75B4665D 58D3F91E
    3A2F613A B6D3D80E 5FB5D090 0988E072
    04CA5D33 CE29829D 7F1A72A8 BD0D8F2F
    DF2C26BB 34DD25E3 B3015490 34A23F4F
    5B
```

```
HASH:
    CF1B73B8 1A4E4705 974EB91F 70EEE863
    0025584C B4F10A51 038A8614 1A1608EF

PVT:
    041D2EFB DDBB4E00 C31F7CD8 6FFA6CA3
    BDD1F0C9 93C113FB 2A10622C FA8328BB
    DAF14480 2672D0CC EF9747E6 0EEAE222
    F68A92ED 815E523C 5CC045B8 06C1883E
    D0

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Create Community filename:
    <your-storage-dir>/storage/communities/community.mikey-sakke.org
With the following content:

Version:
    1.0

CertUri:
    community.mikey-sakke.org

KmsUri:
    kms.mikey-sakke.org

Revoked:
    FALSE

PubEncKey:
    0478475A 19DB9038 50E4402D 01629185
    B58971DB CCB08CA2 7AECEE50 DFA2C981
    DFEF96CA AB8F449C CADC0966 7FC5AC0C
    28B335CC 7D18A013 5482E97C 8E38FA40
    0C517734 7CC4E7B1 128A7015 EFF23788
    77CF4BD2 BBAF911A DD63D9FC 56018134
    B3D83330 D12C981A A1955951 CCF4F55A
    231D69C8 3EE82D92 8EC0DFE7 80237C90
    D93414D5 0EEC3F11 99CD9B06 1C477CDA
    717AD07F 152EF956 ADD52C2F 2FE3B66D
    862040D6 9D10E6E6 4F095A57 E5AB1261
    541DF307 B965243C 0F053420 A92007D5
    21B83F3C 91F290E8 BDB0BE03 BF6B404A
    3539D030 A4438B82 244099DF 90F74332
    B8058462 A37FF4DD 5FC73253 A2892FC6
    7B08205B D87EE768 6ED7C67F B801F3A8
    C0

PubAuthKey:
    04253587 2D7931C6 891210FF 2CDFFD06
    FD464107 AC5F819E 5EACC8AC D4BBA806
    72C813BA 18955F4E A37D9BE3 DD9FAFED
    38BD1BF9 A9DF42B8 FD3D52E1 C64FB62B
    E8

++++++++++++++++++++++++++++++++++++++++++++++++++

UserMenu
--------
    0 - Return to previous menu
    1 - List Users
    2 - Add User
    3 - Delete User
    4 - List User Details (to send)
Selection: 0
```

5.5. Using Generated Data

An example of how to use the generated *Community* and *User* data can be found in es-demo-2.c of the ECCSI-SAKKE project.

6. Doxygen

There is some limited doxygen documentation in the project. To access this you need to install doxygen.

On Linux, you need to install the following repositories (using yum, apt, whatever), or their equivalents. The following assumes CentOS 6:

- yum install graphviz
- yum install boost-graph
- yum install texlive
- yum install texlive-utils
- yum install doxygen

Then (from inside the top level directory, where you cloned this project):

```
doxygen Doxyfile
```

Next, open your web browser (on the same machine) and open file:

```
file://<path-to-this-dir>/doxygen_output/html/index.html
```

7. To do

Pretty much there I think.

- Peer review and testing please, anyone?
- There's some magic numbers to quash.
- There's still a few return values to check.

#8. Contact details jim<-AT->mikey-sakke.org

▸ **Pages** 1

Contents

Clone this wiki locally

```
https://github.com/jim-b/KMS.wiki.git
```