## REVISION HISTORY

<table>
<thead>
<tr>
<th>NUMBER</th>
<th>DATE</th>
<th>DESCRIPTION</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>June 2016</td>
<td>Initial</td>
<td>hw</td>
</tr>
<tr>
<td>v2</td>
<td>January 2017</td>
<td>Fix FID for EF.OPC; Add EF.MLNGC; Explain EF.OPC structure; Refer to ETSI TS for AID</td>
<td>hw</td>
</tr>
<tr>
<td>v3</td>
<td>January 2017</td>
<td>Fix name and FID for EF.AUTH; Fix DF path for EF.MLNGC and EF.AUTH</td>
<td>hw</td>
</tr>
<tr>
<td>v4</td>
<td>April 2018</td>
<td>Removed custom IMSI/Ki programming variant</td>
<td>sk</td>
</tr>
<tr>
<td>v4</td>
<td>September 2020</td>
<td>Added sysmoISIM-SJA2 related information</td>
<td>hw</td>
</tr>
<tr>
<td>v5</td>
<td>November 2020</td>
<td>Added form-factor related text and introduction to SIM related identities</td>
<td>hw</td>
</tr>
<tr>
<td>v6</td>
<td>January 2021</td>
<td>Added NOTEs about erroneous SCP02/OTA key material; Acknowledgements</td>
<td>hw</td>
</tr>
<tr>
<td>v7</td>
<td>February 2021</td>
<td>Various typo fixes; add chapter on shared vs. per-application separate keys</td>
<td>hw</td>
</tr>
<tr>
<td>v8</td>
<td>May 2021</td>
<td>Add chapter about 5G services + files</td>
<td>hw</td>
</tr>
<tr>
<td>v9</td>
<td>September 2021</td>
<td>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.</td>
<td>hw</td>
</tr>
<tr>
<td>v10</td>
<td>February 2022</td>
<td>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.</td>
<td>hw</td>
</tr>
</tbody>
</table>
Contents

1 Introduction 1

2 History 1

3 About SIM / USIM Cards 2
   3.1 SIM Cards .................................................................................................................. 2
   3.2 USIM Cards .................................................................................................................. 2
   3.3 ISIM Cards .................................................................................................................... 2
   3.4 Authentication Algorithms ........................................................................................... 3
      3.4.1 OP or OPc in USIM ................................................................................................. 3
      3.4.2 Shared or application-specific keys ......................................................................... 3
   3.5 Interoperability SIM/USIM vs. Network vs Phone ........................................................ 4
      3.5.1 SIM Card vs. Network Technology 2G/3G/4G/5G .................................................. 4
      3.5.2 SIM Card vs. Phone ............................................................................................... 4
   3.6 Card Form Factors ....................................................................................................... 4
      3.6.1 2FF, 3FF, 4FF ......................................................................................................... 4
      3.6.2 MFF2 ...................................................................................................................... 5
   3.7 Identities ....................................................................................................................... 5
      3.7.1 ICCID ..................................................................................................................... 5
      3.7.2 IMSI ....................................................................................................................... 5
      3.7.3 MSISDN ................................................................................................................ 6

4 sysmoUSIM-SJS1 and sysmoISIM-SJA2 specifications 6
   4.1 Physical Specification ................................................................................................. 7
   4.2 Logical Specification ................................................................................................. 7
      4.2.1 ETSI/3GPP/GP Specification Compliance ................................................................ 7
      4.2.2 Default Algorithms ............................................................................................... 8
      4.2.3 PIN Data ............................................................................................................... 8
      4.2.4 Network Authentication ........................................................................................ 8
      4.2.5 OTA Configuration ............................................................................................... 8
      4.2.6 Subscriber Identities ........................................................................................... 8
   4.3 Product Variants ......................................................................................................... 9
      4.3.1 default 999-70 MCC-MNC (no ADM keys) .......................................................... 9
      4.3.2 default 999-70 MCC-MNC (with ADM keys) ....................................................... 9
      4.3.3 user-defined IMSI/Ki (with ADM keys) ................................................................. 9

5 Smart Card Readers 10
   5.1 Verifying your smart card reader + software stack ...................................................... 10
   5.2 Verifying your smart card reader + software stack ...................................................... 10
   5.3 Mechanical Card Adapters ........................................................................................ 11
6 Testing sysmocom SIM cards with utilities

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>osmo-auc-gen</td>
</tr>
<tr>
<td>6.1.1</td>
<td>SYNOPSIS</td>
</tr>
<tr>
<td>6.1.2</td>
<td>OPTIONS</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Example</td>
</tr>
<tr>
<td>6.2</td>
<td>osmo-sim-auth.py</td>
</tr>
<tr>
<td>6.2.1</td>
<td>SYNOPSIS</td>
</tr>
<tr>
<td>6.2.2</td>
<td>OPTIONS</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Example</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Re-Synchronization</td>
</tr>
</tbody>
</table>

7 Provisioning of different identities or keys

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>pySim-prog.py</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Example</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Error Codes</td>
</tr>
<tr>
<td>7.2</td>
<td>sysmo-usim-tool (sysmoUSIM-SJS1)</td>
</tr>
<tr>
<td>7.2.1</td>
<td>OPC value</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Ki value</td>
</tr>
<tr>
<td>7.2.3</td>
<td>ICCID value</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Authentication Algorithms</td>
</tr>
<tr>
<td>7.2.5</td>
<td>Milenage parameters (Ci/Ri)</td>
</tr>
<tr>
<td>7.2.6</td>
<td>Enable / Disable USIM Application</td>
</tr>
<tr>
<td>7.2.7</td>
<td>APDU trace</td>
</tr>
<tr>
<td>7.3</td>
<td>sysmo-isim-tool (sysmoISIM-SJA2)</td>
</tr>
<tr>
<td>7.3.1</td>
<td>OPC value</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Ki value</td>
</tr>
<tr>
<td>7.3.3</td>
<td>ICCID value</td>
</tr>
<tr>
<td>7.3.4</td>
<td>Authentication Algorithms</td>
</tr>
<tr>
<td>7.3.5</td>
<td>Milenage parameters (Ci/Ri)</td>
</tr>
<tr>
<td>7.3.6</td>
<td>Enable / Disable USIM Application</td>
</tr>
<tr>
<td>7.3.7</td>
<td>USIM record in EF.DIR</td>
</tr>
<tr>
<td></td>
<td>Disabling the USIM application</td>
</tr>
<tr>
<td></td>
<td>APDU trace</td>
</tr>
</tbody>
</table>

8 Disabling / Enabling SQN verification

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>pySim-shell.py</td>
</tr>
</tbody>
</table>

9 USIM cards and 5G

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>Disabling DF_5GS files and services</td>
</tr>
</tbody>
</table>
10 USIM/ISIM cards and IMS

10.1 Disabling the ISIM functionality

10.1.1 Deactivating IMS files + services from ADF.USIM

10.1.2 Locking the ISIM application

11 APDU-Level Card Provisioning

11.1 IMSI

11.2 ICCID

11.2.1 sysmoUSIM-SJS1

11.2.2 sysmoISIM-SJA2

11.3 K / Ki

11.3.1 sysmoUSIM-SJS1

11.3.2 sysmoISIM-SJA2

11.4 OPc or OP

11.4.1 sysmoUSIM-SJS1

11.4.2 sysmoISIM-SJA2

11.5 Authentication Algorithms

11.5.1 sysmoUSIM-SJS1

11.5.2 sysmoISIM-SJA2

11.6 Milenage Configuration (Ci/Ri)

11.6.1 sysmoUSIM-SJS1

11.6.2 sysmoISIM-SJA2

11.7 Disabling the USIM Application

12 Java Card Features

12.1 Application List

12.2 Example Applet

12.3 Installation via 03.48 OTA

12.4 Installation via GlobalPlatform SCP02

13 OTA (Over The Air)

13.1 Transports

13.2 TAR (Toolkit Application Reference)

13.3 MSL (Minimum Security Level)

14 sysmoISIM-SJA2 changelog

14.1 sysmoISIM-SJA2 v1 (September 2020)

14.2 sysmoISIM-SJA2 v2 (September 2021)

15 Glossary

A Osmocom TCP/UDP Port Numbers

B Bibliography / References

References
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, OsmoS5GSN, 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 introduced 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 practice, 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 versions 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

![sysmoUSIM-SJS1](image)

Figure 2: sysmoUSIM-SJS1
Figure 3: sysmolSIM-SJA2

The sysmolUSIM-SJS1 and sysmolISIM-SJA2 are Java SIM card with the following specifications and features:

## 4.1 Physical Specification

- Available mechanical form factor
  - 2FF + 3FF (micro) SIM card (sysmolUSIM-SJS1)
  - 2FF + 4FF (nano) SIM card (sysmolUSIM-SJS1)
  - 2FF + 3FF + 4FF (sysmolISIM-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 sysmolUSIM-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] (sysmolISIM-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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Enabled</th>
<th>Length</th>
<th>Max Attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN1</td>
<td>No</td>
<td>4 digits</td>
<td>3</td>
</tr>
<tr>
<td>PIN2</td>
<td>Yes</td>
<td>4 digits</td>
<td>3</td>
</tr>
<tr>
<td>ADM1</td>
<td>Yes</td>
<td>8 bytes</td>
<td>3</td>
</tr>
<tr>
<td>PUK1</td>
<td>Yes</td>
<td>8 bytes</td>
<td>10</td>
</tr>
<tr>
<td>PUK2</td>
<td>Yes</td>
<td>8 bytes</td>
<td>10</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Length</th>
<th>Key set</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIc</td>
<td>16 bytes</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>KID</td>
<td>16 bytes</td>
<td>1, 2, 3</td>
</tr>
</tbody>
</table>

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):

<table>
<thead>
<tr>
<th>Identity</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSI</td>
<td>99970xxxxxxxxxx (from September 2021 onwards); 90170xxxxxxxxxx (until August 2021)</td>
</tr>
<tr>
<td>ACC</td>
<td>equal distribution</td>
</tr>
<tr>
<td>ICCID</td>
<td>8988211xxxxxxxxxx</td>
</tr>
</tbody>
</table>
### 4.3 Product Variants

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

<table>
<thead>
<tr>
<th>Variant</th>
<th>IMSI</th>
<th>K, OPc</th>
<th>ADM key</th>
<th>OTA key</th>
</tr>
</thead>
<tbody>
<tr>
<td>default 999-70 MCC-MNC (no ADM keys)</td>
<td>99970...</td>
<td>random, provided</td>
<td>not provided</td>
<td>not provided</td>
</tr>
<tr>
<td>default 999-70 MCC-MNC (with ADM keys)</td>
<td>99970...</td>
<td>random, provided</td>
<td>provided</td>
<td>provided</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>SKU</th>
<th>Form-Factor</th>
<th>Link to sysmocom webshop</th>
</tr>
</thead>
<tbody>
<tr>
<td>sysmoISIM-SJA2</td>
<td>1FF + 2FF + 3FF + 4FF</td>
<td><a href="http://shop.sysmocom.de/products/sysmoISIM-SJA2">http://shop.sysmocom.de/products/sysmoISIM-SJA2</a></td>
</tr>
</tbody>
</table>

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/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 http://shop.sysmocom.de/t/sim-card-related/card-readers

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 pcsd and libccid packages, e.g. using `apt-get install pcsd 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
+ T0 = 9F --> Direct Convention
+ T0 = 9F, Y(1): 1001, K: 15 (historical bytes)
   TA(1) = 96 --> F1=512, Di=32, 16 cycles/ETU
   250000 bits/s at 4 MHz, fMax for F1 = 5 MHz => 312500 bits/s
   TD(1) = 80 --> Y(1+1) = 1000, Protocol T = 0
---
   TD(2) = 1F --> Y(1+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 adapter at http://shop.sysmocom.de/products/sim-adapter-pcb

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 git://git.osmocom.org/libosmocore / http://git.osmocom.org/libosmocore/

6.1.1 SYNOPSIS


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.

```bash
$ osmo-auc-gen -3 -a milenage -k 1D8B2562B992549F20D0F42113EAA6FB -o 398153093661279 -r FB1FC74BE07059FEF -s 101
```

The resulting values have the following meaning:

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

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.

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 e60ab2d36480000b832f8983cbb39c6 -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 90170000011000

UMTS Authentication
RES: dc224da20351d4d1
CK: a88a03ff82a8a26e3ea43d82865a725
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 procedure 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 25 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 git://git.osmocom.org/pysim / https://git.osmocom.org/pysim/
- the sysmo-usim-tool program from https://git.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 90171000011000 (it was 90170000011000 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 ←
898821100000011000 -o 398153093661279FB1FC74BE07059FEF -k 1 ←
D8B2562B992549F20D0F42113EAA6FB
Insert card now (or CTRL-C to cancel)
Generated card parameters :
> Name : Magic
> SMSP : e1fffffffffffffffffffffffff0581005155f5fffffff000000
> ICCID : 898821100000011000
> 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 program 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, the 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 at the relevant ETSI/3GPP specs or the following source file from the libosmo core 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)

```bash
$ ./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...  
* Initializing...  
* Reading...  
* Current OPc setting:  
  OP: 0x01  
  OP/OPc: df3d7f95d27005a5441820a31a020bf6

EFOPC 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`)

```bash
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -O df3d7f95d27005a5441820a31a020bf6
```
An OPC value is programmed using the option \texttt{-C} (\texttt{--set-opc})

\begin{verbatim}
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -C df3d7f95d27005a5441820a31a020bf6
\end{verbatim}

### 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 \texttt{-k} (\texttt{--ki}) can be used.

\begin{verbatim}
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -k
\end{verbatim}

Option \texttt{-K} (\texttt{--set-ki}) allows setting the Ki to a user defined value.

\begin{verbatim}
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -K 0123456789ABCDEF0123456789ABCDEF
\end{verbatim}

### 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 \texttt{-i} (\texttt{--iccid}) extracts the current ICCID of the card. Note that the last nibble is padded with \texttt{0xf}, since it is unused.

\begin{verbatim}
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -i
\end{verbatim}

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

#### 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` (`--milencage`) can be used:

```
$ ./sysmo-usim-tool.sjs1.py --adm1 55538407 -l
```
Reading Milenage parameters...
* Initializing...
* 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
```
In the example above, the USIM application is still enabled. We can disable the USIM application using the command line switch `--classic`

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

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

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

### 7.2.7 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.

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

---

Copyright © 2016-2022 sysmocom - s.f.m.c. GmbH
Card transaction: APDU:00d600201000000000000000000000000000000002 → APDU:(no data) SW ← 9000
Card transaction: APDU:00d600301000000000000000000000000000000004 → APDU:(no data) SW ← 9000
Card transaction: APDU:00d600401000000000000000000000000000000008 → 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...
  * Initializing...
  * 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.

```bash
$ ./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...
* Initializing...
* Reading...
* Current Ki setting:
Ki: 0123456789abcdef0123456789abcdef

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

```bash
$ ./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:

```bash
$ ./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...
* Initializing...
* Reading...
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.

Let’s 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:

```bash
$ ./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:

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

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

```bash
./sysmo-isim-tool.sja2.py --adm1 55538407 -L \
00000000000000000000000000000000\ 
000000000000000000000000000000000001\ 
00000000000000000000000000000001\ 
00000000000000000000000000000001\ 
00000000000000000000000000000001\ 
00000000000000000000000000000001\ 
00000000000000000000000000000001\ 
00000000000000000000000000000001\ 
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 talk 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 application:

• 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 entry, 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 --adm 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...
  * Initializing...
  * Reading...
  * Current status of Record No. 1 in EF.DIR:
    61194f10a0000000871002fffffff890709000050055553696d31ffffffffffffffffffffff
    ==> 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 --adm 55538407 -c
```

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

```
$ ./sysmo-isim-tool.sja2.py --adm 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
java -jar ./gp.jar --key-enc KIC1 --key-mac KID1 --key-dek KIK1 --lock-applet ←
A000000871002FFFFFFFF8907090000
```

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
java -jar ./gp.jar --key-enc KIC1 --key-mac KID1 --key-dek KIK1 --unlock-applet ←
A000000871002FFFFFFFF8907090000
```

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...
* Initialing...
* Card transaction: APDU:00a40000023f00 ==> APDU:(no data) SW:9000
* New Milenage Parameters for (EF.MLNGC):
  C1: 00000000000000000000000000000000
  C2: 00000000000000000000000000000000
  C3: 00000000000000000000000000000000
```

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

```bash
$ ./pySim-shell.py -p 0
Using PC/SC reader interface
Waiting for card...
Autodetected card type: sysmoISIM-SJA2
Info: Card is of type: UICC-SIM
AIDs on card:
USIM: a00000000871002000000000000000000 (EF.DIR)
ISIM: a00000000871004000000000000000000 (EF.DIR)
ARA-M: a0000000015141434c00
ADF.ISD: a000000003000000
Detected CardModel: SysmocomSJA2
Welcome to pySim-shell!
pySIM-shell (MF)> verify_adm 91820128
select ADF.USIM/EF.USIM_SQN
```
this assumes your card is inserted into PC/SC reader number 0 (first reader)
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):

```json
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,
    "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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
  ]
}
```

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:

```bash
pySIM-shell (MF/ADF.USIM/EF.USIM_SQN)> update_binary_decoded --json-path $.flag1.sqn_check 0
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,
    "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, 0, 0, 0, 0, 0, 0, 0
  ]
}
```
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.

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 15.

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), deactivate 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 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 17, 2022) onwards, there is a script included at the `scripts/deactivate-5g.script` 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

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

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](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.script` 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 KIC1 --key-mac KID1 --key-dek KIK1 --lock-applet ←
A0000000871004FFFFFFFF8907090000
```

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.

Copyright © 2016-2022 sysmocom - s.f.m.c. GmbH
Likewise, you can unlock the ISIM application using:

```
java -jar ./gp.jar --key-enc KIC1 --key-mac KID1 --key-dek KIK1 --unlock-applet
A0000000871004FFFFFFFF8907090000
```

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.

Copyright © 2016-2022 sysmocom - s.f.m.c. GmbH
11.3.2 sysmoISIM-SJA2

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

<table>
<thead>
<tr>
<th>Application</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM</td>
<td>A515/6F20</td>
</tr>
<tr>
<td>USIM</td>
<td>ADF_USIM/AF20</td>
</tr>
<tr>
<td>ISIM</td>
<td>ADF_ISIM/AF20</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0x00 for OP, 0x01 for OPc (see Section 3.4.1)</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>OP or OPc value, depending on byte at offset 0</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Application</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM</td>
<td>A515/6F20</td>
</tr>
<tr>
<td>USIM</td>
<td>ADF_USIM/AF20</td>
</tr>
<tr>
<td>ISIM</td>
<td>ADF_ISIM/AF20</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0x10 for OP, 0x00 for OPc (see Section 3.4.1)</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>K/Ki value</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>OP or OPc value, depending on byte at offset 0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Algorithm</th>
<th>Supported Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Milenage</td>
<td>2G + 3G</td>
</tr>
<tr>
<td>03</td>
<td>COMP128v1</td>
<td>2G</td>
</tr>
<tr>
<td>04</td>
<td>XOR 2G</td>
<td>2G</td>
</tr>
<tr>
<td>06</td>
<td>COMP128v2</td>
<td>2G</td>
</tr>
<tr>
<td>07</td>
<td>COMP128v3</td>
<td>2G</td>
</tr>
<tr>
<td>08</td>
<td>XOR 3G</td>
<td>3G</td>
</tr>
</tbody>
</table>

**Note**
You need to be authenticated using ADM1 PIN.

### 11.5.2 sysmoSIM-SJA2

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

<table>
<thead>
<tr>
<th>Application</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM</td>
<td>A515/6F20</td>
</tr>
<tr>
<td>USIM</td>
<td>ADF_USIM/AF20</td>
</tr>
<tr>
<td>ISIM</td>
<td>ADF_ISIM/AF20</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Algorithm</th>
<th>Supported Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COMP128v1</td>
<td>SIM</td>
</tr>
<tr>
<td>2</td>
<td>COMP128v2</td>
<td>SIM</td>
</tr>
<tr>
<td>3</td>
<td>COMP128v3</td>
<td>SIM</td>
</tr>
<tr>
<td>4</td>
<td>MILENAGE</td>
<td>SIM+USIM+ISIM</td>
</tr>
<tr>
<td>5</td>
<td>SHA1AKA</td>
<td>SIM+USIM+ISIM</td>
</tr>
</tbody>
</table>

### 11.6 Milenage Configuration (Ci/Ri)

#### 11.6.1 sysmoUSIM-SJS1

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

The structure of the file is as follows:

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>C1</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>C2</td>
</tr>
<tr>
<td>32</td>
<td>16</td>
<td>C3</td>
</tr>
<tr>
<td>48</td>
<td>16</td>
<td>C4</td>
</tr>
<tr>
<td>64</td>
<td>16</td>
<td>C5</td>
</tr>
<tr>
<td>80</td>
<td>1</td>
<td>R1</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>00000000000000000000000000000000</td>
</tr>
<tr>
<td>C2</td>
<td>00000000000000000000000000000001</td>
</tr>
<tr>
<td>C3</td>
<td>00000000000000000000000000000002</td>
</tr>
<tr>
<td>C4</td>
<td>00000000000000000000000000000004</td>
</tr>
<tr>
<td>C5</td>
<td>00000000000000000000000000000008</td>
</tr>
<tr>
<td>R1</td>
<td>0x40</td>
</tr>
<tr>
<td>R2</td>
<td>0x00</td>
</tr>
<tr>
<td>R3</td>
<td>0x20</td>
</tr>
<tr>
<td>R4</td>
<td>0x40</td>
</tr>
<tr>
<td>R5</td>
<td>0x60</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Application</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM</td>
<td>FIXME</td>
</tr>
<tr>
<td>USIM</td>
<td>ADF_USIM/AF21</td>
</tr>
<tr>
<td>ISIM</td>
<td>ADF_ISIM/AF21</td>
</tr>
</tbody>
</table>

The structure of the files is as follows:

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>R1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>R2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>R3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>R4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>R5</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td>C1</td>
</tr>
<tr>
<td>21</td>
<td>16</td>
<td>C2</td>
</tr>
<tr>
<td>37</td>
<td>16</td>
<td>C3</td>
</tr>
<tr>
<td>53</td>
<td>16</td>
<td>C4</td>
</tr>
<tr>
<td>69</td>
<td>16</td>
<td>C5</td>
</tr>
</tbody>
</table>

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 pointing to 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

<table>
<thead>
<tr>
<th>Application</th>
<th>PID</th>
<th>AID</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM Application</td>
<td>A0000000090001FFFFFFFF8900</td>
<td>A0000000090001FFFFFFFF8900000000</td>
</tr>
<tr>
<td>SIM RFM Application</td>
<td>A0000000090001FFFFFFFF8900</td>
<td>A0000000090001FFFFFFFF89B00010</td>
</tr>
<tr>
<td>RAM Application</td>
<td>A0000000090001FFFFFFFF8900</td>
<td>A0000000090001FFFFFFFF89B00010</td>
</tr>
<tr>
<td>USIM RFM App (opt)</td>
<td>FF434E5258104004203</td>
<td>FF434E5258104004203000000000000</td>
</tr>
<tr>
<td>USIM Application</td>
<td>A00000000871002FF49FFFF8900</td>
<td>A00000000871002FF49FFFF89040B0000</td>
</tr>
<tr>
<td>CAF_TP App (opt)</td>
<td>A0000000090001FFFFFFFF8900</td>
<td>A0000000090001FFFFFFFF89B00010</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Application</th>
<th>AID</th>
</tr>
</thead>
<tbody>
<tr>
<td>USIM</td>
<td>A00000000871002FFFFFFFF8907090000</td>
</tr>
<tr>
<td>ISIM</td>
<td>A00000000871004FFFFFFFF8907090000</td>
</tr>
<tr>
<td>HPSIM?</td>
<td>A0000000087ABCDFFFF88907090000</td>
</tr>
<tr>
<td>Remote Application Management</td>
<td>53696D62614E2E52414D</td>
</tr>
<tr>
<td>SIM?</td>
<td>A0000000090001FFFFFFFF8900000000</td>
</tr>
<tr>
<td>Remote File Management</td>
<td>53696D62614E2E52464D</td>
</tr>
<tr>
<td>ARA-M</td>
<td>A00000015141434C00</td>
</tr>
</tbody>
</table>
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 git://git.osmocom.org/sim/hello-stk / https://git.osmocom.org/sim/hello-stk/

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

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 git://git.osmocom.org/sim/sim-tools / http://git.osmocom.org/sim/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 webshop@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

<table>
<thead>
<tr>
<th>sysmoUSIM key</th>
<th>GlobalPlatformPro argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>KIC1</td>
<td>--key-enc</td>
</tr>
<tr>
<td>KID1</td>
<td>--key-mac</td>
</tr>
<tr>
<td>KIK1</td>
<td>--key-dek</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Application</th>
<th>TAR</th>
<th>MSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM (Remote Application Management)</td>
<td>00000</td>
<td>0x06</td>
</tr>
<tr>
<td>SIM RFM (Remote File Management)</td>
<td>B00010</td>
<td>0x06</td>
</tr>
<tr>
<td>USIM RFM (Remote File Management)</td>
<td>B00011</td>
<td>0x06</td>
</tr>
<tr>
<td>ISIM RFM (Remote File Management)</td>
<td>B00020</td>
<td>0x06</td>
</tr>
<tr>
<td>HPSIM RFM (Remote File Management)</td>
<td>B00021</td>
<td>0x06</td>
</tr>
</tbody>
</table>

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 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.WEHSIDL, EF.WEHLPLMN, EF.WPPLMN, EF.WLRLMN, EF.WLRLNDAI
- New DF.ProSe files EF.PROSE_MON, EF.PROSE_ANP, EF.PROSE_FUNC, EF.PROSE_RADIO_COM, EF.PROSE_RADIO_MON, EF.PROSE_RADIO_ANP, EF.PROSE_POLICY, EF.PROSE.PLMN, EF.PROSE_GC, EF.PST, EF.PROSE_UIRC, EF.PROSE_GM_DISCOVERY, 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 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
Circuit-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>
<thead>
<tr>
<th>L4 Protocol</th>
<th>Port Number</th>
<th>Purpose</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>1984</td>
<td>Osmux</td>
<td>osmo-mgw, osmo-bts</td>
</tr>
<tr>
<td>UDP</td>
<td>2427</td>
<td>MGCP GW</td>
<td>osmo-bsc_mgcp, osmo-mgw</td>
</tr>
<tr>
<td>TCP</td>
<td>2775</td>
<td>SMPP (SMS interface for external programs)</td>
<td>osmo-nitb</td>
</tr>
<tr>
<td>TCP</td>
<td>3002</td>
<td>A-bis/IP OML</td>
<td>osmo-bts, osmo-bsc, osmo-nitb</td>
</tr>
<tr>
<td>TCP</td>
<td>3003</td>
<td>A-bis/IP RSL</td>
<td>osmo-bts, osmo-bsc, osmo-nitb</td>
</tr>
<tr>
<td>TCP</td>
<td>4227</td>
<td>telnet (VTY)</td>
<td>osmo-pcap-client</td>
</tr>
<tr>
<td>TCP</td>
<td>4228</td>
<td>telnet (VTY)</td>
<td>osmo-pcap-server</td>
</tr>
<tr>
<td>TCP</td>
<td>4236</td>
<td>Control Interface</td>
<td>osmo-trx</td>
</tr>
<tr>
<td>TCP</td>
<td>4237</td>
<td>telnet (VTY)</td>
<td>osmo-trx</td>
</tr>
<tr>
<td>TCP</td>
<td>4238</td>
<td>Control Interface</td>
<td>osmo-bts</td>
</tr>
<tr>
<td>TCP</td>
<td>4239</td>
<td>telnet (VTY)</td>
<td>osmo-stp</td>
</tr>
<tr>
<td>TCP</td>
<td>4240</td>
<td>telnet (VTY)</td>
<td>osmo-pcu</td>
</tr>
<tr>
<td>TCP</td>
<td>4241</td>
<td>telnet (VTY)</td>
<td>osmo-bts</td>
</tr>
<tr>
<td>TCP</td>
<td>4242</td>
<td>telnet (VTY)</td>
<td>osmo-nitb, osmo-bsc, cellmgr-ng</td>
</tr>
<tr>
<td>TCP</td>
<td>4243</td>
<td>telnet (VTY)</td>
<td>osmo-bsc_mgcp, osmo-mgw</td>
</tr>
<tr>
<td>TCP</td>
<td>4244</td>
<td>telnet (VTY)</td>
<td>osmo-bsc_nat</td>
</tr>
<tr>
<td>TCP</td>
<td>4245</td>
<td>telnet (VTY)</td>
<td>osmo-sgsn</td>
</tr>
<tr>
<td>TCP</td>
<td>4246</td>
<td>telnet (VTY)</td>
<td>osmo-gbproxy</td>
</tr>
<tr>
<td>TCP</td>
<td>4247</td>
<td>telnet (VTY)</td>
<td>OsmocomBB</td>
</tr>
<tr>
<td>TCP</td>
<td>4249</td>
<td>Control Interface</td>
<td>osmo-nitb, osmo-bsc</td>
</tr>
<tr>
<td>TCP</td>
<td>4250</td>
<td>Control Interface</td>
<td>osmo-bsc_nat</td>
</tr>
<tr>
<td>TCP</td>
<td>4251</td>
<td>Control Interface</td>
<td>osmo-sgsn</td>
</tr>
<tr>
<td>TCP</td>
<td>4252</td>
<td>telnet (VTY)</td>
<td>sysmobts-mgr</td>
</tr>
<tr>
<td>TCP</td>
<td>4253</td>
<td>telnet (VTY)</td>
<td>osmo-gtphub</td>
</tr>
<tr>
<td>TCP</td>
<td>4254</td>
<td>telnet (VTY)</td>
<td>osmo-mse</td>
</tr>
</tbody>
</table>
Table 5: (continued)

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

B Bibliography / References

References


[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


[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


[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


[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


[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


[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


[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; UIxCC-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


[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


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