sysmocom

sysmocom - s.f.m.c. GmbH

sysmoUSIM-SJS1 / sysmoISIM-SJA2 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
v4	April 2018	Removed custom IMSI/Ki programming variant	sk
v4	September 2020	Added sysmoISIM-SJA2 related information	hw
v5	November 2020	Added form-factor related text and introduction to SIM related identities	hw
v6	January 2021	Added NOTEs about erroneous SCP02/OTA key material; Acknowledgements	hw
v7	February 2021	Various typo fixes; add chapter on shared vs. per-application separate keys	hw
v8	May 2021	Add chapter about 5G services + files	hw
v9	September 2021	Document change of default MCC-MNC / IMSI from 901-70 to 999-70 to avoid further overlap with allocation of that MNC to Clementvale Baltic OY.	hw
v10	February 2022	Add OTA chapter; document ADF.USIM locking; document ADF.ISIM locking; refer to 5G disabling script in pySim rather than listing lengthy command sequences here.	hw
v11	October 2023	Remove git:// URLs for git repositories; always link to new gitea URLs directly rather than using legacy HTTP redirects.	hw

Contents

1   Introduction

This manual describes the sysmoUSIM-SJS1 and sysmoISIM-SJA2, state-of-the-art Java SIM/USIM cards for authentication in cellular networks.

The target audience are operators of cellular networks (large and small) who use the sysmocom SIM 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 protocol stack, as this is what the sysmoUSIM-SJS1 / sysmoISIM-SJA2 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 and sysmoISIM-SJA2, 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.

In 2020, sysmocom introduced the sysmoISIM-SJA2. The migration to this new card model was mandated by end-of-life of both the chip and the operating system used in the sysmoUSIM-SJS1. The new sysmoISIM-SJA2 has is an almost feature-complete successor. In addition, it adds an ISIM (IMS/VoLTE) and HPSIM application.

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, LTE or 5G. This includes OpenBTS, OsmoBTS, OpenBSC, OsmoSGSN, YateBTS, srsLTE, nextepc, open5gs 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   ISIM Cards

The IMS (IP Multimedia System, required for VoLTE) introduced the *ISIM Application* which runs on top of an ETSI UICC, typically in parallel to the USIM Application.

While IMS/VoLTE can be operated on a legacy USIM without an ISIM application, many related parameters can only be configured with an on-card ISIM application.

NOTE

Many UE/phone manufacturers implement additional constraints/restrictions on when to enable IMS/VoLTE functionality. The presence of an ISIM application on the card may not be sufficient to unlocking VoLTE capabilities.

3.4 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, sysmoUSIM-SJS1 and sysmoISIM-SJA2 all support the full set of COMP128v1, COMP128v2 and COMP128v3 algorithms. For the USIM/ISIM products, this only applies when used in classic SIM protocol, and not via USIM/ISIM 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 practice most networks seem to utilize the MILENAGE algorithm.

The sysmoUSIM-SJS1 and sysmoISIM-SJA2 implement the MILENAGE algorithm.

See also Section 11.5

3.4.1 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 and sysmoISIM-SJA2 support 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].

3.4.2 Shared or application-specific keys

Card with multiple applications, such as the sysmoISIM-SJA2 containing SIM, USIM and ISIM applications can either

- use *shared keys* for all of those applications, meaning that one set of K and OP/OPc is used for authentication to any of those applications, or

- use *separate keys* for each of those applications. This would permit separate K and OP/OPc values for each application, e.g. different keys for authentication of USIM against the radio network and ISIM against the IMS network

---

sysmoUSIM-SJS1 uses shared keys.

In order to ensure more flexibility, sysmoISIM-SJA2 uses *separate keys*.

In order to avoid frequent mis-configuration, `sysmo-isim-tool.sja2.py` as described in Section 7.3 will write the identical key to all of the applications. If you actually need separate keys, you would have to write the keys using custom software or modify the existing open source tools.

3.5 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.

3.5.1 SIM Card vs. Network Technology 2G/3G/4G/5G

Depending on operator configuration, a classic 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). If the phone only supports the classic SIM protocol, the SIM application on the card will be used.

LTE / EUTRAN always requires at least an USIM. Classic SIM are not sufficient for authentication on LTE/EUTRAN.

5G / NR also requires at least an USIM. However, certain additional features like SUCI require SIM card support. The sysmoISIM-SJA2 supports "SUCI calculation by ME" using EF_SUCI_Calc_Info as per 3GPP TS 31.102 Section 5.3.47. It does not support "SUCI Calculation by USIM" as per section 5.3.48.

3.5.2 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 and sysmoISIM-SJA2 support both GSM as well as USIM mode, and are pre-provisioned in a way that both modes are available.

3.6 Card Form Factors

The original SIM cards for (portable, not hand-held!) GSM phones was credit-card sized.

3.6.1 2FF, 3FF, 4FF

Later on, the plug-in (2FF), micro (3FF) and nano (4FF) form-factors have been specified as the microelectronic integration progressed and phones became smaller and smaller.

All the above-mentioned form-factors are removable plastic cards with contacts and embedded SIM card chip.
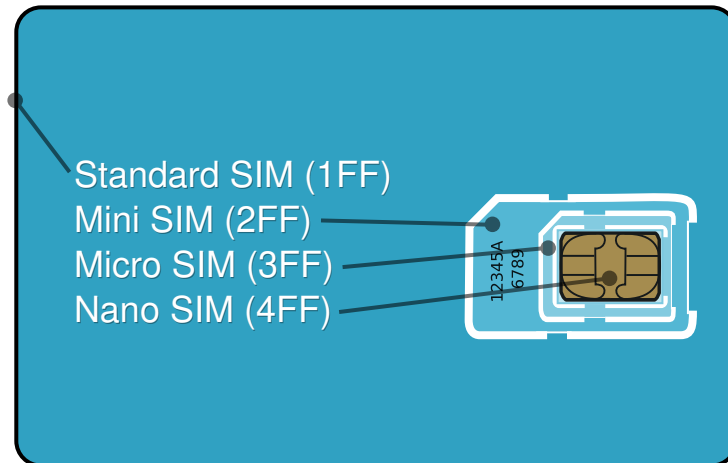
Figure 1: Card Form Factors by Cvdr based on Justin Ormont's work. CC BY-SA 3.0, via Wikimedia Commons

The sysmoUSIM-SJS1 card were available either as combined 2FF+3FF, or as combined 2FF+4FF cards.

The sysmoISIM-SJA2 cards are made available in an "all-in-one" form-factor that covers the full credit-card size, 2FF, 3FF and 4FF form-factors in a single product.

3.6.2 MFF2

Furthermore, ETSI also specified the so called MFF2 form factor, which is a solder-type form-factor, meaning that the SIM is soldered to the circuit board like any other surface-mounted integrated circuit and hence no longer user swappable/removable.

Upon request, sysmocom can also provide the sysmoISIM-SJA2 in MFF2 package.

3.7 Identities

There are several identities associated with the use of SIM cards.

3.7.1 ICCID

The ICCID (Integrated Circuit Card IDentifier) is a [supposedly] globally unique serial number of chip cards. It is specified by ITU-T in ITU-T recommendation E.118. Its layout is based on ISO/IEC 7812.

It can be up to 22 digits long, including the Luhn check digit.

The ICCID is never transmitted over the radio interface and hence doesn't really play a relevant role in terms of operating a private cellular network.

3.7.2 IMSI

The IMSI (International Mobile Subscriber Identifier) is a [supposedly] globally unique number of the subscriber of 3GPP network technology. The number must be unique in public networks, but not necessarily so in private networks.

The first 5-6 digits of the IMSI are typically comprised by the MCC (Mobile Country Code) and MNC (Mobile Network Code).

The MCC specifies the country, and the MNC the card-issuing network within the country.

MNCs are assigned by the respective national telecommunications regulatory authority. The policies differ from country to country, but typically you have to be a licensed mobile network operator (with your own spectrum allocation) in order to receive a MCC allocation and hence be able to issue your own IMSIs within your MCC-MNC.

As sysmocom is not a telecom operator but just a R&D and equipment manufacturing company, we do not have our own MCC/IMSI allocations - neither can we get one.

The default IMSIs provisioned on the sysmoUSIM / sysmoISIM cards are within the MCC-MNC of

- **901-70** (until August 2021), where 901 is a trans-national country code, used for applications that span beyond one country but which are not of global scope. 901-70 used to be allocated but was abandoned/unused for a long time, which is why sysmocom started to use it as the default code on the sysmoUSIM / sysmoISIM products. Unfortunately, meanwhile ITU re-assigned 901-70 to an operator (Clementvale Baltic OY). This means there may be overlaps, and we strongly recommend to change the MCC-MNC and hence IMSI of the cards you use in your networks to a different one. It's best to follow the recommendations of the respective regulatory authority in your jurisdiction.

- **999-70** (from September 2021), where 999 is a newly-allocated trans-national MCC allocated by ITU for private cellular networks. The IMSI allocations within MNCs under MCC 999 are not coordinated or registered, so there is no guarantee that they are globally unique. However, sysmocom will not issue IMSIs multiple times, so if you use only sysmocom cards, you are guaranteed no duplicated within those cards.

3.7.3 MSISDN

The MSISDN (Mobile Subscriber ISDN Number) is the phone number allocated to a subscriber within the global numbering plan for telephony as per ITU-T recommendation E.164.

The MSISDN can optionally be stored on the SIM card, but in reality this is not required for operation of a cellular network: Only the core network elements (particularly the MSC or the IMS core) need to know which MSISDN can be reached behind which IMSI. The phone itself doesn't need to know its own MSISDN to receive or originate calls.

4 sysmoUSIM-SJS1 and sysmoISIM-SJA2 specifications
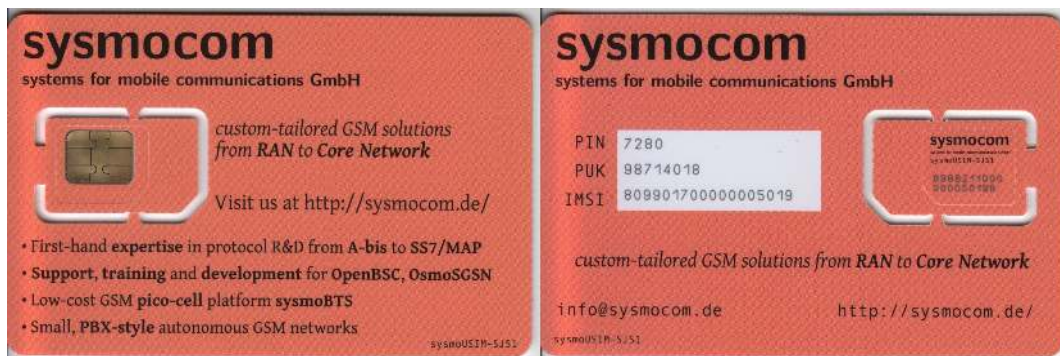
Figure 2: sysmoUSIM-SJS1

Figure 3: sysmoISIM-SJA2

The sysmoUSIM-SJS1 and sysmoISIM-SJA2 are Java SIM card with the following specifications and features:

4.1 Physical Specification

- Available mechanical form factor

  - 2FF + 3FF (micro) SIM card (sysmoUSIM-SJS1)
  - 2FF + 4FF (nano) SIM card (sysmoUSIM-SJS1)
  - 2FF + 3FF + 4FF (sysmoISIM-SJA2)
  - 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]
  - 3GPP TS 31.103; [3gpp-ts-31-103] (sysmoISIM-SJA2 only)

- 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 (sysmoUSIM-SJS1), MILENAGE (sysmoISIM-SJA2)

- 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
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 the cards by default. See Section 3.4.1 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 or sysmoISIM-SJA2, unless the customer has specified different provisioning data at time of purchase (or changed the values after purchase):

Identity	Value
IMSI	99970xxxxxxxxxx (from September 2021 onwards); 90170xxxxxxxxx (until August 2021)
ACC	equal distribution

Identity	Value
ICCID	8988211xxxxxxxxxxx
MSISDN	88211xxxxxx

4.3 Product Variants

The cards are sold by sysmocom in the following different product variants, depending on your needs.

Variant	IMSI	K, OPc	ADM key	OTA key
default 999-70 MCC-MNC (no ADM keys)	99970…	random, provided	not provided	not provided
default 999-70 MCC-MNC (with ADM keys)	99970…	random, provided	provided	provided

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

Other variants could be offered on an individual base if you are interested in bigger volumes of cards (starting around 5000).

SKU	Form-Factor	Link to sysmocom webshop
sysmoISIM-SJA2	1FF + 2FF + 3FF + 4FF	https://shop.sysmocom.de/sysmoISIM-SJA2-SIM-USIM-ISIM-Card-10-pack-with-ADM-keys/sysmoISIM-SJA2-10p-adm
sysmoISIM-SJA2-MFF2	MFF2	https://shop.sysmocom.de/sysmoISIM-SJA2-SIM-USIM-ISIM-MFF2-solder-type-10-pack-with-ADM-keys/-sysmoISIM-SJA2-MFF2-10p-adm

MFF form-factor cards are available upon request on a per-project basis.

4.3.1 default 999-70 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 999-70 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 receive 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 receive 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/ISIM 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/UICC/USIM/ISIM 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 suitable USB CCID compliant card readers at https://shop.sysmocom.de/SIM/Card-Readers-Writers/

Figure 4: Omnikey CardMan 3121 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

Smart 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, reliable adapter at https://shop.sysmocom.de/Professional-SIM-card-adapter-plug-in-micro-nano-SIM-to-full-size/sim-adapter-pcb

We also sell a number of other adapters suitable for different use cases, for example for interfacing

- MFF2-packaged UICC / eUICC with a card reader (solder type)

- MFF2-packaged UICC / eUICC with a card reader (ZIF socket type)

- half-sized cards with card readers whose slot is deeper than the card

- Flex-PCB (FPC) adapters to use MFF2 or full-sized cards in 2FF/3FF/4FF slots

The full range of adapter products is available from https://shop.sysmocom.de/SIM/Adapters/

6 Testing sysmocom SIM cards with utilities

There are some utilities that can be used to test the sysmocom SIM cards 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 https://gitea.osmocom.org/osmocom/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 sysmocom SIM 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 / https://gitea.osmocom.org/sim-card/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 or sysmoISIM-SJA2 card, you can change any identity (IMSI, 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 https://gitea.osmocom.org/sim-card/pysim

- the **sysmo-usim-tool** program from https://gitea.sysmocom.de/sysmocom/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    : e1ffffffffffffffffffffffff0581005155f5ffffffffffffff000000
 > 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 (sysmoUSIM-SJS1)

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 11.4). 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 3.4.1)

```
$ ./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...
 * Initializing...
 * Reading...
 * Current algorithm setting:
```

```
   2G: 0x3
   3G: 0x1
```

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 11.5 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 methods features a set of constants (C1, C2,C3,C4,C5,R1,R2,R3,R4,R5, see Section 11.6). 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

By default, sysmoUSIM-SJS1 acts as both a classic GSM SIM card as well as a USIM on UICC, using an USIM application in parallel to the SIM functionality. This means, that it's up to the phone to decide whether to talk to the card as USIM, or as classic SIM. Typically, modern (3G capable) phones will talk to the card as USIM, and only old, pre-3G phones will talk to it as SIM card.

Please note that the used/supported radio technology (GSM/GPRS/EGPRS vs. UMTS/HSPA/LTE/5G) has no direct relation to which method the phone will use to the smart card. You can very well have a classic GSM SIM card inside a UMTS capable phone while attaching to the UMTS radio network (UTRAN).

In some cases it may become necessary to disable the USIM application. This will basically turn the card into a classic GSM SIM card without any USIM capability.

sysmo-usim-tool can enable and disable the USIM application as described in Section 11.7 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:
   61194f10a0000000871002ffffffff8907090000050055553696d31ffffffffffffffffffffffff
   ==> 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. In order to do this, stop the pcscd daeomn on your system and start it manually using **sudo pcscd -fa**. This will give you a log of the raw traffic between reader and card.

```
$ ./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:00d6000010000000000000000000000000000000000000 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d6001010000000000000000000000000000000000001 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d6002010000000000000000000000000000000000002 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d6003010000000000000000000000000000000000004 ==> APDU:(no data) SW ↩
       :9000
   Card transaction: APDU:00d6004010000000000000000000000000000000000008 ==> 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
```

7.3 sysmo-isim-tool (sysmoISIM-SJA2)

In cases where fine-tuning of sysmoISIM parameters is needed, sysmo-isim-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-isim-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.

7.3.1 OPC value

With sysmo-isim-tool the content of EF.OPC (see Section 11.4). 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 3.4.1)

```
$ ./sysmo-isim-tool.sja2.py --adm1 55538407 -o
sysmoISIM-SJA2 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-isim-tool.sja2.py --adm1 55538407 -O df3d7f95d27005a5441820a31a020bf6
```

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

```
$ ./sysmo-isim-tool.sja2.py --adm1 55538407 -C df3d7f95d27005a5441820a31a020bf6
```

7.3.2 Ki value

sysmo-isim-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-isim-tool.sja2.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-isim-tool.sja2.py --adm1 55538407 -K 0123456789ABCDEF0123456789ABCDEF
```

7.3.3   ICCID value

The ICCID can not be changed on a sysmoISIM-SJA2 card.

If you require sysmoISIM-SJA2 cards with non-default ICCID values, please contact sysmocom for factory-programming them with the specific values you require.

7.3.4   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-isim-tool.sja2.py --adm1 55538407 -t
sysmoISIM-SJA2 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: 0x3
   3G: 0x1
```

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 11.5 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-isim-tool.sja2.py --adm1 55538407 -T 6:8
```

7.3.5   Milenage parameters (Ci/Ri)

The milenage authentication methods features a set of constants (C1, C2,C3,C4,C5,R1,R2,R3,R4,R5, see Section 11.6). To read the current configuration from the card, command line option **-l** (**--milenage**) can be used:

```
$ ./sysmo-isim-tool.sja2.py --adm1 55538407 -l
sysmoUSIM-SJA2 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-isim-tool.sja2.py --adm1 55538407 -L \
00000000000000000000000000000000\
00000000000000000000000000000001\
00000000000000000000000000000002\
00000000000000000000000000000004\
00000000000000000000000000000008\
40:00:20:40:60
```

7.3.6 Enable / Disable USIM Application

By default, sysmoISIM-SJA2 acts as both a classic GSM SIM card as well as a USIM on UICC, using an USIM application in parallel to the SIM functionality. This means, that it's up to the phone to decide whether to talk to the card as USIM, or as classic SIM. Typically, modern (3G capable) phones will talk to the card as USIM, and only old, pre-3G phones will talk to it as SIM card.

Please note that the used/supported radio technology (GSM/GPRS/EGPRS vs. UMTS/HSPA/LTE/5G) has no direct relation to which method the phone will use to the smart card. You can very well have a classic GSM SIM card inside a UMTS capable phone while attaching to the UMTS radio network (UTRAN).

In some cases it may become necessary to disable the USIM application. This will basically turn the card into a classic GSM SIM card without any USIM capability.

There are two steps needed to disable the USIM applicaiton:

• removing it from the directory `EF.DIR` (done by `pySim-shell` or `sysmo-isim-tool.sja2.py`)

• locking the USIM card application (done via `GlobalPlatformPro`)

USIM record in EF.DIR

sysmo-isim-tool can add and remove the USIM application from EF.DIR as described in Section 11.7 In order to remove the USIM entryu, record No.1, which contains the USIM AID, will be overwritten with 0xFF. To add the USIM application again. **sysmo-isim-tool** restores the factory default to record No.1.

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

```
$ ./sysmo-isim-tool.sja2.py --adm1 55538407 -m
sysmoISIM-SJA2 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:
   61194f10a0000000871002ffffffff890709000050055553696d31ffffffffffffffffffffffff
   ==> 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-isim-tool.sja2.py --adm1 55538407 -c
```

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

```
$ ./sysmo-isim-tool.sja2.py --adm1 55538407 -u
```

Disabling the USIM application

Removing the EF.DIR record alone may not be sufficient for all phones to stop using the card as a USIM. Some phones/modems might try to blindly select the USIM application, even though it is not listed in EF.DIR anymore.

In order to make it impossible to even select the USIM application, it can be locked via the GlobalPlatform SET STATUS command, which must be secured with the SCP02 protocol.

There is an open source tool available supporting this: GlobalPlatformPro available from https://github.com/martinpaljak/-GlobalPlatformPro

You can use the following command to lock the USIM application:

```
java -jar ./gp.jar --key-enc KIC1❶ --key-mac KID1 --key-dek KIK1 --lock-applet  ↩
    A0000000871002FFFFFFFF8907090000
```

❶   you must substitute the KIC1, KID1, and KIK1 parameters with the card-specific KIC1, KID1 and KIK1 key material for your specific card. Those values are provided by sysmocom together with the ADM1 key by e-mail to the person placing the order in the webshop.

Likewise, you can re-enable the USIM application using

```
java -jar ./gp.jar --key-enc KIC1❶ --key-mac KID1 --key-dek KIK1 --unlock-applet  ↩
    A0000000871002FFFFFFFF8907090000
```

❶   you must substitute the KIC1, KID1, and KIK1 parameters with the card-specific KIC1, KID1 and KIK1 key material for your specific card. Those values are provided by sysmocom together with the ADM1 key by e-mail to the person placing the order in the webshop.

7.3.7 APDU trace

In case of problems it may be helpful to trace the exact APDU commands which are exchanged with between card and reader. In order to do this, stop the pcscd daemon on your system and start it manually using **sudo pcscd -fa**. This will give you a log of the raw traffic between reader and card.

```
$ ./sysmo-isim-tool.sja2.py --adm1 55538407 -L \
sysmoUSIM-SJA2 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:00d6000010000000000000000000000000000000000000 ==> APDU:(no data) SW ←
      :9000
   Card transaction: APDU:00d6001010000000000000000000000000000000000001 ==> APDU:(no data) SW ←
      :9000
   Card transaction: APDU:00d6002010000000000000000000000000000000000002 ==> APDU:(no data) SW ←
      :9000
   Card transaction: APDU:00d6003010000000000000000000000000000000000004 ==> APDU:(no data) SW ←
      :9000
   Card transaction: APDU:00d6004010000000000000000000000000000000000008 ==> 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 Disabling / Enabling SQN verification

A USIM as specified by 3GPP performs a *SQN freshness check* at every authentication. This ensures that an attacker cannot re-play old authentication vectors.

In some exceptional circumstances in laboratory testing / development you may want to create a USIM that does not check the SQN freshness upon authentication.

 WARNING
> This disables a major security feature of the 3G/4G/5G authentication and key agreement. Without SQN check, there is no protection against authentication replay attacks! Only use if you really know what you're doing, and only in a lab.

8.1 pySim-shell.py

In order to disable the SQN check on a sysmoISIM-SJA2 using `pySim-shell`, please follow the below steps

First, we open the card in pySim-shell and verify the ADM1 pin as well as navigate to `ADF.USIM/EF.USIM_SQN`

```
$ ./pySim-shell.py -p 0 ❶
Using PC/SC reader interface
Waiting for card...
Autodetected card type: sysmoISIM-SJA2
Info: Card is of type: UICC-SIM
AIDs on card:
 USIM: a0000000871002ffffffff8907090000 (EF.DIR)
 ISIM: a0000000871004ffffffff8907090000 (EF.DIR)
 ARA-M: a00000015141434c00
 ADF.ISD: a000000003000000
Detected CardModel: SysmocomSJA2
Welcome to pySim-shell!
pySIM-shell (MF)> verify_adm 91820128 ❷
select ADF.USIM/EF.USIM_SQN
null
pySIM-shell (MF/ADF.USIM/EF.USIM_SQN)>
```

❶   this assumes yor card is inserted into PC/SC reader number 0 (first reader)

❷   make sure to use your card-specific ADM1 value here!

Next, we read the file contents and observe that sqn_check is set to 1 (enabled):

```
pySIM-shell (MF/ADF.USIM/EF.USIM_SQN)> read_binary_decoded
{
    "flag1": {
        "skip_next_sqn_check": 0,
        "delta_max_check": 1,
        "age_limit_check": 0,
        "sqn_check": 1, ❶
        "ind_len": 5
    },
    "flag2": {
        "rfu": 0,
        "dont_clear_amf_for_macs": 0,
        "aus_concealed": 1,
        "autn_concealed": 1
    },
    "delta_max": 8589934592,
    "age_limit": 8589934592,
    "freshness": [
        64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ↩
            0, 0, 0, 0, 0
    ]
}
```

❶   This is the line indicating that SQN checking is currently enabled

Now, we use a json-path expression to set the sqn_check field to 0, and re-reads the file contents to verify:

```
pySIM-shell (MF/ADF.USIM/EF.USIM_SQN)> update_binary_decoded --json-path $.flag1.sqn_check  ↩
    0 ❶
"45030002000000000000200000000000000000400000000000000000000000000000000000000000000000000000000000
pySIM-shell (MF/ADF.USIM/EF.USIM_SQN)> read_binary_decoded
{
    "flag1": {
        "skip_next_sqn_check": 0,
        "delta_max_check": 1,
        "age_limit_check": 0,
        "sqn_check": 0, ❷
        "ind_len": 5
    },
    "flag2": {
        "rfu": 0,
        "dont_clear_amf_for_macs": 0,
        "aus_concealed": 1,
        "autn_concealed": 1
    },
    "delta_max": 8589934592,
    "age_limit": 8589934592,
    "freshness": [
        64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  ↩
            0, 0, 0, 0, 0
    ]
}
```

❶   The command matches the JSON document (as seen in `read_binary_decoded` above), and uses the json-path
        `$.flag1.sqn_check` to identify an element within. Finally, the existing element value is replaced with the new
        value `0` given on the command line.

❷   We can now observe that the `sqn_check` parameter has been set to `0` (disabled)

>   **NOTE**
>   For the ISIM application (if you use it), the same procedure should be repeated for `ADF.ISIM/EF.ISIM_SQN`

9   USIM cards and 5G

5G / NR is fully backwards compatible in terms of SIM cards. There is no strict requirement for a USIM to know about 5G or to
have any specific additional functionality.

USIMs were first introduced with 3G/UMTS in Release 99. When 4G and 5G were introduced, 3GPP made sure old cards would
continue to work on new networks.

Nevertheless, there are plenty of additional / new *optional* files that 3GPP specified in later releases to support particularly more
advanced use cases.

The sysmoISIM-SJA2 contains all of the optional files for 5G up to Release 16.

The specific functionality of a USIM card regarding the use in 5G networks is fully standardized by 3GPP. There is nothing
specific to the sysmoISIM about this.

There are a number of 5G related parameters on a USIM, primarily in the files in DF.5GS. Those files are all optional. As
sysmocom products are used by many researchers and developers, the sysmoISIM-SJA2 includes *all* of those optional files.
Without those files present on the card, for example the important privacy feature "SUCI" could not be used.

The predecessor product (sysmoUSIM-SJS1) does not contain the optional DF.5GS or the optional files beneath it, as the product predates the development of 5G. A key aspect of introducing the successor was to add support for all of those 5G related use cases, as well as advanced IMS use cases requiring a separate ISIM application on the card.

When those 5G related files exist *and* they are active, they need to contain valid data. sysmocom is not the operator of your 5G network, so as the SIM card manufacturer sysmocom has no idea what kind of parameters your specific network in its specific configuration supports. It is up to you as the network operator to ensure they contain configuration consistent with your network and its configuration.

The contents of each of those files is specified by [3gpp-ts-31-102] Section 4.4.11 *Contents of files at the DF_5GS level*. The commands for reading and writing (such as READ BINARY, READ RECORD, WRITE BINARY, WRITE RECORD) are fully specified in [etsi-ts102221].

9.1   Disabling DF_5GS files and services

If you prefer to not populate those files with configuration data consistent to your network, you can disable those files. To do so, you would have to:

1. disable the related services from EF.UST (Services no. 122 through no. 130)

2. if that's insufficient (some UE may ignore the EF.UST contents and "blindly" try to read a file), deactive all of the files below relevant files using the DEACTIVATE FILE command as specified in [etsi-ts102221] Section 11.1.14.

sysmocom is selling SIM cards that are compliant with the relevant 3GPP, ETSI and ISO specifications. It is not selling any associated software.

You can use whatever software conforming to the above-mentioned 3GPP and ETSI standards to configure the cards.

The open source `pySim` software implements support for [almost] all of the thousands of settings over hundreds of files that can be configured on SIM/USIM/ISIM cards. It is an open source collaborative project and not something that is part of the product you bought from sysmocom.

Irrespective of the above, sysmocom has been and continues to contribute many man-months of development resources to improve pySim capabilities. But it is a collaborative open source project, and anyone can, if they have a related requirement, submit patches to improve and extend it.

pySim-shell versions after May 2021 contain support for the to operations described above:

- the `deactivate_file` command to deactivate the currently selected file

- the `ust_service_deactivate` command to deactivate services from EF.UST.

Furthermore, from pySim gc89a1a9 (February 17, 2022) onwards, there is a script included at the `scripts/deactivate-5g.scri` which will perform deactivation of all 5G related EF.UST services and deactivation of all 5G related files.

Example pySim-shell command for deactivating 5G services and files

```
$ ./pySim-shell.py -p0 --script ./scripts/deactivate-5g.script
```

Note
The above command only works if you either edit the script to insert your card-individual ADM1 pin, or you store the ADM1 in a CSV file accessible to pySim (see the `verify_adm` section in the pySim-shell documentation)

Should you later decide you need to re-enable some services and files, you can similarly use the `activate_file` and `ust_service_activate` commands to achieve the inverse operations.

If you're new to `pySim-shell`, the video recording of a recent Osmocom Developer Call available from https://people.osmocom.org/-tnt/osmodevcall/osmodevcall-20210409-laforge-pysim-shell_h264_420.mp4 may be useful.

---

10   USIM/ISIM cards and IMS

IMS, the IP Multimedia System is how circuit-switched calls like voice calls are implemented in 3GPP networks over modern cellular technologies like LTE/4G, NR/5G and WiFi.

IMS can work without any specific configuration or support from the SIM. In this mode, automatic provisioning and default parameters (such as deriving the IMPI from the IMSI) are used.

However, for advanced configuration, there are two ways of providing IMS related configuration on cards:

- via some additional files in `ADF.USIM`, or

- via a completely separate additional `ADF.ISIM` application

The two methods are exclusive of each other: If ADF.ISIM is present, the IMS related files in ADF.USIM must not be present.

The sysmoUSIM-SJA1 has no support for IMS and no related files.

The sysmoISIM-SJA2 contains an ADF.ISIM application and all files specified by 3GPP up to Release 15 (SJA2v1) or Release 16 (SJA2v2).

So if you want to use the sysmoISIM-SJA2 with IMS, you have two options: You can either ensure that the IMS related configuration on the card reflects your network side configuration, or you can disable the ISIM on the card and use the 3GPP default / fallback mechanism.

10.1   Disabling the ISIM functionality

If you prefer to not populate those IMS related files with configuration data consistent to your network, you can disable those files. To do so, you would have to:

1. disable the related services from ADF.USIM/EF.UST

2. deactivate the related files in ADF.USIM

3. remove the ISIM application from EF.DIR

4. lock the entire ISIM application to prevent it from being selected

sysmocom is selling SIM cards that are compliant with the relevant 3GPP, ETSI and ISO specifications. It is not selling any associated software.

You can use whatever software conforming to the above-mentioned 3GPP and ETSI standards to configure the cards.

10.1.1   Deactivating IMS files + services from ADF.USIM

The open source `pySim` software by no means implements support for all of the thousands of settings over hundreds of files that can be configured on SIM/USIM/ISIM cards. It is an open source collaborative project and not something that is part of the product you bought from sysmocom.

Irrespective of the above, sysmocom has been and continues to contribute many man-months of development resources to improve pySim capabilities. But it is a collaborative open source project, and anyone can, if they have a related requirement, submit patches to improve and extend it.

pySim-shell versions after May 2021 contain support for the to operations described above:

- the `deactivate_file` command to deactivate the currently selected file

- the `ust_service_deactivate` command to deactivate services from EF.UST.

Furthermore, from pySim gc89a1a9 (February 21, 2022) onwards, there is a script included at the `scripts/deactivate-ims.scr` which will perform deactivation of all 5G related EF.UST services and deactivation of all IMS related files **in ADF.USIM**.

Example pySim-shell command for deactivating IMS services and files in ADF.USIM

```
$ ./pySim-shell.py -p0 --script ./scripts/deactivate-ims.script
```

It is expected that some commands will fail, for example if the related files did not exist. A non-existent file is like a deactivated file from the perspective of the phone or modem.

Should you later decide you need to re-enable some services and files, you can similarly use the `activate_file` and `ust_service_activate` commands to achieve the inverse operations.

If you're new to `pySim-shell`, the video recording of a recent Osmocom Developer Call available from https://people.osmocom.org/-tnt/osmodevcall/osmodevcall-20210409-laforge-pysim-shell_h264_420.mp4 may be useful.

10.1.2   Locking the ISIM application

Locking the ISIM application will make it inaccessible and it is no longer possible to select that application by the phone/modem. Like any on-card application, locking can be performed via the standard GlobalPlatform `SET STATUS` Command.

GlobalPlatform commands need to be cryptographically protected via the `SCP02` protocol, using the key material provided together with the sysmoISIM-SJA2 card.

An open source tool implementing both `SCP02` and the `SET STATUS` command is `GlobalPlatformPro`, which is available from https://github.com/martinpaljak/GlobalPlatformPro

You can use the following command to lock the ISIM application:

```
java -jar ./gp.jar --key-enc KIC1❶ --key-mac KID1 --key-dek KIK1 --lock-applet  ↩
    A0000000871004FFFFFFFF8907090000
```

❶   you must substitute the KIC1, KID1, and KIK1 parameters with the card-specific KIC1, KID1 and KIK1 key material for your specific card. Those values are provided by sysmocom together with the ADM1 key by e-mail to the person placing the order in the webshop.

Likewise, you can unlock the ISIM application using:

```
java -jar ./gp.jar --key-enc KIC1❶ --key-mac KID1 --key-dek KIK1 --unlock-applet  ↩
    A0000000871004FFFFFFFF8907090000
```

❶   you must substitute the KIC1, KID1, and KIK1 parameters with the card-specific KIC1, KID1 and KIK1 key material for your specific card. Those values are provided by sysmocom together with the ADM1 key by e-mail to the person placing the order in the webshop.

11   APDU-Level Card Provisioning

This section describes how to update common parameters 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.

11.1 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.

11.2 ICCID

11.2.1 sysmoUSIM-SJS1

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.

11.2.2 sysmoISIM-SJA2

The ICCID cannot be changed in the sysmoISIM-SJA2 after factory programming.

11.3 K / Ki

11.3.1 sysmoUSIM-SJS1

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.

11.3.2 sysmoISIM-SJA2

In order to change the Ki/K, simply perform a standard UPDATE BINARY command on the following file(s):

Application	Path	
SIM	A515/6F20	
USIM	ADF_USIM/AF20	
ISIM	ADF_ISIM/AF20	

The Ki/K value is a 16-byte value stored at offset 1.

11.4 OPc or OP

11.4.1 sysmoUSIM-SJS1

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

See Section 3.4.1 on a discussion about whether to store the global OP value or a card-individual pre-computed OPc.

Copyright © 2016-2022 sysmocom - s.f.m.c. GmbH

Offset	Size	Description
0	1	0x00 for OP, 0x01 for OPc (see Section 3.4.1)
1	16	OP or OPc value, depending on byte at offset 0

Note
You need to be authenticated using ADM1 PIN.

11.4.2 sysmoISIM-SJA2

In order to change the OPc or OP, simply perform a standard UPDATE BINARY command on the following files:

Application	Path	
SIM	A515/6F20	
USIM	ADF_USIM/AF20	
ISIM	ADF_ISIM/AF20	

See Section 3.4.1 on a discussion about whether to store the global OP value or a card-individual pre-computed OPc.

Offset	Size	Description
0	1	0x10 for OP, 0x00 for OPc (see Section 3.4.1)
1	16	K/Ki value
17	16	OP or OPc value, depending on byte at offset 0

Note
You need to be authenticated using ADM1 PIN.

11.5 Authentication Algorithms

11.5.1 sysmoUSIM-SJS1

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	COMP128v1	2G
04	XOR 2G	2G
06	COMP128v2	2G
07	COMP128v3	2G
08	XOR 3G	3G

Note
You need to be authenticated using ADM1 PIN.

11.5.2 sysmoISIM-SJA2

In order to change the Authentication Algorithms used, simply perform a standard UPDATE BINARY command on the following file(s):

Application	Path	
SIM	A515/6F20	
USIM	ADF_USIM/AF20	
ISIM	ADF_ISIM/AF20	

The lower nibble of the first octet (offset 0) contains the algorithm identifier:

Value	Algorithm	Supported Mode
1	COMP128v1	SIM
2	COMP128v2	SIM
3	COMP128v3	SIM
4	MILENAGE	SIM+USIM+ISIM
5	SHA1AKA	SIM+USIM+ISIM

11.6 Milenage Configuration (Ci/Ri)

11.6.1 sysmoUSIM-SJS1

The Milenage configuration constants C_i (i=1..5) and R_i (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
C3	00000000000000000000000000000002
C4	00000000000000000000000000000004
C5	00000000000000000000000000000008
R1	0x40
R2	0x00

Parameter	Value
R3	0x20
R4	0x40
R5	0x60

11.6.2   sysmoISIM-SJA2

The Milenage configuration constants Ci (i=1..5) and Ri (i=1..5) can be configured in the following binary files:

Application	Path	
SIM	FIXME	
USIM	ADF_USIM/AF21	
ISIM	ADF_ISIM/AF21	

The structure of the files is as follows:

Offset	Size	Description
0	1	R1
1	1	R2
2	1	R3
3	1	R4
4	1	R5
5	16	C1
21	16	C2
37	16	C3
53	16	C4
69	16	C5

Note
You need to be authenticated using ADM1 PIN.

11.7   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.

12   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).

12.1　Application List

Table 1: List of Java applications on sysmoUSIM-SJS1 card

Application	PID	AID
SIM Application	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF8900000000
SIM RFM Application	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF89B00010
RAM Application	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF89B00010
USIM RFM App (opt)	FF434E52581040040203	FF434E525810400402030000000000
USIM Application	A0000000871002FF49FFFF8900	A0000000871002FF49FFFF89040B0000
CAT_TP App (opt)	A0000000090001FFFFFFFF8900	A0000000090001FFFFFFFF89B00010

Table 2: List of Java applications on sysmoISIM-SJA2 card

Application	AID
USIM	A0000000871002FFFFFFFF8907090000
ISIM	A0000000871004FFFFFFFF8907090000
HPSIM?	A000000087ABCDFFFFFFFF8907090000
Remote Application Management	53696D62614E2E52414D
SIM?	A0000000090001FFFFFFFF8900000000
Remote File Management	53696D62614E2E52464D
ARA-M	A00000015141434C00

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].

12.2　Example Applet

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

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

12.3　Installation via 03.48 OTA

This means the Java Applet will be installed OTA (Over The Air) via SMS messages.

There are open source tools provided at https://gitea.osmocom.org/sim-card/sim-tools which can be used to download the example applet (or other applets) onto the card.

The `shadysim.py` tool will format the Java Applet into small, SMS-sized chunks conformant to the TS 03.48 OTA messages, and then either

1. emulate a phone talking to a SIM card, informing the card about received OTA SMS, or

2. output the SMS as hexdumps so you can use them e.g. by typing into the OsmoMSC or OsmoNITB VTY of a private cellular network, or send them via SMPP or any other SMS delivery method you may have available.

NOTE

Due to a mistake in the SIM card factory producing the first batch of sysmoISIM-SJA2, sysmocom provided the wrong KID1/KIK1 values to customers before January 27, 2021. Should you experience problems successfully establishing OTA or SCP02 communication and have bought your card before that date, please reach out to web-shop@sysmocom.de and request re-sending of the (corrected) key material by e-mail. We apologize for any inconvenience caused.

12.4   Installation via GlobalPlatform SCP02

If you have the SIM card in a local card reader attached to your computer, you don't have to use the OTA SMS method as implemented by the sim-tools above.

Instead, you can use the GlobalPlatform / JavaCard SCP02 method of installing applets. This method is completely unrelated to SIM cards or 3GPP.

One commonly used Open Source program for this is the GlobalPlatformPro tool available from https://github.com/martinpaljak/-GlobalPlatformPro

The keys received with the sysmoUSIM-SJS1 or sysmoISIM-SJA2 have to be used like this:

Table 3: Key mapping for GlobalPlatformPro

sysmoUSIM key	GlobalPlatformPro argument
KIC1	--key-enc
KID1	--key-mac
KIK1	--key-dek

NOTE

Due to a mistake in the SIM card factory producing the first batch of sysmoISIM-SJA2, sysmocom provided the wrong KID1/KIK1 values to customers before January 27, 2021. Should you experience problems successfully establishing OTA or SCP02 communication and have bought your card before that date, please reach out to webshop@sysmocom.de and request re-sending of the (corrected) key material by e-mail. We apologize for any inconvenience caused.

13   OTA (Over The Air)

Using OTA, the operator can communicate from a backend system (so-called "OTA platform") with software on the SIM card itself.

The OTA protocol / framework was first described in GSM TS 03.48, later.

13.1   Transports

OTA can happen over a number of different transports * SMS * SMSCB (Cell Broadcast) * USSD * BIP * CAT-TP * HTTPS

SMS is by far the most common mechanism for OTA access to SIM cards, including the sysmoISIM-SJA2.

13.2 TAR (Toolkit Application Reference)

Individual applications on the card can be identified by their TAR (Toolkit Application Reference) value.

The following table lists the TAR values for the sysmoISIM-SJA2 and sysmoISIM-SJA5:

Table 4: List of TAR and MSL for sysmoISIM-SJA2 and sysmoISIM-SJA5 card

Application	TAR	MSL
RAM (Remote Application Management)	00000	0x06
SIM RFM (Remote File Management)	B00010	0x06
USIM RFM (Remote File Management)	B00011	0x06
ISIM RFM (Remote File Management)	B00020	0x06
HPSIM RFM (Remote File Management)	B00021	0x06

13.3 MSL (Minimum Security Level)

The MSL (Minimum Security Level) defines which of the OTA security features are minimally mandatory for the card to accept an OTA command.

See the above table for the MSL values of the individual OTA applications on the card.

A MSL of 0x06 (sysmoISIM-SJA2 and sysmoISIM-SJA5 default value) indicates that both integrity protection and encryption shall be used. Counter based replay protection is possible, but not mandatory.

14 sysmoISIM-SJA2 changelog

This chapter documents the changes to the sysmoISIM-SJA2 product over time.

14.1 sysmoISIM-SJA2 v1 (September 2020)

- Initial version of sysmoISIM-SJA2 product

- Files as of 3GPP Release 15

- ISIM application as per 3GPP TS 31.103

14.2 sysmoISIM-SJA2 v2 (September 2021)

- Change from default MCC-MNC 901-70 to MCC-MNC 999-70, as 901-70 was re-assigned by ITU to a public operator (Clementvale Baltic OY) and with 999 there is a proper MCC for private cellular networks now.

- Update from 3GPP Release 15 filesystem to include all files of 3GPP Release 16

- Introduce support for BER-TLV files

- Fix EF.UMPC: disable indicating SUSPEND support

- Mark DF.TELECOM as LINKABLE

- Fix EF.MCSCONFIG, EF.V2X_CONFIG to be BER-TLV EF, not transparent EF

- New DF.V2X/EF.V2XP_PC5 and EF.V2XXP_Uu files

- Fix EF.CFIS: work around UE that read it despite EF.SST/EF.UST not indicating CFIS support

- EF.UST: Disable service 123 (SUCI calc info) by default, most users are not aware how to configure it

- EF.AD: set the bit that allows powering down the card during eDRX cycles

- New EF.WRI, EF.HWSIDL, EF.WEHPLMNPI, EF.WHPI, EF.WLRPLMN, EF.HPLMNDAI

- New DF.ProSe files EF.PROSE_MON, EF.PROSE_ANN, EF.PROSE_FUNC, EF.PROSE_RADIO_COM, EF.PROSE_RADIO_MON, EF.PROSE_RADIO_ANN, EF.PROSE_POLICY, EF.PROSE_PLMN, EF.PROSE_GC, EF.PST, EF.PROSE_UIRC, EF.PROSE_GM_D, EF.PROSE_RELAY, EF.PROSE_RELAY_DISCOVERY

- New DF.ACDC files EF.ACDC_LIST, EF.ACDC_OS_CONFIG90

- Enlarge EF.5GAUTHKEYS from 68 to 110 bytes

- Rename EF.NSI to EF.SUPI_NAI, enlarge from 100 to 200 bytes

- New EF.URSP and EF.TN3GPPPSNN

- Update EF.IST: FIXME

- New ADF.ISIM file: EF.WebRTCURI

Commercially, sysmocom decided to remove the "without ADM keys" option from active sales, as the amount of inquiries generated by people ordering the wrong variant was creating a lot of extra effort - defeating the original idea of having a lower-cost product that creates less support effort.

15 sysmoISIM-SJA2 v2.1 (May 2023)

- Reduce EF.VGCS and EF.VBS size from 400 to 200 bytes

- Allow creation of arbitrary files below ADF.USIM

16 Acknowledgements

sysmocom would like to thank a number of individuals in the context of improving the availability of freely available programmable SIM cards and related tools

- **Sylvain Munaut** for developing the original `pySim` tool

- **Philipp Maier** for developing the `sysmo-usim-tool`

- **Benoit Michau** for the python `card` abstraction library

- **Kevin Redon** for `Osmocom SIMtrace`

- **Eric Butler** and **Karl Koscher** of shadytel for their hello world Java cardlet and the `sim-tools` for OTA installation

- **Supreeth Herle** for all of his research on the role of SIM cards in VoLTE/IMS, CarrierPrivileges and many related contributions to `pySim`

- **Bertrand Martel** for his open source implementation `ARA-M` applet, which we also pre-install on the sysmoISIM-SJA2

- **Martin Paljak** for his work on `GlobalPlatformPro`

17 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

AQPSK

    Adaptive QPSK, a modulation scheme used by VAMOS channels on Downlink

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

CBC

Cell Broadcast Centre; central entity of Cell Broadcast service

CBCH

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

CBS

Cell Broadcast Service

CBSP

Cell Broadcast Service Protocol (*3GPP TS 48.049* [3gpp-ts-48-049])

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

CSFB

Circiut-Switched Fall Back; Mechanism for switching from LTE/EUTRAN to UTRAN/GERAN when circuit-switched services such as voice telephony are required.

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

DSCP

Differentiated Services Code Point (*IETF RFC 2474* [ietf-rfc2474])

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

EIR

Equipment Identity Register; core network element that stores and manages IMEI numbers

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

GPRS

General Packet Radio Service; the packet switched 2G technology

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

GSUP

Generic Subscriber Update Protocol. Osmocom-specific alternative to TCAP/MAP

GT

Global Title; an address in SCCP

GTP

GPRS Tunnel Protocol; used between SGSN and GGSN

HLR

Home Location Register; central subscriber database of a GSM network

HNB-GW

Home NodeB Gateway. Entity between femtocells (Home NodeB) and CN in 3G/UMTS.

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 14-digit decimal number to globally identify a mobile device, optionally with a 15th checksum digit

IMEISV

IMEI software version; unique 14-digit decimal number to globally identify a mobile device (same as IMEI) plus two software version digits (total digits: 16)

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* [ietf-rfc791])

IPA

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

Iu

Interface in 3G/UMTS between RAN and CN

IuCS

Iu interface for circuit-switched domain. Used in 3G/UMTS between RAN and MSC

IuPS

Iu interface for packet-switched domain. Used in 3G/UMTS between RAN and SGSN

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

LU

Location Updating; can be of type IMSI-Attach or Periodic. Procedure that indicates a subscriber's physical presence in a given radio cell.

M2PA

MTP2 Peer-to-Peer Adaptation; a SIGTRAN Variant (*RFC 4165* [ietf-rfc4165])

M2UA

MTP2 User Adaptation; a SIGTRAN Variant (*RFC 3331* [ietf-rfc3331])

M3UA

MTP3 User Adaptation; a SIGTRAN Variant (*RFC 4666* [ietf-rfc4666])

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

MNCC

Mobile Network Call Control; Unix domain socket based Interface between MSC and external call control entity like osmo-sip-connector

MNO

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

MO

Mobile Originated. Direction from Mobile (MS/UE) to Network

MS

Mobile Station; a mobile phone / GSM Modem

MSC

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

MSC pool

A number of redundant MSCs serving the same core network, which a BSC / RNC distributes load across; see also the "MSC Pooling" chapter in OsmoBSC's user manual [userman-osmobsc] and *3GPP TS 23.236* [3gpp-ts-23-236]

MSISDN

Mobile Subscriber ISDN Number; telephone number of the subscriber

MT

Mobile Terminated. Direction from Network to Mobile (MS/UE)

MTP

Message Transfer Part; SS7 signaling protocol (*ITU-T Q.701* [itu-t-q701])

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

NRI

Network Resource Indicator, typically 10 bits of a TMSI indicating which MSC of an MSC pool attached the subscriber; see also the "MSC Pooling" chapter in OsmoBSC's user manual [userman-osmobsc] and *3GPP TS 23.236* [3gpp-ts-23-236]

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

PC

Point Code; an address in MTP

PCH

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

PCP

Priority Code Point (*IEEE 802.1Q* [?])

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

SCCP

Signaling Connection Control Part; SS7 signaling protocol (*ITU-T Q.711* [itu-t-q711])

SDCCH

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

SDK

Software Development Kit

SGs

Interface between MSC (GSM/UMTS) and MME (LTE/EPC) to facilitate CSFB and SMS.

SGSN

Serving GPRS Support Node; Core network element for packet-switched services in GSM and UMTS.

SIGTRAN

Signaling Transport over IP (*IETF RFC 2719* [ietf-rfc2719])

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

SS7

Signaling System No. 7; Classic digital telephony signaling system

SS

Supplementary Services; query and set various service parameters between subscriber and core network (e.g. USSD, 3rd-party calls, hold/retrieve, advice-of-charge, call deflection)

SSH

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

SSN

Sub-System Number; identifies a given SCCP Service such as MSC, HLR

STP

Signaling Transfer Point; A Router in SS7 Networks

SUA

SCCP User Adaptation; a SIGTRAN Variant (*RFC 3868* [ietf-rfc3868])

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])

TOS

Type Of Service; bit-field in IPv4 header, now re-used as DSCP (*IETF RFC 791* [ietf-rfc791])

TRX

Transceiver; element of a BTS serving a single carrier

TS

Technical Specification

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

USSD

Unstructured Supplementary Service Data; textual dialog between subscriber and core network, e.g. **100 → Your extension is 1234*

VAMOS

Voice services over Adaptive Multi-user channels on One Slot; an optional extension for GSM specified in Release 9 of 3GPP GERAN specifications (*3GPP TS 48.018* [3gpp-ts-48-018]) allowing two independent UEs to transmit and receive simultaneously on traffic channels

VCTCXO

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

VLAN

Virtual LAN in the context of Ethernet (*IEEE 802.1Q* [ieee-802.1q])

VLR

Visitor Location Register; volatile storage of attached subscribers in the MSC

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 5: TCP/UDP port numbers

L4 Protocol	Port Number	Purpose	Software
UDP	1984	Osmux	osmo-mgw, osmo-bts
UDP	2427	MGCP GW	osmo-bsc_mgcp, osmo-mgw
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	4227	telnet (VTY)	osmo-pcap-client
TCP	4228	telnet (VTY)	osmo-pcap-server
TCP	4236	Control Interface	osmo-trx
TCP	4237	telnet (VTY)	osmo-trx
TCP	4238	Control Interface	osmo-bts
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, osmo-mgw
TCP	4244	telnet (VTY)	osmo-bsc_nat
TCP	4245	telnet (VTY)	osmo-sgsn
TCP	4246	telnet (VTY)	osmo-gbproxy
TCP	4247	telnet (VTY)	OsmocomBB
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

Table 5: (continued)

L4 Protocol	Port Number	Purpose	Software
TCP	4255	Control Interface	osmo-msc
TCP	4256	telnet (VTY)	osmo-sip-connector
TCP	4257	Control Interface	osmo-ggsn, ggsn (OpenGGSN)
TCP	4258	telnet (VTY)	osmo-hlr
TCP	4259	Control Interface	osmo-hlr
TCP	4260	telnet (VTY)	osmo-ggsn
TCP	4261	telnet (VTY)	osmo-hnbgw
TCP	4262	Control Interface	osmo-hnbgw
TCP	4263	Control Interface	osmo-gbproxy
TCP	4264	telnet (VTY)	osmo-cbc
TCP	4265	Control Interface	osmo-cbc
TCP	4266	D-GSM MS Lookup: mDNS serve	osmo-hlr
TCP	4267	Control Interface	osmo-mgw
TCP	4268	telnet (VTY)	osmo-uecups
SCTP	4268	UECUPS	osmo-uecups
TCP	4269	telnet (VTY)	osmo-e1d
TCP	4270	telnet (VTY)	osmo-isdntap
TCP	4271	telnet (VTY)	osmo-smlc
TCP	4272	Control Interface	osmo-smlc
TCP	4273	telnet (VTY)	osmo-hnodeb
TCP	4274	Control Interface	osmo-hnodeb
TCP	4275	telnet (VTY)	osmo-upf
TCP	4276	Control Interface	osmo-upf
TCP	4277	telnet (VTY)	osmo-pfcp-tool
TCP	4278	Control Interface	osmo-pfcp-tool
UDP	4729	GSMTAP	Almost every osmocom project
TCP	5000	A/IP	osmo-bsc, osmo-bsc_nat
UDP	23000	GPRS-NS over IP default port	osmo-pcu, osmo-sgsn, osmo-gbproxy
TCP	48049	BSC-CBC (CBSP) default port	osmo-bsc, osmo-cbc

B Bibliography / References

References

[1] [userman-ice1usb] Osmocom Project: icE1usb User Manual.

[2] [userman-ogt] Pau Espin: osmo-gsm-tester User Manual.

[3] [userman-remsim] Harald Welte: osmo-remsim User Manual.

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

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

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

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

[8] [vty-ref-osmobts] Osmocom Project: OsmoBTS VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmobts-trx-vty-reference.pdf   https://ftp.osmocom.org/docs/latest/osmobts-sysmo-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-lc15-vty-reference.pdf   https://ftp.osmocom.org/docs/latest/-osmobts-oc2g-vty-reference.pdf   https://ftp.osmocom.org/docs/latest/osmobts-octphy-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-virtual-vty-reference.pdf

[9] [userman-osmocbc] Osmocom Project: OsmoCBC User Manual. https://ftp.osmocom.org/docs/latest/-osmocbc-usermanual.pdf

[10] [vty-ref-osmocbc] Osmocom Project: OsmoCBC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmocbc-vty-reference.pdf

[11] [userman-osmogbproxy] Osmocom Project: OsmoGBProxy User Manual. https://ftp.osmocom.org/docs/-latest/osmogbproxy-usermanual.pdf

[12] [vty-ref-osmogbproxy] Osmocom Project: OsmoGBPRoxy VTY Reference Manual. https://ftp.osmocom.org/-docs/latest/osmogbproxy-vty-reference.pdf

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

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

[15] [userman-osmohlr] Osmocom Project: OsmoHLR User Manual. https://ftp.osmocom.org/docs/latest/osmohlr-usermanual.pdf

[16] [vty-ref-osmohlr] Osmocom Project: OsmoHLR VTY Reference Manual. https://ftp.osmocom.org/docs/latest/-osmohlr-vty-reference.pdf

[17] [userman-osmohnbgw] Osmocom Project: OsmoHNBGW User Manual. https://ftp.osmocom.org/docs/latest/-osmohnbgw-usermanual.pdf

[18] [vty-ref-osmohnbgw] Osmocom Project: OsmoHNBGW VTY Reference Manual. https://ftp.osmocom.org/-docs/latest/osmohnbgw-vty-reference.pdf

[19] [userman-osmomgw] Osmocom Project: OsmoMGW User Manual. https://ftp.osmocom.org/docs/latest/-osmomgw-usermanual.pdf

[20] [vty-ref-osmomgw] Osmocom Project: OsmoMGW VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmomgw-vty-reference.pdf

[21] [userman-osmomsc] Osmocom Project: OsmoMSC User Manual. https://ftp.osmocom.org/docs/latest/-osmomsc-usermanual.pdf

[22] [vty-ref-osmomsc] Osmocom Project: OsmoMSC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmomsc-vty-reference.pdf

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

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

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

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

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

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

[29] [userman-osmosipconnector] Osmocom Project: OsmoSIPconnector User Manual. https://ftp.osmocom.org/-docs/latest/osmosipconnector-usermanual.pdf

[30] [vty-ref-osmosipconnector] Osmocom Project: OsmoSIPconnector VTY Reference Manual. https://ftp.osmocom.org/docs/latest/osmosipconnector-vty-reference.pdf

[31] [userman-osmosmlc] Osmocom Project: OsmoSMLC User Manual. https://ftp.osmocom.org/docs/latest/-osmosmlc-usermanual.pdf

[32] [vty-ref-osmosmlc] Osmocom Project: OsmoSMLC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmosmlc-vty-reference.pdf

[33] [userman-osmostp] Osmocom Project: OsmoSTP User Manual. https://ftp.osmocom.org/docs/latest/osmostp-usermanual.pdf

[34] [vty-ref-osmostp] Osmocom Project: OsmoSTP VTY Reference Manual. https://ftp.osmocom.org/docs/latest/-osmostp-vty-reference.pdf

[35] [userman-osmotrx] Osmocom Project: OsmoTRX User Manual. https://ftp.osmocom.org/docs/latest/osmotrx-usermanual.pdf

[36] [vty-ref-osmotrx] Osmocom Project: OsmoTRX VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmotrx-uhd-vty-reference.pdf   https://ftp.osmocom.org/docs/latest/osmotrx-lms-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmotrx-ipc-vty-reference.pdf   https://ftp.osmocom.org/docs/latest/-osmotrx-usrp1-vty-reference.pdf

[37] [3gpp-ts-23-041] 3GPP TS 23.041: Technical realization of Cell Broadcast Service (CBS)

[38] [3gpp-ts-23-048] 3GPP TS 23.048: Security mechanisms for the (U)SIM application toolkit; Stage 2 https://www.3gpp.org/DynaReport/23048.htm

[39] [3gpp-ts-23-236] 3GPP TS 23.236: Intra-domain connection of Radio Access Network (RAN) nodes to multiple Core Network (CN) nodes https://www.3gpp.org/DynaReport/23236.htm

[40] [3gpp-ts-24-007] 3GPP TS 24.007: Mobile radio interface signalling layer 3; General Aspects https://www.3gpp.org/DynaReport/24007.htm

[41] [3gpp-ts-24-008] 3GPP TS 24.008: Mobile radio interface Layer 3 specification; Core network protocols; Stage 3. https://www.3gpp.org/dynareport/24008.htm

[42] [3gpp-ts-31-101] 3GPP TS 31.101: UICC-terminal interface; Physical and logical characteristics https://www.3gpp.org/DynaReport/31101.htm

[43] [3gpp-ts-31-102] 3GPP TS 31.102: Characteristics of the Universal Subscriber Identity Module (USIM) application https://www.3gpp.org/DynaReport/31102.htm

[44] [3gpp-ts-31-103] 3GPP TS 31.103: Characteristics of the IMS Subscriber Identity Module (ISIM) application https://www.3gpp.org/DynaReport/31103.htm

[45] [3gpp-ts-31-111] 3GPP TS 31.111: Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) https://www.3gpp.org/DynaReport/31111.htm

[46] [3gpp-ts-31-115] 3GPP TS 31.115: Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications https://www.3gpp.org/DynaReport/31115.htm

[47] [3gpp-ts-31-116] 3GPP TS 31.116: Remote APDU Structure for (U)SIM Toolkit applications https://www.3gpp.org/DynaReport/31116.htm

[48] [3gpp-ts-35-205] 3GPP TS 35.205: 3G Security; Specification of the MILENAGE algorithm set: General

[49] [3gpp-ts-35-206] 3GPP TS 35.206: 3G Security; Specification of the MILENAGE algorithm set: Algorithm specification https://www.3gpp.org/DynaReport/35206.htm

[50] [3gpp-ts-44-006] 3GPP TS 44.006: Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification https://www.3gpp.org/DynaReport/44006.htm

[51] [3gpp-ts-44-018] 3GPP TS 44.018: Mobile radio interface layer 3 specification; Radio Resource Control (RRC) protocol https://www.3gpp.org/DynaReport/44018.htm

[52] [3gpp-ts-44-064] 3GPP TS 44.064: Mobile Station - Serving GPRS Support Node (MS-SGSN); Logical Link Control (LLC) Layer Specification https://www.3gpp.org/DynaReport/44064.htm

[53] [3gpp-ts-45-002] 3GPP TS 45.002: Digital cellular telecommunications system (Phase 2+) (GSM); GSM/EDGE Multiplexing and multiple access on the radio path https://www.3gpp.org/DynaReport/45002.htm

[54] [3gpp-ts-48-008] 3GPP TS 48.008: Mobile Switching Centre - Base Station system (MSC-BSS) interface; Layer 3 specification https://www.3gpp.org/DynaReport/48008.htm

[55] [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 https://www.3gpp.org/DynaReport/48016.htm

[56] [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) https://www.3gpp.org/DynaReport/48018.htm

[57] [3gpp-ts-48-049] 3GPP TS 48.049: Digital cellular communications system; Base Station Controller - Cell Broadcast Centre (BSC-CBC) interface specification; Cell Broadcast Service Protocol (CBSP) https://www.3gpp.org/DynaReport/48049.htm

[58] [3gpp-ts-48-056] 3GPP TS 48.056: Base Station Controller - Base Transceiver Station (BSC - BTS) interface; Layer 2 specification https://www.3gpp.org/DynaReport/48056.htm

[59] [3gpp-ts-48-058] 3GPP TS 48.058: Base Station Controller - Base Transceiver Station (BSC - BTS) Interface; Layer 3 specification https://www.3gpp.org/DynaReport/48058.htm

[60] [3gpp-ts-51-011] 3GPP TS 51.011: Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface

[61] [3gpp-ts-51-014] 3GPP TS 51.014: Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface https://www.3gpp.org/DynaReport/51014.htm

[62] [3gpp-ts-52-021] 3GPP TS 52.021: Network Management (NM) procedures and messages on the A-bis interface https://www.3gpp.org/DynaReport/52021.htm

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

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

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

[66] [ieee-802.1q] IEEE 802.1Q: Bridges and Bridged Networks https://ieeexplore.ieee.org/document/6991462

[67] [ietf-rfc768] IETF RFC 768: User Datagram Protocol https://tools.ietf.org/html/rfc768

[68] [ietf-rfc791] IETF RFC 791: Internet Protocol https://tools.ietf.org/html/rfc791

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

[70] [ietf-rfc1035] IETF RFC 1035: Domain Names - Implementation and Specification https://tools.ietf.org/html/-rfc1035

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

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

[73] [ietf-rfc2474] IETF RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv44 and IPv6 Headers https://tools.ietf.org/html/rfc2474

[74] [ietf-rfc2719] IETF RFC 2719: Signal Transport over IP https://tools.ietf.org/html/rfc2719

[75] [ietf-rfc3331] IETF RFC 3331: Message Transfer Part 2 User Adaptation Layer https://tools.ietf.org/html/-rfc3331

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

[77] [ietf-rfc3596] IETF RFC 3596: DNS Extensions to Support IP Version 6 https://tools.ietf.org/html/rfc3596

[78] [ietf-rfc3868] IETF RFC 3868: SCCP User Adaptation Layer https://tools.ietf.org/html/rfc3868

[79] [ietf-rfc4165] IETF RFC 4165: Message Transfer Part 2 Peer-to-Peeer Adaptation Layer https://tools.ietf.org/-html/rfc4165

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

[81] [ietf-rfc4666] IETF RFC 4666: Message Transfer Part 3 User Adaptation Layer https://tools.ietf.org/html/-rfc4666

[82] [ietf-rfc5771] IETF RFC 5771: IANA Guidelines for IPv4 Multicast Address Assignments https://tools.ietf.org/html/rfc5771

[83] [itu-t-q701] ITU-T Q.701: Functional Description of the Message Transfer Part (MTP) https://www.itu.int/rec/-T-REC-Q.701/en/

[84] [itu-t-q711] ITU-T Q.711: Functional Description of the Signalling Connection Control Part https://www.itu.int/rec/T-REC-Q.711/en/

[85] [itu-t-q713] ITU-T Q.713: Signalling connection control part formats and codes https://www.itu.int/rec/T-REC-Q.713/en/

[86] [itu-t-q714] ITU-T Q.714: Signalling connection control part procedures https://www.itu.int/rec/T-REC-Q.714/en/

[87] [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

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

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

[90] [freeswitch_pbx] FreeSWITCH SIP PBX https://freeswitch.org