SA Low Level Driver

Release Notes

Applies to Product Release: 01.00.06.00
Publication Date: May 07, 2014

Contributors to this document

Texas Instruments, Incorporated
20450 Century Boulevard
Germantown, MD 20874 USA

Contents

Release Notes

SA Low Level Driver version 01.00.06.00

Overview

This document provides the release information for the latest Security Accelerator Low Level Driver (SA LLD) which should be used by drivers and application that interface with SA. Although SASS supports 3GPP specific Ciphering and Authentication algorithms such as Kasumi F8/F9 and Snow3G F8, those algorithms are locked out in this standard SA LLD distribution. In order to access 3GPP specific functionalities, one must obtain the SASS 3GPP Enabler as well from TI.

SA LLD module includes:

- Compiled library (Big and Little) Endian of SA Low Level Driver.
- Sources[1]
- Example and unit test code.
- API reference guide
- Software Manifest Documentation

This release notes is for SA LLD version 1.0.6.0(1_0_6_0)
For the rest of the document the keyword *SA_Version* will indicate the SA LLD version of this release.
The SA LLD is usually released with a PDK package; the keyword *PDK_Version* indicates the corresponding PDK version.

LLD Dependencies

- This release of SA LLD requires CSL package released with PDK.

TI MAS Components

This release is built and provided with the following TI MAS components:

- ti.mas.types 5.5.4.0 (Common data type definitions)
- ti.mas.swtools 4.6.0.15 (Internally used s/w tools)

[1] Available in source release only

- ti.mas.secutil 1.1.4.0 (Common security utilities)
- ti.mas.pktutl 2.1.1.0 (Common packet utilities)
- ti.mas.aes 1.2.5.1 (AES cipher algorithm utility)
- ti.mas.sha1 1.1.4.0 (SHA1 hash algorithm utility)

Label and Version Information

Table 1 lists the software label and versions supported by this release.

Table 1 Label and versions supported by this release

Label/Version Information
SALLD.01.00.06.00

Resolved Incident Reports (IR)

Table 2 provides information on IR resolutions incorporated into this release.

Table 2 Resolved IRs for this Release

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00107358	Major	[SA LLD] Need to build and test the SA LLD to enable the call stack trace
SDOCM00107087	Major	SA LLD function Sa_getPDSPVersion does not return correct version value
SDOCM00107684	Major	Memory overwritten in function Sa_create()

Known Issues/Limitations

Table 3 Known Issue IRs for this Release

IR Parent/ Child Number	Severity Level	IR Description
00042288 00042377 00042378	Major	CCS C6678 simulator does not support CPPI multi-buffer operation at PA/SA module

Component Compatibility

The example and unit test have been verified with certain version of PDK package and may need to be modified as per API changes introduced by other components if used with a different version of PDK. Following are the version dependencies.

TCI6614 SoC: PDK package released with SC-MCSDK version 02.02.01.03
C6670/C6678 SoC: PDK package released with BIOS-MCSDK version 02.01.01.04
PA LLD version 01.03.00.07

New/Updated Features and Quality

Release 1.0.6.0

- Update TI DSP Codegen tools to 7.3.16
- Resolved IRs as listed at section "Resolved Incident Reports (IR)".

Release 1.0.5.4

- This release includes the following feature enhancements
    - 3GPP Count-C report option: SASS will copy 32-bit Count-C into the timestamp field within the CPPI descriptor when the Air-Ciphering configuration flag sa_AC_CONFIG_COPY_COUNTC is set.
    - 3GPP Air Ciphering with user-supplied IV: SASS will use user-supplied IV instead of the one generated from internal configuration parameters to perform air-ciphering or authentication operation when IV and count-C are supplied in the SA psInfo command. Use macro sa_PSINFO_SET_COUNTC() and sa_PSINFO_SET_IV() to set the user-supplied Count-C and IV respectively.
    - IPSEC ESP NAT-T support: Enhance Sa_chanSendData() and Sa_chanReceiveData() functions to insert and remove ESP NAT-T header respectively. To enable NAT-T insertion, call Sa_chanSendData() API with NAT-T related parameters stored at data structure Sa_ipsecNatTInfo_t within Sa_PktInfo_t and the valid bit sa_PKT_INFO_VALID_IPSEC_NAT_T_INFO set. The NAT-T header will be removed by API Sa_chanReceiveData() autonomously.
- Resolved IRs as listed below

IR Parent/ Child Number	Severity Level	IR Description
00097130	Major	Enhance SA_LLD to support CPU-less handling of IPSec NAT traversal in egress direction
00097504	Major	SA LLD: Insert Count-C into descriptor in the from-air operation
00097505	Major	SA LLD: Add option to use user-specified IV in stead of firmware-generated IV for MAC-I computation
00097735	Major	SA LLD: SRTP null-encryption mode does not work

Release 1.0.5.3

- Enhanced SA LLD system statistics (Sa_SysStats_t) to include protocol-specific system statistics in addition to the system error statistics. To access the system error statistics, the

application should be updated to include "err." in front of each system error parameters. For example:

p_stats->errNoMem ➔ p_stats->err.errNoMem

- Resolved IRs as listed below:

IR Parent/ Child Number	Severity Level	IR Description
00095522	Minor	SA LLD: System statistics enhancements
00096436	Major	SA LLD: Authentication failed in IPSEC ESP CCM mode
00096439	Major	SA LLD: Authentication failed at Data Mode CCM operation
0097108	Minor	SA LLD: SRTP should support 48-bit sequence number for From-To re-key operation

Release 1.0.5.2

- Added the following parameters in Sa_IpsecStats_t
  - replayCxt: Snapshot of the replay window context
  - txSN: Sequence Number (32-bit) of the last transmitted packet
- Enhanced data mode operation by providing the command label updating data structure to allow application to update command label with Macros in stead of invoking Sa_chanSendData API.
  - Add command label updating data structure Sa_CmdLbUpdateInfo_t
  - Add command label updating Macro sa_mDmResetCmdLb() and sa_mDmUpdateCmdLb()
- Resolved IRs as listed below.

IR Parent/ Child Number	Severity Level	IR Description
00092404	Minor	Support for retrieving and updating command label for SA channel in Data mode
00094857	Major	Additional IPSec Stats for transitioning IPSec crypto from NetCP SA to Software
00095796	Major	SA LLD: SRTCP and SRTP decryption is broken
00095798	Major	SA LLD: Add IPSEC AH GMAC support in data mode

Release 1.0.5.1

- IPSEC byte counters are provided for encryption and decryption, via new parameters in Sa_IpsecStats_t
  - txByteCountHi
  - txByteCountLo
  - rxByteCountHi

    - rxByteCountLo
- Transmit sequence number rollover check is implemented such that during tramsit operation, packets with overflowing ESN will be dropped.
- Resolved IRs as listed below

IR Parent/ Child Number	Severity Level	IR Description
00094371	Major	Provide IPSEC TX sequnece number rollover check
00094373	Major	Provide IPSEC byte counters

.

Release 1.0.5.0

- SRTCP is provided by SA LLD as a software-only operation. In this release, we have added SRTCP related test cases into SA LLD unit test suite and fully verified its operation. The following two parameters in the SRTCP statistics are renamed to be consistent with SRTP statistics:
    - txNwrap ➔ txRekey
    - rxNwrap ➔ rxRekey
- Resolved IRs as listed below:

IR Parent/ Child Number	Severity Level	IR Description
00094275	Major	SA LLD: SRTCP operation is broken
00095004	Minor	SA LLD:Cleanup API actions when invalid instance is detected
00095005	Minor	SA LLD:Null-pointer check required prior to security context buffer access since SC may not be allocated in unidirectional SA

-

Release 1.0.4.1

- SA LLD is enhanced to support DES-CBC mode operation. Please note that DES-CBC has been proved to be a non-secure algorithm. It is only supported for backward compatibility with old security devices.
    - Add cipher mode sa_CipherMode_3DES_CBC for 3DES-CBC operation. The cipher mode sa_CipherMode_DES_CBC indicates DES-CBC mode now.
- Define new macro sa_SWINFO_UPDATE_DEST_INFO(info, queueID, flowIndex) to update destination information within swInfo[2] after Sa_chanSendData() API is invoked in Data mode.
- Resolved IRs as listed below:

IR Parent/ Child Number	Severity Level	IR Description

IR Parent/ Child Number	Severity Level	IR Description
00092795	Minor	Data Mode Sa_chanSendData() enhancement to configure receive queue and CPPI flow
00093284	Major	SA LLD: Add DES-CBC mode support for IPSEC ESP operation

Release 1.0.4.0

- SA LLD is enhanced to support mixed-Endian mode operation such that the security channel is created and configured by the master processor, (e.g. ARM which is outside of SoC), while the data processing is handled by slave processor(s) (e.g. DSPs). When in this mode, SA LLD must create and maintain a shadow instance, which is a copy of the master instance with opposite Endianness. New APIs are created to query the shadow instances, which should be passed to and used by the slave processor(s).
    - Add new parameter ctrlBitMap to the size configuration data structure Sa_SizeCfg_t and Sa_ChanConfig_t. The ctrlBitMap should be set to sa_SIZE_CONFIG_CREATE_SHADOW_INST to enable system and channel shadow instance for mixed-Endian mode operation.
    - Add Sa_SizeCfg_t* to Sa_Config_t to control shadow instance creation at API Sa_create().
    - Add new APIs Sa_getShadowHandle() and Sa_chanGetShadowHandle() to query the shadow instances.

  Note: Only the following use case is supported at this moment, further enhancements may be provided:
    - IPSEC ESP/AH mode
    - Control Path at ARM and Data Path at DSPs

- Added new API Sa_getPDSPVersion() to query the version number of PDSP image to provide the capability to verify the compatibility of the SA LLD and the PDSP image which may be downloaded by another processor. The version number of the PDSP image should be identical to the version number of the SA LLD.
- One additional OSAL API's has been introduced to enhance SA LLD portability. User's upgrading from the previous version of SA LLD will need to provide these additional OSAL API's. Detail about the API is documented in the file "sa_osal.h".
    - int Osal_saGetSysEndianMode(void);
- Resolved IRs as listed below

IR Parent/ Child Number	Severity Level	IR Description
00092796	Minor	SA Incorrect statistics for IP Sec Unidirectional channel
00092403	Major	SA LLD: Support Mixed-Endian mode operation across ARM and DSP processors in IPSEC mode

IR Parent/ Child Number	Severity Level	IR Description
00092180	Major	SA LLD: Provide combined (MAS and SASS) libraries
00091770	Minor	SA LLD: Add API to query the version numbers of SASS PDSP images
00091166	Minor	SA LLD: Add PKA (Public Key Accelerator) test case
00092963	Major	SA LLD: IPSEC AH AES-XCBC mode is broken
00093063	Major	SA LLD: IPSEC ESP AES-XCBC operation with Null-encryption cause the SASS to crash
00093195	Major	SA LLD:AES-CBC-192(256) Decryption fail when SA channel is setup by ARM

Release 1.0.3.2

- Cleanup and enhance the SA Air-Ciphering Mode operations including the following features:
  - Simultaneous Null-Ciphering and Null-Authentication operation
  - Preserve application private information on PS_INFO through air cipher channel
- Resolved IRs as listed below

IR Parent/ Child Number	Severity Level	IR Description
00091701	Minor	SA LLD: Enhanced Cache-related OSAL functions per Cache Advisory 12
00091388	Major	SA LLD: 3GPP Air Ciphering Mode enhancements including null-Ciphering support

Release 1.0.3.1

- Cleanup and enhance the SA Data Mode operations including the following features:
  - 3GPP Air Ciphering Mode such as Kasumi-F8 and Kasumi-F9
  - Simultaneous Null Encryption and Null Authentication modes
- Resolved IRs as listed below

IR Parent/ Child Number	Severity Level	IR Description
00091013	Major	SA LLD: RTSC component is not usable if built at Linux due to an Eclipse configuration problem

IR Parent/ Child Number	Severity Level	IR Description
00091014	Minor	SA LLD: Test 6 of the SA Basic Example is broken
00091165	Major	SA LLD: Some of Data Mode operations are broken

Release 1.0.3.0

- Implement a new API Sa_chanGetSwInfo() which returns the SA-specific software information required for all packets to be delivered to SASS. It is an optional utility function to query the SA-specific software information in both directions in case Sa_chanSendData() is not used and/or the callout function ChanRegister() is not implemented.
- Added support to build SA LLD for ARM cortex-A8 Linux user space. SA LLD example has been ported to Linux user space to demonstrate the usage of SA LLD from Linux user space. Linux user space support is available only for devices with ARM processor.
- 2 additional OSAL API's have been introduced to enhance SA LLD portability. User's upgrading from the previous version of SA LLD will need to provide these additional OSAL API's. Detail about the API is documented in the file "sa_osal.h".
    - uint16_t Osal_saGetProcId (void);
    - void* Osal_saGetSCPhyAddr(void* vaddr);
- Resolved IRs as listed below:

IR Parent/ Child Number	Severity Level	IR Description
00089138	Major	Add SALLD API to retrieve the SA-specific software information for both Tx and Rx Channel
00089349	Major	API Sa_resetControl() fails to reset PDSP when called at the second time
00089473	Major	Enhance SA LLD to support Linux User Mode operation
00089634	Major	Can not handle GTPU packet from PA if PA command set is executed

NOTE: The example and unit test have been verified with certain version of PDK package and may need to be modified as per API changes introduced by other components if used with a different version of PDK. Following are the version dependencies.

- **TCI6614 SoC:** PDK package released with SC-MCSDK version 02.00.00.05
- **C6670/C6678 SoC:** PA LLD version 01.02.01.00

Release 1.0.2.1

- Enhance the SA system and channel size configuration structures to provide buffer size and alignment requirements based on system cache configuration.
    - Sa_SizeCfg_t: add cacheLineSize (For example: L1: 64; L2:128;Non-cacheable: 0)
    - Sa_ChanSizeCfg_t: add cacheLineSize (For example: L1:64; L2:128:Non-shared: 0)
- Enhance the SA LLD to support multi-core operation by adding the following two APIs
    - Sa_start(): To activates the SA LLD instance. This function should be called once at all secondary cores which share the same SALLD instance.
    - Sa_chanStart(): To activates the local channel instance at secondary cores. This API should be called once at any secondary core which invokes the same SALLD channel that is created and configured at the master core.
- SA initialization configuration structure (Sa_Config_t) is enhanced to include the following parameters:
    - baseAddr: specify the SASS base address which is defined at "ti/csl/cslr_device.h"
- The size of ID field at the following data structures and APIs is increased to 32-bit.
    - Sa_Config_t
    - Sa_chanConfig_t
    - Sa_getID()
    - Sa_chanGetID()
- Enhance the following bit definition to the IPSEC Configuration Control bitmap to specify whether to set the security context to be permanent.
    - sa_IPSEC_CONFIG_PERMANENT_SC
- Remove unused OSAL functions Osal_saMalloc() and Osal_saFree()
- Add new OSAL functions Osal_saBeginScAccess() and Osal_saEndScAccess() since the security context cache protection requirement may be different from the system and channel instances.
- Resolved IRs as listed below"

IR Parent/ Child Number	Severity Level	IR Description
00086959	Major	SRTP Authentication fails with multiple channels execution with Sequence number > 32768
00086557	Minor	SALLD: Remove XDIAS dependency completely
00086186	Major	SA LLD:TX/RX Operation Multi-segment Support

IR Parent/ Child Number	Severity Level	IR Description
00085646	Major	SA LLD: Authentication failure while interoperating with VoIP in HMAC-SHA1
00085205	Major	Enhance the IV generation for IPSEC Tx packets in AES-CBC and 3DES-CBC modes
00084471	Major	SA LLD: Add IPSEC ESP/AH AES-XCBC-MAC-96 support
00084408	Major	SALLD: Problem running asymmetric applications on multiple cores
00083157	Minor	SA LLD: SA Base address should be passed in at Sa_create()
00083156	Minor	SA LLD: typo at API Sa_getRandonNum()

Release 1.0.2.0:

- Initial engineering release to support earlier integration of TI internal projects.
- Enhance 3GPP Air-Ciphering code to support LTE PDU of both data plane and control plane.

Release 1.0.1.4:

- Cleanup OSAL functions at examples and unit tests
- Enhance SALLD Destination Info structure to include the following two parameters:
    - swInfo0: User-defined channel-specific parameter which will be placed in SwInfo0 for packets from SA
    - swInfo1: User-defined channel-specific parameter which will be placed in SwInfo1 for packets from SA
- Enhance SALLD Air Ciphering Statistics Structure to include Count-C. Refer to the API file salld.h for new structure definitions.
- Resolved IRs

IR Parent/ Child Number	Severity Level	IR Description
00081320	Major	SA 3GPP Air Ciphering enhancements: Add CMAC and Kasumi-F9 support

IR Parent/ Child Number	Severity Level	IR Description
00081318	Minor	SA Enhancements: pass user-defined swInfo[2] into the CPPI descriptor of output packets
00068893	Major	Clean up SA LLD Release notes ECCN information

Release 1.0.1.3

- Support C66 ELF only
- Update SA examples and Unit Tests per Keystone C6608 PDK 1.0.0.5 or C6616 PDK 1.0.0.12

Release 1.0.1.2

- Added 3GPP Air Ciphering key stream generation support for GSM A5/3, ECSD A5/3 and GEA3
- Update SA examples and Unit Tests per Keystone C6608 PDK 1.0.0.4 or C6616 PDK 1.0.0.11
- Add the following OSAL functions
    - Osal_saBeginMemAccess
    - Osal_saEndMemAccess

Release 1.0.1.1

- Added SA LLD version related APIs
- Update SA examples and Unit Tests per Keystone C6608 PDK 1.0.0.2
- Resolved IRs

IR Parent/ Child Number	Severity Level	IR Description
77247	Major	SA PDSP Firmware lockup while processing 1514-byte IPSEC ESP packet
77248	Major	SA Firmware may drop packets air-ciphering mode
77316	Major	IPSEC ESP Unit test failed in AES-CBC encryption mode at little-endian VDB

Release 1.0.1.0

- Modified types from XDC to C99
- Support Both ELF and COFF
- Changed all source, header, and example code to reflect CSL include path change in Keystone C6616 CSL version 1.0.0.17
- Changed XDC tool version to 3.20.00.41 in example and test projects

- Modified the SA LLD APIs to remove the TI XDIAS dependency
    - Sa_numAlloc() and Sa_chanNumAlloc() are removed. The constant sa_N_BUFS and sa_CHAN_N_BUFS should be used to define the number of memory buffers required by SA LLD instead.
    - Sa_alloc (const IALG_Params*, struct IALG_Fxns **, IALG_MemRec *) is replaced with the new API Sa_getBufferReq(Sa_SizeCfg_t *, int size[], int align[]).
        - The ILAG_Params type of configuration information is replaced by Sa_SizeCfg_t.
        - The optional struct IALG_Fxns pointer is removed
        - The IALG_MemRec is replaced with the memory size and alignment requirement arrays.
    - Sa_chanAlloc (const IALG_Params*, struct IALG_Fxns **, IALG_MemRec *) is replaced with the new API Sa_chanGetBufferReq(Sa_ChanSizeCfg_t *, int size[], int align[]).
        - The ILAG_Params type of configuration information is replaced by Sa_ChanSizeCfg_t.
        - The optional struct IALG_Fxns pointer is removed
        - The IALG_MemRec is replaced with the memory size and alignment requirement arrays.
    - Sa_init (IALG_Handle , const IALG_MemRec *, IALG_Handle , const IALG_Params *) is replaced with the new API Sa_create (Sa_Config_t *cfg, void* bases[], Sa_Handle *pHandle).
        - The ILAG_Params type of configuration information is replaced by Sa_Config_t.
        - This API should be called with the allocated memory buffer base addresses in stead of the IALG_MemRec.
        - This API will return the SA LLD handle which identifies the SA LLD instance and should be used for all other SA LLD Common API calls.
    - Sa_chanInit (IALG_Handle , const IALG_MemRec *, IALG_Handle , const IALG_Params *) is replaced with the new API Sa_chanCreate (Sa_Handle handle, Sa_ChanConfig_t *cfg, void* bases[], Sa_ChanHandle *pChanHdl).
        - The ILAG_Params type of configuration information is replaced by Sa_ChanConfig_t.
        - This API should be called with the allocated memory buffer base addresses in stead of the IALG_MemRec.
        - This API will return the SA LLD channel handle which identifies the SA LLD channel instance and should be used for all other SA LLD Channel API calls.
    - Sa_free (IALG_Handle , IALG_MemRec *) is replaced with the new API Sa_close (Sa_Handle handle, void* bases[])
        - This function returns the allocated memory buffer base addresses in array bases[] so that the application can free the buffers.

    - Sa_chanFree (IALG_Handle , IALG_MemRec *) is replaced with the new API Sa_chanClose (Sa_ChanHandle handle, void* bases[])

- This function returns the allocated memory buffer base addresses in array bases[] so that the application can free the buffers.

Release 1.0.0.1:

- Modifications to the PDSP firmware to support the latest Keystone simulator (0.8.0.1)
- Modifications to the examples and unit tests to support the new CPPI specification (4.2.9)

Release 1.0.0.0:

- Initial internal Release

Licensing

Please refer to the software Manifest document for the details.

Delivery Package

The delivery package from Texas Instruments will be delivered as follows:

```
setupwin32_salld_<SA_Version>.exe
setuplinux_salld_<SA_Version>.bin
```

Installation Instructions

Installation prerequisite

- Install CCSv5 (or CCSv4)
- Install the functional simulator for the device under test if it is not included in CCSv5.
- Install the dependent components as listed in section "LLD Dependencies". The LLD's are expected to be installed as part of PDK for the device.

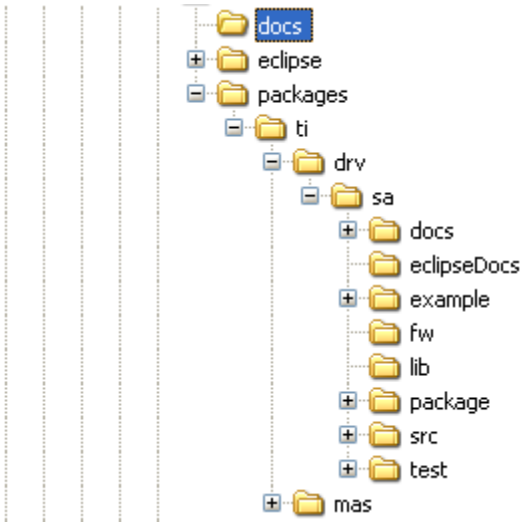
Installation guidelines

The steps to be followed for installation of the SA LLD release are as follows:

1. Download the release executable
2. Run the executable file; follow the instructions and install the SA LLD software.
3. Install the dependent components as listed in section "LLD Dependencies". The LLD's are expected to be installed as part of PDK for the device.

Directory structure

If SA LLD is installed in default location following would be directory structure:



The following table explains each individual directory covered as part of SA LLD installation:

Directory Name	Description
`ti/drv/sa`	The top level directory contains the following:- 1. *Exported Driver header file* Header files which are provided by the SA low level driver and should be used by the application developers for driver customization and usage.
`ti/drv/sa/build`	The directory contains internal XDC build related files which are used to create the SA low level driver package.
`ti/drv/sa/docs`	The directory contains the SA low level driver documentation.
`Ti/drv/sa/eclipseDocs`	The directory contains the Eclipse help configuration files
`ti/drv/sa/example`	The "`example`" directory in the SA low level driver contains the example projects
`ti/drv/sa/test`	The "test" directory in the SA low level driver contains unit test code and projects
`ti/drv/sa/lib`	The "`lib`" folder has pre-built Big and Little Endian libraries for the SA low level driver along with their *code/data size information*.
`ti/drv/sa/fw`	C data files required to configure the SA hardware sub-system.
`ti/drv/sa/package`	Internal SA low level driver package files.
`ti/mas`	The directory contains all the dependent TI-MAS modules which are required by the SA LLD
`eclipse`	The directory contains the Eclipse plug-in help files
`docs`	The directory contains top-layer documents such as SA LLD user guide
`ti/drv/sa/src`	The "src" directory contains the SA LLD source code. This directory is only available in a source release package.

Customer Documentation List

Table 4 lists the documents that are accessible through the **/docs** folder on the product installation CD or in the delivery package.

Table 4 Product Documentation included with this Release

Document #	Document Title	File Name
1	API documentation (generated by Doxygen)	sa\docs\doxygen\html\index.html
2	Release Notes (this document)	sa\docs\ReleaseNotes_SA_LLD.pdf
3	Software Manifest document	sa\docs\SA_LLD_1_0_SoftwareManifest.pdf
4	User Guide	docs\UserGuide_SA_LLD.pdf