sysmocom

sysmocom - s.f.m.c. GmbH

sysmoUSIM User Manual

by Harald Welte

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
v1	June 2016	Initial	hw
v2	January 2017	Fix FID for EF.OPC; Add EF.MLNGC; Explain EF.OPC structure; Refer to ETSI TS for AID	hw
v3	January 2017	Fix name and FID for EF.AUTH; Fix DF path for EF.MLNGC and EF.AUTH	hw

Contents

1 Introduction

This manual describes the sysmoUSIM-SJS1, a state-of-the-art Java SIM/USIM card for authentication in cellular networks.

The target audience are operators of cellular networks (large and small) who use the sysmoUSIM-SJS1 in order to identify the subscribers to their network.

As an operator of a cellular network, having significant knowledge about SIM/USIM card operation and configuration is a key aspect of running a secure and safe cellular network.

A specific emphasis is given to cellular networks running on the Osmocom/OpenBSC protocol stack, as this is what the sysmoUSIM-SJS1 was specifically introduced for, and why sysmocom is selling it. However, there is nothing restricting the cards to use in networks based on Osmocom software.

Please note however, that unless you have a specific support contract with sysmocom on said configuration, sysmocom will not be able to help you with questions regarding the use of sysmoUSIM-SJS1, particularly configurations not described in this manual.

2 History

When the Open Source GSM network-side protocol stack implementation OpenBSC started in 2009, it created a new opportunity for interested individuals and organizations to operate small-scale private or public, local or regional cellular networks without the dependency to the classic vendors of cellular technology.

If you want to run such a network without security, you can technically do that with pretty much any SIM card of any other operator (though their contract terms might not permit that legally).

Once you want to use cryptographic authentication and/or encryption in such networks, you need to issue your own SIM cards.

Traditional suppliers of SIM cards only sell to commercial public GSM operators and deal in quantities millions or at least hundreds of thousands. Individual SIM cards might be available for R&D and testing, but they are super expensive.

Also, cards from the classic suppliers are pre-provisioned to a given operator profile at manufacturing time, and do not provide the customer to re-program them later.

To solve the problem, sysmocom started to sell a series of SIM cards since 2011. Those cards are

- sold in small quantities
- provided with the card-individual pre-programmed identities and keys
- customer-reprogrammable, i.e. the sysmocom customer can change IMSI, MSISDN, ICCID and keys of the card as needed

For the first couple of years, the sysmoSIM-GR1 and later sysmoSIM-GR2 were sold. Those cards are not documented here as they have no longer been for sale for quite some time.

In 2014, sysmocom introduced the sysmoUSIM-SJS1. Contrary to its SIM-only predecessors, this card was the first sysmocom USIM, which prepared them for use in 3G networks, offering mutual authentication as part of UMTS AKA.

Furthermore, the sysmoUSIM-SJS1 was the first Java SIM card available from sysmocom, enabling the users to develop and run their own Java card applets, to use remote file management to update the SIM card files, etc.

All the above features make the cards ideal for the following user groups:

- users of small-scale local or regional cellular networks, whether GSM, GPRS, EDGE, UMTS, HSPA or LTE. This includes OpenBTS, OsmoBTS, OpenBSC, OsmoSGSN and many other Free Software implementations, but also includes proprietary small networks
- researchers using small-scale private networks for security analysis of mobile phones
- researchers and developers interested in SIM card related security issues, particularly in terms of STK and OTA

3 About SIM / USIM Cards

The SIM (Subscriber Identity Module) contains the cryptographic identity of a subscriber in a cellular network.

3.1 SIM Cards

The GSM (2G) network first introduce the SIM and specified its properties in ETSI TS 11.11. Later 2G extensions like GPRS and EDGE/EGPRS used the same SIM to authenticate the subscriber to the network.

Next to the cryptographic subscriber identity, SIM cards can also store a variety of other configuration parameters as well as user data, such as:

- the operator name
- the MSISDN of the subscriber
- received and sent SMSs
- phone book data
- parameters related to SMS (SMSC, . . .)
- the most recently used cell ARFCN
- the most recently negotiated TMSI and Kc

sysmocom has been selling various types of SIM cards over the years, specifically the sysmoSIM-GR1 and sysmoSIM-GR2 cards, in all the different form factors

3.2 USIM Cards

The UMTS (3G) networks introduced the *USIM Application* which runs on top of an ETSI UICC. The *USIM Application* covers the UMTS specific parts, while the ETSI UICC covers general aspects of chip cards, irrespective of their specific application.

A USIM implements the functions required by UMTS Authentication and Key Agreement. The particular differentiators compared to the SIM are:

- mutual authentication, i.e. the USIM also authenticates the network
- replay protection by introduction of a sequence number

Most USIMs also implement the SIM card protocol for backwards compatibility, so they can be used in older GSM-only phones.

3.3 Authentication Algorithms

A GSM network can support any authentication algorithm, as long as that algorithm is implemented in the (U)SIM and the AUC. As those are both controlled by the home operator of the subscriber, the operator can freely choose any algorithm for authentication.

In practise, not every operator has both the cryptographic expertise and a market power significant enough to have SIM manufacturers implement their algorithm.

So a set of algorithms was designed by the GSMA, and subsequently used by many operators in their networks: the COMP128 family.

There are three version of COMP128: v1, v2 and v3. Only v3 is considered reasonably secure, while COMP128v1 has been publicly demonstrated to be broken already in 1997.

sysmoSIM-GR1, sysmoSIM-GR2 and sysmoUSIM-SJS1 all support the full set of COMP128v1, COMP128v2 and COMP128v3 algorithms. For the USIM products, this only applies when used in classic SIM protocol, and not via USIM protocol.

In USIMs, the situation is similar in that only the USIM and the AUC need to know the algorithm, and thus an operator can implement and deploy whatever they want.

However, in practise most networks seem to utilize the MILENAGE algorithm.

The sysmoUSIM-SJS1 implements the MILENAGE algorithm.

See also Section 8.5

3.3.1 OP or OPc in USIM

The 128-bit value OOP is the Operator Variant Algorithm Configuration field, which was included to provide separation between the functionality of the algorithms when used by different operators. It is left to each operator to select a value of OP.

The algorithm set is designed to be secure whether or not OP is publicly known; however, operators may see some advantage in keeping their value of OP secret as a secret OP is one more hurdle in an attacker's path.

The USIM can be configured to either store an OP value, or an OPc value. OPc is computed by XOR of OP and EK(OP).

So the operator and card-issuer has the choice to either:

- use one OP value all across his network, and store that value on each card, or
- pre-compute a card-specific OPc value, and store that individual OPc on each card

The latter choice (OPc on card) is generally considered more secure, as the reverse engineering of one OPc does not reveal any security parameters relevant beyond that single card.

The sysmoUSIM-SJS1 supports storing either the card-individual OPc as well as the global OP value and thus gives maximum flexibility to the user.

For more details on OP and OPc as well as the rationale for preferring OPc storage on the card, see Section 5.1 of 3GPP TS 35.206 [3gpp-ts-35-206] as well as Section 8.3 of 3GPP TS 35.205 [3gpp-ts-35-205].

3.4 Interoperability SIM/USIM vs. Network vs Phone

There are several permutations between GERAN-only, GERAN/UTRAN or UTRAN-only phones, classic GSM SIM, USIM-only and combined USIM/SIM cards as well as the respective UTRAN/GERAN networks and the associated AUCs.

The following paragraphs are intended to shed some light on the respective interoperability.

3.4.1 SIM Card vs. Network

Depending on operator configuration, a GSM SIM card may also be used on a UMTS UTRAN network. The authentication then is of course only one-way and not mutual. Also, the generated encryption and integrity keys are generated from the expanded shorter GSM keys, and thus considered less strong.

An USIM can always be used over a GSM/GERAN network. If the phone supports the USIM protocol, it will be able to use USIM AKA even over a GERAN network (if permitted/enabled by the network).

3.4.2 SIM Card vs. Phone

A UMTS/UTRAN capable phone will first try to use the inserted SIM in USIM protocol. If that is not available, it will fall back to use the classic GSM SIM card protocol.

A GSM-only phone will inter-operate only with USIM cards which feature the backwards-compatible GSM SIM card protocol.

sysmoUSIM-SJS1 support both GSM as well as USIM mode, and are pre-provisioned in a way that both modes are available.

3.5 Card Form Factors

images/GSM_Micro_SIM_Card_vs._GSM_Mini_Sim_Card_-_Break_Apart.svg

Figure 1: Card Form Factors by Cvdr based on Justin Ormont's work. CC BY-SA 3.0, via Wikimedia Commons

4 sysmoUSIM-SJS1 specifications

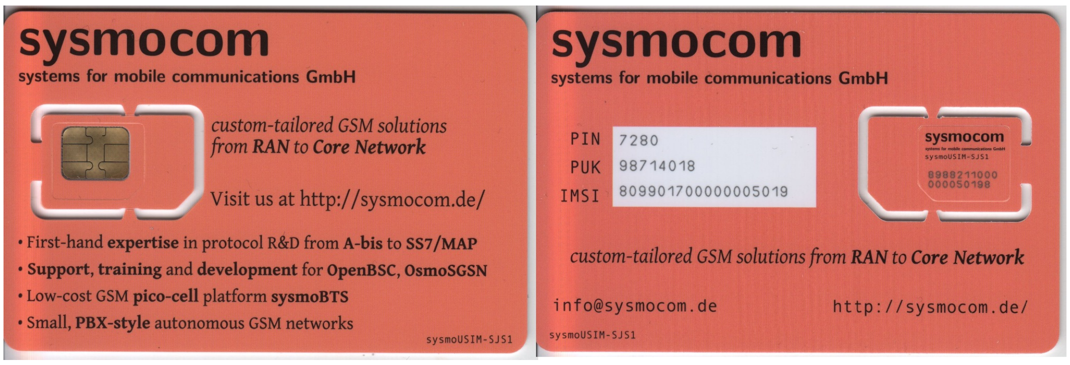
sysmocom
systems for mobile communications GmbH
custom-tailored GSM solutions from RAN to Core Network
Visit us at http://sysmocom.de/
• First-hand expertise in protocol R&D from A-bis to SS7/MAP
• Support, training and development for OpenBSC, OsmoSGSN
• Low-cost GSM pico-cell platform sysmoBTS
• Small, PBX-style autonomous GSM networks
sysmoUSIM-SJS1
PIN 7280
PUK 98714018
IMSI 809901700000005019
custom-tailored GSM solutions from RAN to Core Network
info@sysmocom.de
http://sysmocom.de/
sysmoUSIM-SJS1

Figure 2: sysmoUSIM-SJS1

The sysmoUSIM-SJS1 is a Java SIM card with the following specifications and features:

4.1 Physical Specification

- Available mechanical form factor
  - 2FF + 3FF (micro) SIM card
  - 2FF + 4FF (nano) SIM card
  - MFF (solder-type) chip
- 64kBytes user flash memory
  - 500.000 write cycles
- Temperature Range: -25 to +85 Centigrade chip temperature

4.2 Logical Specification

4.2.1 ETSI/3GPP/GP Specification Compliance

The sysmoUSIM-SJS1 adheres to the following specifications / spec versions:

- Combined SIM and USIM application
  - ETSI TS 102 221; [etsi-ts102221]
  - 3GPP TS 51.011 (R4); [3gpp-ts-51-011]
  - 3GPP TS 31.101 (R6); [3gpp-ts-31-101]

- 3GPP TS 31.102 (R6); [3gpp-ts-31-102]

- Java Card 2.2.1
- Global Platform 2.1.1
- Sim Toolkit Support
  - 3GPP TS 51.014 (R4); [3gpp-ts-51-014]
  - 3GPP TS 31.111 (R6); [3gpp-ts-31-111]
- OTA (Over-The-Air) Support
- Remote File Management / Remote App Management
  - 3GPP TS 23.048 (R4); [3gpp-ts-23-048]
  - 3GPP TS 31.115 (R6); [3gpp-ts-31-115]
  - 3GPP TS 31.116 (R6); [3gpp-ts-31-116]

4.2.2 Default Algorithms

- Default 2G Authentication Algorithm: COMP128v1
- Default 3G Authentication Algorithm: MILENAGE (3GPP TS 35.206; [3gpp-ts-35-206]

4.2.3 PIN Data

Each card has card-unique PINs pre-provisioned.

Code	Enabled	Length	Max Attempts
PIN1	No	4 digits	3
PIN2	Yes	4 digits	3
ADM1	Yes	8 bytes	3
ADM2	Yes	8 bytes	3
PUK1	Yes	8 bytes	10
PUK2	Yes	8 bytes	10

4.2.4 Network Authentication

- K (3G) == Ki (2G): card-individual 16 bytes
- OPc (3G): card-individual 16 bytes

NOTE
The cards are provided with card-individual OPc value. Sometimes operators chose to have a globally shared OP value, and OPc computed from OP+K. This is **not** what sysmocom provides with sysmoUSIM-SJS1 by default. See Section 3.3.1 for more details.

4.2.5 OTA Configuration

The following card-unique OTA keys are configured:

Code	Length	Key set
KIc	16 bytes	1, 2, 3
KID	16 bytes	1, 2, 3

4.2.6 Subscriber Identities

The following subscriber identities are pre-programmed into the sysmoUSIM-SJS1, unless the customer has specified different provisioning data at time of purchase (or changed the values after purchase):

Identity	Value
IMSI	90170xxxxxxxxxx
ACC	equal distribution
ICCID	8988211xxxxxxxxxxxx
MSISDN	88211xxxxxx

4.3 Product Variants

The sysmoUSIM-SJS1 is sold by sysmocom in the following different product variants, depending on your needs.

Variant	IMSI	K, OPc	ADM key	OTA key
default 901-70 MCC-MNC (no ADM keys)	90170. . .	random, provided	not provided	not provided
default 901-70 MCC-MNC (with ADM keys)	90170. . .	random, provided	provided	provided
user-defined IMSI/Ki (with ADM keys)	user-specified	user-specified	provided	provided

All three variants are available in quantities as low as 10-unit packs from the sysmocom webshop, in different physical sizes.

SKU	Form-Factor	Link to sysmocom webshop
sysmoUSIM-SJS1	1FF + 2FF + 3FF	http://shop.sysmocom.de/products/sysmousim-sjs1
sysmoUSIM-SJS1-4ff	1FF + 2FF + 4FF	http://shop.sysmocom.de/products/sysmousim-sjs1-4ff

MFF form-factor sysmoUSIM-SJS1 are available upon requeset on a per-project basis.

4.3.1 default 901-70 MCC-MNC (no ADM keys)

In this configuration you will receive the pre-provisioned SIM cards with identities (IMSI, ICCID, MSISDN) as outlined above, as well as key (K, OPc) data that is random and card-unique.

The identities and neither the key data can **not** be changed.

This variant is sufficient if you want a cost-efficient solution for subscriber identification but do not need any flexibility to re-configure your cards.

4.3.2 default 901-70 MCC-MNC (with ADM keys)

In this configuration you will receive the pre-provisioned SIM cards with identities (IMSI, ICCID, MSISDN) as outlined above, as well as key (K, OPc) data that is random and card-unique.

You will furthermore receive the ADM1 key, which can be used to fully change any identity or key data, as well as the content of any other file on the card.

The identities and keys **can** be changed by the customer.

You will also receice the OTA keys (KIC1 and KID1) which are needed for installation of Java applets on the card.

This variant is sufficient if you want a cost-efficient solution for subscriber identification but still want to have some flexibility to re-configure your cards.

4.3.3 user-defined IMSI/Ki (with ADM keys)

In this configuration, you will specify to sysmocom the IMSI/ICCID/MSISDN ranges to be pre-provisioned on the cards before shipping to you. The key (K, OPc) data can also be specified by you, or randomly generated.

You will furthermore receive the ADM1 key, which can be used to fully change any identity or key data, as well as the content of any other file on the card.

The identities and keys **can** be changed by the customer.

You will also receice the OTA keys (KIC1 and KID1) which are needed for installation of Java applets on the card.

This variant is the most flexible variant, and provides you to get cards tailored to your custom requirements, while still having flexibility to change any parameter yourself after purchase.

5 Smart Card Readers

SIM/UICC/USIM cards are smart cards compliant to the electrical parameters of ISO 7816-3, both in terms of voltage but also in terms of signal / timing. This is the same standard as used by many other smart cards, including all kinds of identification cards, debit/credit cards, cryptographic smart cards, etc.

In order to interface a SIM/USIM to a computer, you thus need a smart card interface device (colloquially called "card reader") compliant to ISO 7816-3.

In order to support maximum compatibility with software programs, the reader should inter-operate with the pcsc-lite software stack on your GNU/Linux based operating system.

The easiest type of readers in recent years have proven to be USB attached smart card readers compliant to the USB CCID specification.

Compliance to USB CCID ensures that a variety of vendor-neutral/independent drivers will work on virtually any operating system.

sysmocom offers a suitable USB CCID compliant card reader at http://shop.sysmocom.de/products/scr3310

Figure 3: SCR3310 USB CCID Smard Card Reader

5.1 Verifying your smart card reader + software stack

For details on how to configure your smart card reader / driver stack, please consult related documentation. In the case of USB CCID readers and pcsc-lite, any modern GNU/Linux distribution should have everything pre-configured without any manual intervention required.

In case of Ubuntu or Debian GNU/Linux, you only need to install the pcscd and libccid packages, e.g. using **apt-get install pcscd libccid**

5.2 Verifying your smart card reader + software stack

Every smart card returns a so-called ATR (Answer-To-Reset) as soon as it is first interrogated by the card reader.

You can use the **pcsc_scan** utility in order to read the status of your card reader and obtain the ATR of the currently-inserted card.

Example output of pcsc_scan with a sysmoUSIM-SJS1 inserted

```
$ pcsc_scan
PC/SC device scanner
V 1.4.26 (c) 2001-2011, Ludovic Rousseau <ludovic.rousseau@free.fr>
Compiled with PC/SC lite version: 1.8.15
Using reader plug'n play mechanism
Scanning present readers...
0: Alcor Micro AU9560 00 00

Sat May 21 21:38:31 2016
Reader 0: Alcor Micro AU9560 00 00
  Card state: Card inserted,
  ATR: 3B 9F 96 80 1F C7 80 31 A0 73 BE 21 13 67 43 20 07 18 00 00 01 A5
ATR: 3B 9F 96 80 1F C7 80 31 A0 73 BE 21 13 67 43 20 07 18 00 00 01 A5
+ TS = 3B --> Direct Convention
+ T0 = 9F, Y(1): 1001, K: 15 (historical bytes)
  TA(1) = 96 --> Fi=512, Di=32, 16 cycles/ETU
    250000 bits/s at 4 MHz, fMax for Fi = 5 MHz => 312500 bits/s
  TD(1) = 80 --> Y(i+1) = 1000, Protocol T = 0
-----
  TD(2) = 1F --> Y(i+1) = 0001, Protocol T = 15 - Global interface bytes following
-----
  TA(3) = C7 --> Clock stop: no preference - Class accepted by the card: (3G) A 5V B 3V C  ↩
      1.8V
+ Historical bytes: 80 31 A0 73 BE 21 13 67 43 20 07 18 00 00 01
  Category indicator byte: 80 (compact TLV data object)
    Tag: 3, len: 1 (card service data byte)
      Card service data byte: A0
        - Application selection: by full DF name
        - BER-TLV data objects available in EF.DIR
        - EF.DIR and EF.ATR access services: by GET RECORD(s) command
        - Card with MF
    Tag: 7, len: 3 (card capabilities)
      Selection methods: BE
        - DF selection by full DF name
        - DF selection by path
        - DF selection by file identifier
        - Implicit DF selection
        - Short EF identifier supported
        - Record number supported
      Data coding byte: 21
        - Behaviour of write functions: proprietary
        - Value 'FF' for the first byte of BER-TLV tag fields: invalid
        - Data unit in quartets: 2
      Command chaining, length fields and logical channels: 13
        - Logical channel number assignment: by the card
        - Maximum number of logical channels: 4
    Tag: 6, len: 7 (pre-issuing data)
      Data: 43 20 07 18 00 00 01
+ TCK = A5 (correct checksum)

Possibly identified card (using /home/laforge/.cache/smartcard_list.txt):
3B 9F 96 80 1F C7 80 31 A0 73 BE 21 13 67 43 20 07 18 00 00 01 A5
        sysmoUSIM-SJS1 (Telecommunication)
        http://www.sysmocom.de/products/sysmousim-sjs1-sim-usim
```

5.3 Mechanical Card Adapters

Smard card readers most often only are available for insertion of full-size (credit-card sized) smart cards.

Thus, you may need a mechanical adapter that converts the physical size of your SIM card to the full-sized card as supported by the smart card reader. The adapter is not required, if your SIM is still in full size (credit card size), but generally required if the card is already broken out and now has the 2FF, 3FF or 4FF form-factor

sysmocom offers a suitable low-cost adapter at http://shop.sysmocom.de/products/sim-adapter-pcb

6 Testing sysmoUSIM-SJS1 with utilities

There are some utilities that can be used to test the sysmoUSIM-SJS1 cards.

6.1 osmo-auc-gen

The **osmo-auc-gen** utility can be used to generate authentication triplets (GSM) or quintuples (UMTS AKA) from the secret key. It replicates the core operation that usually happens in the AUC component of the cellular network.

In order to use the tool to generate authentication triplets / quintuples, you need

- the secret key data (K, OPc) associated with the respective card for which the triplets / quintuples are to be generated
- the authentication algorithm to be used
- the osmo-auc-gen utility, part of git://git.osmocom.org/libosmocore / http://git.osmocom.org/libosmocore/

6.1.1 SYNOPSIS

osmo-auc-gen [-2|-3] [-a comp128v1|comp128v2|comp128v3|milenage] -k *KEY* -o *OPC* [-r *RAND*] [-f *AMF*] [-s *SQN*] [-A *AUTS*]

6.1.2 OPTIONS

-2
: Use 2G (GSM) Authentication

-3
: Use 3G (UMTS) Authentication

-a comp128v1|comp128v2|comp128v3|milenage
: Specify the algorithm to use for computing the authentication data

-k KEY
: Specify the secret key (Ki in case of GSM, K in case of UMTS) as 16 hex-encoded bytes

-o OPC
: Specify the secret OPC value (in case of UMTS AKA) as 16 hex-encoded bytes

-r RAND
: Optionally specify the random challenge as 16 hex-encoded bytes. If none is specified, a weak pseudo-random value is used. Don’t use this in practise.

-f AMF
: Optionally specify the AMF value for UMTS AKA

-s SQN
: Specify the UMTS AKA sequence number as integer. 0 is used as default.

-A AUTS
: Specify the UMTS AKA re-synchronization value AUTS, as received from the card

6.1.3 Example

The below example shows an **osmo-auc-gen** invocation using the given values for K, OPC and RAND.

```
$ osmo-auc-gen -3 -a milenage -k 1D8B2562B992549F20D0F42113EAA6FB -o 398153093661279 ↩
    FB1FC74BE07059FEF -r 000102030405060708090a0b0c0d0e0f -s 101
osmo-auc-gen (C) 2011-2012 by Harald Welte
This is FREE SOFTWARE with ABSOLUTELY NO WARRANTY

RAND:   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
AUTN:   e6 0a b2 d3 64 48 00 00 b8 32 f8 98 3c bb 39 c6
IK:     d5 9c 8e 92 93 bc 73 5e 62 39 24 47 a1 e6 58 8a
CK:     a8 8a 03 ff f8 2a 8a 26 e3 ea 43 d8 28 65 a7 25
RES:    dc 22 4d a2 03 51 d4 d1
SRES:   df 73 99 73
Kc:     fc c5 ea f2 e2 15 06 d7
```

The resulting values have the following meaning:

Parameter	**Gen**	**Transmitted to SIM**	**Meaning**
RAND	2G+3G	X	The random challenge used. Transmitted to (U)SIM
AUTN	3G	X	The authentication nonce
IK	3G	-	Integrity Protection Key
CK	3G	-	Ciphering Key
RES	3G	-	Authentication result; compared by MSC/SGSN with value received from card
SRES	2G	-	Authentication result; compared by MSC/SGSN with value received from card
Kc	2G	-	Ciphering key for GSM A5 / GPRS GEA encryption

6.2 osmo-sim-auth.py

The **osmo-sim-auth.py** utility can be used to perform authentication against a SIM/USIM card located in a smart card reader attached to your computer. It performs the exact same set of operations against the SIM/USIM card as a mobile phone would as part of a cellular network.

This permits you to test the authentication functions of your card without the complexity of running an entire cellular network.

In order to use this tool, you need

- a smart card reader supported by the pcsc-lite software stack as described in Section 5.
- a sysmoUSIM-SJS1 card which you would like to test
- optionally a mechanical adapter that converts the physical size of your SIM card to that of the smart card reader. This is not required, if your SIM is still in full size (credit card size), but generally required if the card is already broken out and now has the 2FF, 3FF or 4FF form-factor
- an authentication challenge to test the card with. This can e.g. be created by a prior call to osmo-auc-gen.
- the osmo-sim-auth.py script from git://git.osmocom.org/osmo-sim-auth / http://git.osmocom.org/osmo-sim-auth/

6.2.1 SYNOPSIS

osmo-sim-auth.py [-h] -r *RAND* [-d] [-s|-a *AUTN*]

6.2.2 OPTIONS

-h, --help
Print help message

-r, --rand RAND
Specify the random challenge (RAND) to be sent to the card as 16 hex-encoded bytes

-a, --autn AUTN
Specify the Authentication Nonce (AUTN) to be sent to card as 16 hex-encoded bytes. Must be specified in case of UMTS AKA authentication.

-d, --debug
Enable debug output

-s, --sim
Enable GSM SIM authentication mode (default mode is USIM)

6.2.3 Example

Using the AUTN and RAND parameters from the previous example of osmo-auc-gen, we can run the following example against the real card:

```
./osmo-sim-auth.py -a e60ab2d364480000b832f8983cbb39c6 -r 000102030405060708090a0b0c0d0e0f
[+] UICC AID found:
found [AID 1] 3GPP || USIM || (255, 255) || (255, 255) || (137, 7, 9,
0, 0)
[+] USIM AID selection succeeded

Testing USIM card with IMSI 901700000011000

UMTS Authentication
RES:    dc224da20351d4d1
CK:     a88a03fff82a8a26e3ea43d82865a725
IK:     d59c8e9293bc735e62392447a1e6588a
Kc:     fcc5eaf2e21506d7

GSM Authentication
SRES:   dc4ca85d
Kc:     6efa00fbbd41dc00
```

As we can see, the computed values by the card correspond to those values computed by the network. Thus, the authentication procedure is a success.

6.2.4 Re-Synchronization

If the SQN value on card-side and network-side are not in sync, **osmo-sim-auth.py** will not return RES/SERS, but instead return an AUTS value for re-synchronization.

This value then needs to be passed to **osmo-auc-gen** (-A parameter), which will then compute the current SQN value.

A SQN value higher than the one determined by the AUTS proecdure must be used as input to **osmo-auc-gen** to generate a new authentication quintuples (-s parameter). The SQN value has to be such one that at least causes a changes of bit 2^5 or bit 6. Please refer to 3GPP TS 33.102 Release 11, annex C. 3.2, "Management of sequence numbers which are not time-based", where the following parameter values are suggested for reference: Length of IND in bits = 5, Length of the array: a = 32. The last one relates to verification of sequence numbers in the USIM. Minimum value that satisfies the requirements is a value that is achieved by applying incremental step of 32.

7 Provisioning of different identities or keys

If you have a variant of the card-individual ADM1 key of your sysmoUSIM-SJS1 card, you can change any identity (IMSI, ICCID, MSISDN) stored on the (U)SIM, as well as the private key data (K, OPC).

In order to do so, you will need:

- a smart card reader supported by the pcsc-lite software stack on Linux. We recommend the use of a USB CCID compliant card reader.
- a sysmoUSIM-SJS1 card which you would like to modify
- optionally a mechanical adapter that converts the physical size of your SIM card to that of the smart card reader. This is not required, if your SIM is still in full size (credit card size), but generally required if the card is already broken out and now has the 2FF, 3FF or 4FF form-factor
- the ADM1 key for the card
- the **pySim-prog.py** program from git://git.osmocom.org/pysim / http://git.osmocom.org/pysim/
- the **sysmo-usim-tool** program from git://git.sysmocom.de/sysmo-usim-tool / http://git.sysmocom.de/sysmo-usim-tool

Note
In order to provision different identities and/or keys on your card, you need to purchase a variant of the card that provides the ADM keys to you (see Section 4.3.2 and Section 4.3.3, respectively).

7.1 pySim-prog.py

7.1.1 Example

In the below example, we are changing the card's IMSI to 901710000011000 (it was 901700000011000 before), and specify a new set of K and OPC values.

Full example of re-programming the card using pysim

```
$ ./pySim-prog.py -p 0 -t sysmoUSIM-SJS1 -a 32627241 -x 901 -y 71 -i 901710000011000 -s  ↩
    8988211000000110000 -o 398153093661279FB1FC74BE07059FEF -k 1 ↩
    D8B2562B992549F20D0F42113EAA6FB
Insert card now (or CTRL-C to cancel)
Generated card parameters :
 > Name    : Magic
 > SMSP    : e1ffffffffffffffffffffffff0581005155f5ffffffffffff000000
 > ICCID   : 8988211000000110000
 > MCC/MNC : 901/71
 > IMSI    : 901710000011000
 > Ki      : 1D8B2562B992549F20D0F42113EAA6FB
 > OPC     : 398153093661279FB1FC74BE07059FEF
 > ACC     : None

Programming ...
Done !
```

7.1.2 Error Codes

The following is a non-comprehensive list of error codes that you might encounter while attempting to programm a SIM card. We only list the ones that are most likely to be encountered.

ValueError: Please provide a PIN-ADM as there is no default one
You didn't provide the ADM key at the command line

RuntimeError: SW match failed ! Expected 9000 and got 63c2.
You provided a wrong ADM key to the card.

RuntimeError: SW match failed ! Expected 9000 and got 6983.
The number of attempts to enter the ADM1 key was exceeded. At this point, there card cannot be recovered and you will never be able to authenticate using ADM1 key again. Still, the card can be used normally, you just cannot make any changes requiring ADM1.

For a more complete list of status and error codes, take a look a the relevant ETSI/3GPP specs or the following source file from the libosmocore project:

```
http://cgit.osmocom.org/libosmocore/tree/src/sim/card_fs_uicc.c
```

7.2 sysmo-usim-tool

In cases where fine-tuning of sysmoUSIM parameters is needed, sysmo-usim-tool adds an extra level of inspection and control. The tool requires the ADM1 key for all operations. The ADM1 key is always specified with the option **-a** or **--adm1** as an 8 digit number.

When supplying the ADM1 key, some extra care has to be taken, since the card will irreversibly lock down when it receives up to three wrong authentication keys.

In order to prevent the user from accidentally damaging the card by messing up the authentication keys, sysmo-usim-tool checks the retry counter before each authentication attempt. If it finds a decreased counter, it will refuse to try another authentication attempt until the option **-f** (**--force**) added to the command line.

7.2.1 OPC value

With sysmo-usim-tool the content of EF.OPC (see Section 8.4). can be inspected and modified if necessary. Option **-o** (**--opc**) displays the OPC value and the OP flag. (0x00 for OP, 0x01 for OPc, see Section 3.3.1)

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -o
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Terminal: OMNIKEY CardMan 4321 00 00
 * Protocol: 1

Authenticating...
 * Remaining attempts: 3
 * Authenticating...
 * Authentication successful
 * Remaining attempts: 3

Reading OP/C value...
 * Initalizing...
 * Reading...
 * Current OPc setting:
   OP: 0x1
   OP/OPc: df3d7f95d27005a5441820a31a020bf6
```

EF.OPC can either hold an OPC or an OP value. The first byte denotes if the following 16 bytes are an OPC or OP value.

An OP value is programmed using option **-O** (**--set-op**)

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -O df3d7f95d27005a5441820a31a020bf6
```

An OPC value is programmed using the option **-C** (**--set-opc**)

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -C df3d7f95d27005a5441820a31a020bf6
```

7.2.2 Ki value

Sysmo-usim-tool also provides access to EF.KI, which holds the precious network authentication key. In order to read out the KI (EF.KI), the option **-k** (**--ki**) can be used.

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -k
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Terminal: OMNIKEY CardMan 4321 00 00
 * Protocol: 1

Authenticating...
 * Remaining attempts: 3
 * Authenticating...
 * Authentication successful
 * Remaining attempts: 3

Reading KI value...
 * Initalizing...
 * Reading...
 * Current KI setting:
   KI: 0123456789abcdef0123456789abcdef
```

Option **-K** (**--set-ki**) allows setting the Ki to a user defined value.

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -K 0123456789ABCDEF0123456789ABCDEF
```

7.2.3 ICCID value

The ICCID can also be changed. Sysmo-usim-tool automatically takes care about the correct swapping of the byte-nibbles. The file has room for 10 bytes or 19-20 decimal digits.

The option **-i** (**--iccid**) extracts the current ICCID of the card. Note that the last nibble is padded with *0xf*, since it is unused

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -i
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Terminal: OMNIKEY CardMan 4321 00 00
 * Protocol: 1

Authenticating...
 * Remaining attempts: 3
 * Authenticating...
 * Authentication successful
 * Remaining attempts: 3

Reading ICCID value...
 * Initalizing...
 * Reading...
 * Current ICCID setting:
   ICCID: 1234567890123456789f
```

A new value can be programmed using the **-I** (**--set-iccid**) command line option:

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -I 8988211000000106578
```

7.2.4 Authentication Algorithms

It is also possible to modify the contents of EF.AUTH, which determines the authentication scheme that is used. Two schemes can be set up, one for 2G and one for 3G.

To inspect which authentication algorithms are currently in configured, the option **-t** (**--auth**) can be used as follows:

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -t
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Terminal: OMNIKEY CardMan 4321 00 00
 * Protocol: 1

Authenticating...
 * Remaining attempts: 3
 * Authenticating...
 * Authentication successful
 * Remaining attempts: 3

Reading Authentication parameters...
 * Initalizing...
 * Reading...
 * Current algorithm setting:
   2G: 0x6
   3G: 0x8
```

The authentication algorithm types are represented as two hex numbers. In the example above. COMP12v1 is configured for 2G and 3G uses Milenage. See also Section 8.5 for a complete list with all authentication algorithms available.

Lets assume that the configuration has to be changed in order to use COMP128v2 (=6) for 2G and XOR 3G (=8) for 3G. To program the authentication parameters option **-T** (**--set-auth**) followed by the colon separated values for 2G and 3G is used. The command line would look like this:

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -T 6:8
```

7.2.5 Milenage parameters (Ci/Ri)

The milenage authentication methos features a set of constants (C1, C2,C3,C4,C5,R1,R2,R3,R4,R5, see Section 8.6). To read the current configuration from the card, command line option **-l** (**--milenage**) can be used:

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -l
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Terminal: OMNIKEY CardMan 4321 00 00
 * Protocol: 1

Authenticating...
 * Remaining attempts: 3
 * Authenticating...
 * Authentication successful
 * Remaining attempts: 3
```

```
Reading Milenage parameters...
 * Initalizing...
 * Reading...
 * Current Milenage Parameters in (EF.MLNGC):
   C1: 00000000000000000000000000000000
   C2: 00000000000000000000000000000001
   C3: 00000000000000000000000000000002
   C4: 00000000000000000000000000000004
   C5: 00000000000000000000000000000008
   R1: 0x40
   R2: 0x0
   R3: 0x20
   R4: 0x40
   R5: 0x60
```

In order to set a new milenage configuration. The option **-L** (**--set-milenage**) is used, followed by the concatenated values of C2, C3, C4, C5, R1, R2, R3, R4 and R5. The parameters may be separated using a colon to increase human readability

The following example resets the milenage parameters to their factory default

```
./sysmo-usim-tool.sjs1.py --adm1 55538407 -L \
00000000000000000000000000000000\
00000000000000000000000000000001\
00000000000000000000000000000002\
00000000000000000000000000000004\
00000000000000000000000000000008\
40:00:20:40:60
```

7.2.6 Enable / Disable USIM Application

In some cases it may become necessary to disable the USIM application. **sysmo-usim-tool** can enable and disable the USIM application as described in Section 8.7 In order to disable USIM, record No.1, which contains the USIM aid, will be overwritten with 0xFF. To enable the USIM application again. **sysmo-usim-tool** restores the factory default to record No.1, which turns the USIM application on again.

The current setting can be inspected using the command line switch **-m** (**--mode**)

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -m
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Terminal: OMNIKEY CardMan 4321 00 00
 * Protocol: 1

Authenticating...
 * Remaining attempts: 3
 * Authenticating...
 * Authentication successful
 * Remaining attempts: 3

Reading SIM-Mode...
 * Initalizing...
 * Reading...
 * Current status of Record No. 1 in EF.DIR:
   61194f10a0000000871002ffffffff890709000050055553696d31ffffffffffffffffffffff
   ==> USIM application enabled
```

In the example above, the USIM application is still enabled. We can disable the USIM application using the command line switch **-c** (**--classic**)

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -c
```

In order to restore the USIM functionality again we can use option **-u** (**--usim**)

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -u
```

7.2.7 APDU trace

In case of problems it may be helpful to trace the exact APDU commands which are exchanged with betweeen card and reader. Use command line switch **-v** or **--verbose** to get a full APDU trace along with the normal output.

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -L \
sysmoUSIM-SJS1 parameterization tool
Copyright (c)2017 Sysmocom s.f.m.c. GmbH

Initializing smartcard terminal...
 * Terminal: OMNIKEY CardMan 4321 00 00
 * Protocol: 1

Authenticating...
   Card transaction: APDU:0020000a00 ==> APDU:(no data) SW:63c3
 * Remaining attempts: 3
 * Authenticating...
   Card transaction: APDU:0020000a083535353338343037 ==> APDU:(no data) SW:9000
 * Authentication successful
   Card transaction: APDU:0020000a00 ==> APDU:(no data) SW:63c3
 * Remaining attempts: 3

Programming Milenage parameters...
 * Initalizing...
   Card transaction: APDU:00a4000c023f00 ==> APDU:(no data) SW:9000
 * New Milenage Parameters for (EF.MLNGC):
   C1: 00000000000000000000000000000000
   C2: 00000000000000000000000000000001
   C3: 00000000000000000000000000000002
   C4: 00000000000000000000000000000004
   C5: 00000000000000000000000000000008
   R1: 0x40
   R2: 0x0
   R3: 0x20
   R4: 0x40
   R5: 0x60
   Card transaction: APDU:00a4000c027fcc ==> APDU:(no data) SW:9000
   Card transaction: APDU:00a4000c026f01 ==> APDU:(no data) SW:9000
 * Programming...
   Card transaction: APDU:00d60000100000000000000000000000000000000000 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d60010100000000000000000000000000000000001 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d60020100000000000000000000000000000000002 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d60030100000000000000000000000000000000004 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d60040100000000000000000000000000000000008 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d600500140 ==> APDU:(no data) SW:9000
   Card transaction: APDU:00d600510100 ==> APDU:(no data) SW:9000
   Card transaction: APDU:00d600520120 ==> APDU:(no data) SW:9000
   Card transaction: APDU:00d600530140 ==> APDU:(no data) SW:9000
   Card transaction: APDU:00d600540160 ==> APDU:(no data) SW:9000
```

8 APDU-Level Card Provisioning

This section describes how to update common parametes stored on the sysmoUSIM-SJS1. This is required if you wish to perform such changes from your own software or scripts, or if you wish to extend existing software.

It is assumed that the reader has some fundamental knowledge about the general smart card command structure (APDUs) as outlined in ISO 7816-4.

In general, the principle to make changes to the card is

- authenticate yourself with the card-specific ADM1 pin value. This unlocks a variety of UPDATE BINARY / UPDATE RECORD commands on files that are not accessible to the regular user
- perform any number of UPDATE BINARY / UPDATE RECORD commands to change the contents of any file, depending on your needs.

8.1 IMSI

In order to change the IMSI, simply perform a standard UPDATE BINARY command on the EF.IMSI (7F20/6F07).

Note
You need to be authenticated using ADM1 PIN.

8.2 ICCID

In order to change the ICCID, simply perform a standard UPDATE BINARY command on the EF.ICCID (2FE2).

Note
You need to be authenticated using ADM1 PIN.

8.3 K / Ki

In order to change the Ki, simply perform a standard UPDATE BINARY command on the EF.KI (7F20/00FF)

Note
You need to be authenticated using ADM1 PIN.

8.4 OPc or OP

In order to change the OPc or OP, simply perform a standard UPDATE BINARY command on the EF.OPC (7F20/00F7)

See Section 3.3.1 on a discussion about whether to store the global OP value or a card-individual pre-computed OPc.

Offset	Size	Description
0	1	0x00 for OP, 0x01 for OPc (see Section 3.3.1)
1	16	OP or OPc value, depending on byte at offset 0

Note
You need to be authenticated using ADM1 PIN.

8.5 Authentication Algorithms

In order to change the Authentication Algorithms used, simply perform a standard UPDATE BINARY command on the EF.AUTH (7FCC/6F00)

This file contains two bytes:

- the first byte specifies the 2G algorithm to use
- the second byte specifies the 3G algorithm to use

Value	Algorithm	Supported Mode
01	Milenage	2G + 3G
03	COMP12v1	2G
04	XOR 2G	2G
06	COMP128v2	2G
07	COMP128v3	2G
08	XOR 3G	3G

Note
You need to be authenticated using ADM1 PIN.

8.6 Milenage Configuration (Ci/Ri)

The Milenage configuration constants Ci (i=1..5) and Ri (i=1.5) can be configured in EF.MLNGC (7FCC/6F01)

The structure of the file is as follows:

Offset	Size	Description
0	16	C1
16	16	C2
32	16	C3
48	16	C4
64	16	C5
80	1	R1
81	1	R2
82	1	R3
83	1	R4
84	1	R5

Note
You need to be authenticated using ADM1 PIN.

If this file doesn't exist, then the below default values are provisioned, in accordance with [3gpp-ts-35-206] Section 4.1.

Parameter	Value
C1	00000000000000000000000000000000
C2	00000000000000000000000000000001

Parameter	Value
C3	00000000000000000000000000000002
C4	00000000000000000000000000000004
C5	00000000000000000000000000000008
R1	0x40
R2	0x00
R3	0x20
R4	0x40
R5	0x60

8.7 Disabling the USIM Application

The sysmoUSIM offers both a modern USIM application, as well as a classic SIM application. In some situations you may want to restrict/reduce the feature set to that of a classic GSM SIM card.

In order to make sure that mobile phones will no longer find the USIM application on the card, we recommend you modify the EF.DIR and remove the record pointingto ADF.USIM. This way, USIM detection will fail, and phones will fall back to the classic GSM SIM protocol.

According to Annex E of [etsi-ts101220], the AID of the USIM applications starts with the prefix of A0000000871002. Remove the matching record from EF.DIR.

For more information on EF.DIR, see Section 13.1 of [etsi-ts102221].

Note
You need to be authenticated using ADM1 PIN.

9 Java Card Features

The sysmoUSIM-SJS1 is a Java Card (and Java SIM/USIM card) compliant to the specifications listed in Section 4.2.1

Note
In order to install and/or manage Java Card applets on your card, you need to purchase a variant of the card that provides the ADM keys to you (see Section 4.3.2 and Section 4.3.3, respectively).

9.1 Application List

Table 1: List of Java applications oa card

Application	PID	AID
SIM Application	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF8900000000
SIM RFM Application	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF89B00010
RAM Application	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF89B00010
USIM RFM App (opt)	FF434E52581040040203	FF434E52581040040203000000000000
USIM Application	A0000000871002FF49FFFF8900	A0000000871002FF49FFFF89040B0000
CAT_TP App (opt)	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF89B00010

The detailed coding/suffix of the PID / AID may change from card batch to card batch. In case of any questions, please refer to Annex E of [etsi-ts101220] for the AID prefixes applications have to start with or the PID values.

For more information on EF.DIR, see Section 13.1 of [etsi-ts102221].

9.2 Example Applet

There is an example "Hello World" applet provided in source code, you can find it at git://git.osmocom.org/sim/hello-stk / https://git.osmocom.org/sim/hello-stk/

Please follow the instructions at https://osmocom.org/projects/cellular-infrastructure/wiki/Shadysimpy to install the hello world STK applet.

9.3 Tools

The tools provided at git://git.osmocom.org/sim/sim-tools / http://git.osmocom.org/sim/sim-tools/ can be used to download the example applet (or other applets) onto the card.

Installation can happen either locally by inserting the SIM into a smart card reader attached to your PC, or remotely over the air of your Osmocom based GSM network by sending OTA SMS via the SMPP interface of OsmoNITB.

10 Glossary

2FF
2nd Generation Form Factor; the so-called plug-in SIM form factor

3FF
3rd Generation Form Factor; the so-called microSIM form factor

3GPP
3rd Generation Partnership Project

4FF
4th Generation Form Factor; the so-called nanoSIM form factor

A Interface
Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.008* [3gpp-ts-48-008])

A3/A8
Algorithm 3 and 8; Authentication and key generation algorithm in GSM and GPRS, typically COMP128v1/v2/v3 or MILENAGE are typically used

A5
Algorithm 5; Air-interface encryption of GSM; currently only A5/0 (no encryption), A5/1 and A5/3 are in use

Abis Interface
Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.058* [3gpp-ts-48-058] and *3GPP TS 52.021* [3gpp-ts-52-021])

ACC
Access Control Class; every BTS broadcasts a bit-mask of permitted ACC, and only subscribers with a SIM of matching ACC are permitted to use that BTS

AGCH
Access Grant Channel on Um interface; used to assign a dedicated channel in response to RACH request

AGPL
GNU Affero General Public License, a copyleft-style Free Software License

ARFCN
Absolute Radio Frequency Channel Number; specifies a tuple of uplink and downlink frequencies

AUC
: Authentication Center; central database of authentication key material for each subscriber

BCCH
: Broadcast Control Channel on Um interface; used to broadcast information about Cell and its neighbors

BCC
: Base Station Color Code; short identifier of BTS, lower part of BSIC

BTS
: Base Transceiver Station

BSC
: Base Station Controller

BSIC
: Base Station Identity Code; 16bit identifier of BTS within location area

BSSGP
: Base Station Subsystem Gateway Protocol (*3GPP TS 48.018* [3gpp-ts-48-018])

BVCI
: BSSGP Virtual Circuit Identifier

CBCH
: Cell Broadcast Channel; used to transmit Cell Broadcast SMS (SMS-CB)

CC
: Call Control; Part of the GSM Layer 3 Protocol

CCCH
: Common Control Channel on Um interface; consists of RACH (uplink), BCCH, PCH, AGCH (all downlink)

Cell
: A cell in a cellular network, served by a BTS

CEPT
: Conférence européenne des administrations des postes et des télécommunications; European Conference of Postal and Telecommunications Administrations.

CGI
: Cell Global Identifier comprised of MCC, MNC, LAC and BSIC

dB
: deci-Bel; relative logarithmic unit

dBm
: deci-Bel (milliwatt); unit of measurement for signal strength of radio signals

DHCP
: Dynamic Host Configuration Protocol (*IETF RFC 2131* [ietf-rfc2131]

downlink
: Direction of messages / signals from the network core towards the mobile phone

DSP
: Digital Signal Processor

dvnixload
: Tool to program UBL and the Bootloader on a sysmoBTS

EDGE
: Enhanced Data rates for GPRS Evolution; Higher-speed improvement of GPRS; introduces 8PSK

EGPRS
Enhanced GPRS; the part of EDGE relating to GPRS services

ESME
External SMS Entity; an external application interfacing with a SMSC over SMPP

ETSI
European Telecommunications Standardization Institute

FPGA
Field Programmable Gate Array; programmable digital logic hardware

Gb
Interface between PCU and SGSN in GPRS/EDGE network; uses NS, BSSGP, LLC

GERAN
GPRS/EDGE Radio Access Network

GGSN
GPRS Gateway Support Node; gateway between GPRS and external (IP) network

GMSK
Gaussian Minimum Shift Keying; modulation used for GSM and GPRS

GPL
GNU General Public License, a copyleft-style Free Software License

Gp
Gp interface between SGSN and GGSN; uses GTP protocol

GPS
Global Positioning System; provides a highly accurate clock reference besides the global position

GSM
Global System for Mobile Communications. ETSI/3GPP Standard of a 2G digital cellular network

GSMTAP
GSM tap; pseudo standard for encapsulating GSM protocol layers over UDP/IP for analysis

GTP
GPRS Tunnel Protocol; used between SGSN and GGSN

HLR
Home Location Register; central subscriber database of a GSM network

HPLMN
Home PLMN; the network that has issued the subscriber SIM and has his record in HLR

IE
Information Element

IMEI
International Mobile Equipment Identity; unique identifier for the mobile phone

IMSI
International Mobile Subscriber Identity; 15-digit unique identifier for the subscriber/SIM; starts with MCC/MNC of issuing operator

IP
Internet Protocol (*IETF RFC 791* [?])

IPA
ip.access GSM over IP protocol; used to multiplex a single TCP connection

LAC
Location Area Code; 16bit identifier of Location Area within network

LAPD
Link Access Protocol, D-Channel (*ITU-T Q.921* [itu-t-q921])

LAPDm
Link Access Protocol Mobile (*3GPP TS 44.006* [3gpp-ts-44-006])

LLC
Logical Link Control; GPRS protocol between MS and SGSN (*3GPP TS 44.064* [3gpp-ts-44-064])

Location Area
Location Area; a geographic area containing multiple BTS

MCC
Mobile Country Code; unique identifier of a country, e.g. 262 for Germany

MFF
Machine-to-Machine Form Factor; a SIM chip package that is soldered permanently onto M2M device circuit boards.

MGW
Media Gateway

MM
Mobility Management; part of the GSM Layer 3 Protocol

MNC
Mobile Network Code; identifies network within a country; assigned by national regulator

MNO
Mobile Network Operator; operator with physical radio network under his MCC/MNC

MS
Mobile Station; a mobile phone / GSM Modem

MSC
Mobile Switching Center; network element in the circuit-switched core network

MSISDN
Mobile Subscriber ISDN Number; telephone number of the subscriber

MVNO
Mobile Virtual Network Operator; Operator without physical radio network

NCC
Network Color Code; assigned by national regulator

NITB
Network In The Box; combines functionality traditionally provided by BSC, MSC, VLR, HLR, SMSC functions; see OsmoNITB

NSEI
NS Entity Identifier

NVCI
NS Virtual Circuit Identifier

NWL
Network Listen; ability of some BTS to receive downlink from other BTSs

NS
Network Service; protocol on Gb interface (*3GPP TS 48.016* [3gpp-ts-48-016])

OCXO
Oven Controlled Crystal Oscillator; very high precision oscillator, superior to a VCTCXO

OML
Operation & Maintenance Link (ETSI/*3GPP TS 52.021* [3gpp-ts-52-021])

OpenBSC
Open Source implementation of GSM network elements, specifically OsmoBSC, OsmoNITB, OsmoSGSN

OpenGGSN
Open Source implementation of a GPRS Packet Control Unit

OpenVPN
Open-Source Virtual Private Network; software employed to establish encrypted private networks over untrusted public networks

Osmocom
Open Source MObile COMmunications; collaborative community for implementing communications protocols and systems, including GSM, GPRS, TETRA, DECT, GMR and others

OsmoBSC
Open Source implementation of a GSM Base Station Controller

OsmoNITB
Open Source implementation of a GSM Network In The Box, combines functionality traditionally provided by BSC, MSC, VLR, HLR, AUC, SMSC

OsmoSGSN
Open Source implementation of a Serving GPRS Support Node

OsmoPCU
Open Source implementation of a GPRS Packet Control Unit

OTA
Over-The-Air; Capability of operators to remotely reconfigure/reprogram ISM/USIM cards

PCH
Paging Channel on downlink Um interface; used by network to page an MS

PCU
Packet Control Unit; used to manage Layer 2 of the GPRS radio interface

PDCH
Packet Data Channel on Um interface; used for GPRS/EDGE signalling + user data

PIN
Personal Identification Number; a number by which the user authenticates to a SIM/USIM or other smart card

PLMN
Public Land Mobile Network; specification language for a single GSM network

PUK
PIN Unblocking Code; used to unblock a blocked PIN (after too many wrong PIN attempts)

RAC
Routing Area Code; 16bit identifier for a Routing Area within a Location Area

RACH
Random Access Channel on uplink Um interface; used by MS to request establishment of a dedicated channel

RAM
Remote Application Management; Ability to remotely manage (install, remove) Java Applications on SIM/USIM Card

RF
: Radio Frequency

RFM
: Remote File Management; Ability to remotely manage (write, read) files on a SIM/USIM card

Roaming
: Procedure in which a subscriber of one network is using the radio network of another network, often in different countries; in some countries national roaming exists

Routing Area
: Routing Area; GPRS specific sub-division of Location Area

RR
: Radio Resources; Part of the GSM Layer 3 Protocol

RSL
: Radio Signalling Link (*3GPP TS 48.058* [3gpp-ts-48-058])

RTP
: Real-Time Transport Protocol (*IETF RFC 3550* [ietf-rfc3550]); Used to transport audio/video streams over UDP/IP

SACCH
: Slow Associate Control Channel on Um interface; bundled to a TCH or SDCCH, used for signalling in parallel to active dedicated channel

SDCCH
: Slow Dedicated Control Channel on Um interface; used for signalling and SMS transport in GSM

SDK
: Software Development Kit

SIM
: Subscriber Identity Module; small chip card storing subscriber identity

Site
: A site is a location where one or more BTSs are installed, typically three BTSs for three sectors

SMPP
: Short Message Peer-to-Peer; TCP based protocol to interface external entities with an SMSC

SMSC
: Short Message Service Center; store-and-forward relay for short messages

SSH
: Secure Shell; *IETF RFC 4250* [ietf-rfc4251] to 4254

syslog
: System logging service of UNIX-like operating systems

System Information
: A set of downlink messages on the BCCH and SACCH of the Um interface describing properties of the cell and network

TCH
: Traffic Channel; used for circuit-switched user traffic (mostly voice) in GSM

TCP
: Transmission Control Protocol; (*IETF RFC 793* [ietf-rfc793])

TFTP
: Trivial File Transfer Protocol; (*IETF RFC 1350* [ietf-rfc1350])

TRX
: Transceiver; element of a BTS serving a single carrier

u-Boot
Boot loader used in various embedded systems

UBI
An MTD wear leveling system to deal with NAND flash in Linux

UBL
Initial bootloader loaded by the TI Davinci SoC

UDP
User Datagram Protocol (*IETF RFC 768* [ietf-rfc768])

UICC
Universal Integrated Chip Card; A smart card according to *ETSI TR 102 216* [etsi-tr102216]

Um interface
U mobile; Radio interface between MS and BTS

uplink
Direction of messages: Signals from the mobile phone towards the network

USIM
Universal Subscriber Identity Module; application running on a UICC to provide subscriber identity for UMTS and GSM networks

VCTCXO
Voltage Controlled, Temperature Compensated Crystal Oscillator; a precision oscillator, superior to a classic crystal oscillator, but inferior to an OCXO

VPLMN
Visited PLMN; the network in which the subscriber is currently registered; may differ from HPLMN when on roaming

VTY
Virtual TeletYpe; a textual command-line interface for configuration and introspection, e.g. the OsmoBSC configuration file as well as its telnet link on port 4242

A Osmocom TCP/UDP Port Numbers

The Osmocom GSM system utilizes a variety of TCP/IP based protocols. The table below provides a reference as to which port numbers are used by which protocol / interface.

Table 2: TCP/UDP port numbers

L4 Protocol	Port Number	Purpose	Software
UDP	2427	MGCP GW	osmo-bsc_mgcp
TCP	2775	SMPP (SMS interface for external programs)	osmo-nitb
TCP	3002	A-bis/IP OML	osmo-bts, osmo-bsc, osmo-nitb
TCP	3003	A-bis/IP RSL	osmo-bts, osmo-bsc, osmo-nitb
TCP	4239	telnet (VTY)	osmo-stp
TCP	4240	telnet (VTY)	osmo-pcu
TCP	4241	telnet (VTY)	osmo-bts
TCP	4242	telnet (VTY)	osmo-nitb, osmo-bsc, cellmgr-ng
TCP	4243	telnet (VTY)	osmo-bsc_mgcp
TCP	4244	telnet (VTY)	osmo-bsc_nat
TCP	4245	telnet (VTY)	osmo-sgsn
TCP	4246	telnet (VTY)	osmo-gbproxy
TCP	4247	telnet (VTY)	OsmocomBB

Table 2: (continued)

L4 Protocol	Port Number	Purpose	Software
TCP	4249	Control Interface	osmo-nitb, osmo-bsc
TCP	4250	Control Interface	osmo-bsc_nat
TCP	4251	Control Interface	osmo-sgsn
TCP	4252	telnet (VTY)	sysmobts-mgr
TCP	4253	telnet (VTY)	osmo-gtphub
TCP	4254	telnet (VTY)	osmo-msc
TCP	4255	Control Interface	osmo-msc
TCP	4256	telnet (VTY)	osmo-sip-connector
TCP	4257	Control Interface	ggsn (OpenGGSN)
TCP	4258	telnet (VTY)	osmo-hlr
TCP	4259	Control Interface	osmo-hlr
TCP	4260	telnet (VTY)	ggsn (OpenGGSN)
UDP	4729	GSMTAP	Almost every osmocom project
TCP	5000	A/IP	osmo-bsc, osmo-bsc_nat
UDP	2427	GSMTAP	osmo-pcu, osmo-bts
UDP	23000	GPRS-NS over IP default port	osmo-pcu, osmo-sgsn, osmo-gbproxy

B Bibliography / References

References

[1] [osmobts-abis-spec] Neels Hofmeyr & Harald Welte. OsmoBTS Abis Protocol Specification. http://ftp.osmocom.org/docs/latest/osmobts-abis.pdf

[2] [userman-osmobts] Osmocom Project: OsmoBTS User Manual. http://ftp.osmocom.org/docs/latest/osmobts-usermanual.pdf

[3] [vty-ref-osmobts] Osmocom Project: OsmoBTS VTY Reference Manual. http://ftp.osmocom.org/docs/latest/-osmobts-vty-reference.pdf

[4] [userman-osmobsc] Osmocom Project: OsmoBSC User Manual. http://ftp.osmocom.org/docs/latest/osmobsc-usermanual.pdf

[5] [vty-ref-osmobsc] Osmocom Project: OsmoBSC VTY Reference Manual. http://ftp.osmocom.org/docs/latest/-osmobsc-vty-reference.pdf

[6] [userman-osmopcu] Osmocom Project: OsmoPCU User Manual. http://ftp.osmocom.org/docs/latest/osmopcu-usermanual.pdf

[7] [vty-ref-osmopcu] Osmocom Project: OsmoPCU VTY Reference Manual. http://ftp.osmocom.org/docs/latest/-osmopcu-vty-reference.pdf

[8] [userman-osmonitb] Osmocom Project: OsmoNITB User Manual. http://ftp.osmocom.org/docs/latest/-osmonitb-usermanual.pdf

[9] [vty-ref-osmonitb] Osmocom Project: OsmoNITB VTY Reference Manual. http://ftp.osmocom.org/docs/-latest/osmonitb-vty-reference.pdf

[10] [userman-osmosgsn] Osmocom Project: OsmoSGSN User Manual. http://ftp.osmocom.org/docs/latest/-osmosgsn-usermanual.pdf

[11] [vty-ref-osmosgsn] Osmocom Project: OsmoSGSN VTY Reference Manual. http://ftp.osmocom.org/docs/-latest/osmonitb-vty-reference.pdf

[12] [userman-osmoggsn] Osmocom Project: OpenGGSN User Manual. http://ftp.osmocom.org/docs/latest/-osmoggsn-usermanual.pdf

[13] [vty-ref-osmoggsn] Osmocom Project: OsmoGGSN VTY Reference Manual. http://ftp.osmocom.org/docs/-latest/osmoggsn-vty-reference.pdf

[14] [3gpp-ts-23-048] 3GPP TS 23.048: Security mechanisms for the (U)SIM application toolkit; Stage 2 http://www.3gpp.org/DynaReport/23048.htm

[15] [3gpp-ts-24-007] 3GPP TS 24.007: Mobile radio interface signalling layer 3; General Aspects http://www.3gpp.org/DynaReport/24007.htm

[16] [3gpp-ts-24-008] 3GPP TS 24.008: Mobile radio interface Layer 3 specification; Core network protocols; Stage 3. http://www.3gpp.org/dynareport/24008.htm

[17] [3gpp-ts-31-101] 3GPP TS 31.101: UICC-terminal interface; Physical and logical characteristics http://www.3gpp.org/DynaReport/31101.htm

[18] [3gpp-ts-31-102] 3GPP TS 31.102: Characteristics of the Universal Subscriber Identity Module (USIM) application http://www.3gpp.org/DynaReport/31102.htm

[19] [3gpp-ts-31-111] 3GPP TS 31.111: Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) http://www.3gpp.org/DynaReport/31111.htm

[20] [3gpp-ts-31-115] 3GPP TS 31.115: Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications http://www.3gpp.org/DynaReport/31115.htm

[21] [3gpp-ts-31-116] 3GPP TS 31.116: Remote APDU Structure for (U)SIM Toolkit applications http://www.3gpp.org/DynaReport/31116.htm

[22] [3gpp-ts-35-205] 3GPP TS 35.205: 3G Security; Specification of the MILENAGE algorithm set: General

[23] [3gpp-ts-35-206] 3GPP TS 35.206: 3G Security; Specification of the MILENAGE algorithm set: Algorithm specification http://www.3gpp.org/DynaReport/35206.htm

[24] [3gpp-ts-44-006] 3GPP TS 44.006: Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification http://www.3gpp.org/DynaReport/44006.htm

[25] [3gpp-ts-44-064] 3GPP TS 44.064: Mobile Station - Serving GPRS Support Node (MS-SGSN); Logical Link Control (LLC) Layer Specification http://www.3gpp.org/DynaReport/44064.htm

[26] [3gpp-ts-48-008] 3GPP TS 48.008: Mobile Switching Centre - Base Station system (MSC-BSS) interface; Layer 3 specification http://www.3gpp.org/DynaReport/48008.htm

[27] [3gpp-ts-48-016] 3GPP TS 48.016: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN) interface; Network service http://www.3gpp.org/DynaReport/48016.htm

[28] [3gpp-ts-48-018] 3GPP TS 48.018: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN); BSS GPRS protocol (BSSGP) http://www.3gpp.org/DynaReport/48018.htm

[29] [3gpp-ts-48-056] 3GPP TS 48.056: Base Station Controller - Base Transceiver Station (BSC - BTS) interface; Layer 2 specification http://www.3gpp.org/DynaReport/48056.htm

[30] [3gpp-ts-48-058] 3GPP TS 48.058: Base Station Controller - Base Transceiver Station (BSC - BTS) Interface; Layer 3 specification http://www.3gpp.org/DynaReport/48058.htm

[31] [3gpp-ts-51-011] 3GPP TS 51.011: Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface

[32] [3gpp-ts-51-014] 3GPP TS 51.014: Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface http://www.3gpp.org/DynaReport/51014.htm

[33] [3gpp-ts-52-021] 3GPP TS 52.021: Network Management (NM) procedures and messages on the A-bis interface http://www.3gpp.org/DynaReport/52021.htm

[34] [etsi-tr102216] ETSI TR 102 216: Smart cards http://www.etsi.org/deliver/etsi_tr/102200_102299/102216/-03.00.00_60/tr_102216v030000p.pdf

[35] [etsi-ts102221] ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics http://www.etsi.org/deliver/etsi_ts/102200_102299/102221/13.01.00_60/ts_102221v130100p.pdf

[36] [etsi-ts101220] ETSI TS 101 220: Smart Cards; ETSI numbering system for telecommunication application providers http://www.etsi.org/deliver/etsi_ts/101200_101299/101220/12.00.00_60/ts_101220v120000p.pdf

[37] [ietf-rfc768] IETF RFC 768: Internet Protocol https://tools.ietf.org/html/rfc791

[38] [ietf-rfc793] IETF RFC 793: Transmission Control Protocol https://tools.ietf.org/html/rfc793

[39] [ietf-rfc1350] IETF RFC 1350: Trivial File Transfer Protool https://tools.ietf.org/html/rfc1350

[40] [ietf-rfc2131] IETF RFC 2131: Dynamic Host Configuration Protocol https://tools.ietf.org/html/rfc2131

[41] [ietf-rfc3550] IETF RFC 3550: RTP: A Transport protocol for Real-Time Applications https://tools.ietf.org/-html/rfc3550

[42] [ietf-rfc4251] IETF RFC 4251: The Secure Shell (SSH) Protocol Architecture https://tools.ietf.org/html/-rfc4251

[43] [itu-t-q921] ITU-T Q.921: ISDN user-network interface - Data link layer specification https://www.itu.int/rec/-T-REC-Q.921/en

[44] [smpp-34] SMPP Develoepers Forum. Short Message Peer-to-Peer Protocol Specification v3.4 http://docs.nimta.com/SMPP_v3_4_Issue1_2.pdf

[45] [gnu-agplv3] Free Software Foundation. GNU Affero General Public License. http://www.gnu.org/licenses/-agpl-3.0.en.html