SA Low Level Driver

Release Notes

Applies to Product Release: 04.00.00.04
Publication Date: Jan 15, 2020

Contributors to this document

Texas Instruments, Incorporated
20250 Century Boulevard
Germantown, MD 20874 USA

Contents

Overview....1
Licensing....1
Delivery Package....1
Installation Instructions....2
Customer Documentation List....3
LLD Dependencies....3
Label and Version Information....3
Resolved Incident Reports (IR)....3
Known Issues/Limitations....4
Component Compatibility....4
New/Updated Features and Quality....4

Release Notes

SA Low Level Driver 04.00.00.04

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/F9, 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
- Example and unit test code.
- API reference guide
- Software Manifest Documentation

This release notes is for SA LLD version 3.0.0.18(3_0_0_18)
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.

Licensing

Please refer to the software Manifest document for the details.

Delivery Package

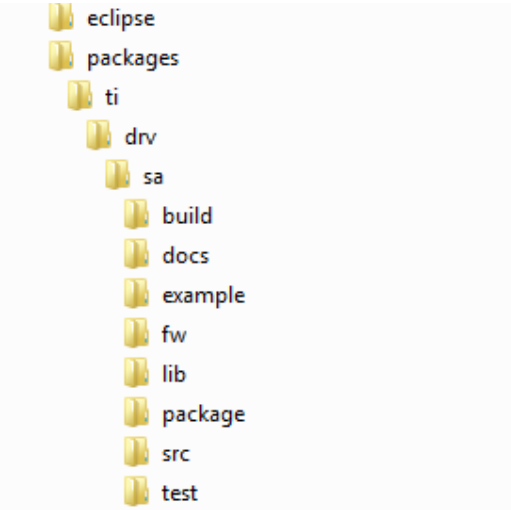
There is no separate delivery package. The SA LLD is being delivered as part of PDK within the MCSDK.

Installation Instructions

The LLD is currently bundled as part of Platform Development Kit (PDK) within the MCSDK. Refer installation instruction to the release notes provided for PDK.

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/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.
`eclipse`	The directory contains the Eclipse plug-in help files
`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_3_0_SoftwareManifest.pdf
4	User Guide	sa\docs\UserGuide_SA_LLD.pdf

LLD Dependencies

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

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.04.00.00.04

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

Known Issues/Limitations

Table 3 Known Issue/IRs for this Release

IR Parent/ Child Number	Severity Level	IR Description
102793	Minor	SA LLD Unit Test does not support IPSec Tunnel which is half duplex

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.

k2k SoC: PDK package released with MCSDK version 03.01.04.07 and above
k2h SoC: PDK package released with MCSDK version 03.01.04.07 and above
k2l SoC: PDK package released with MCSDK version 03.01.04.07 and above
k2e SoC: PDK package released with MCSDK version 03.01.04.07 and above

New/Updated Features and Quality

This is an **engineering release,** tested by the development team only.

Release 4.0.0.04

- support for J721e SOC added

Release 4.0.0.03

- PRSDK-4654: remove build-id section from mpu linker command file

Release 4.0.0.02

- Fixed static analysis issues

Release 4.0.0.01

- Sa Unit Test/Examples are built on top of UDMA/SciClient for AM65x

Release 4.0.0.00

- Added support for AM65x
- Resolved IRs

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-4177	Major	SA_MultiCoreExample/SA_MultiCoreExample_w3gpp DSP LE/BE hangs on k2h

Release 3.0.0.21

- Updated buildlib.xs to include Rules.make from ti/build infrastructure.
- Resolved IRs as listed at section "Resolved Incident Reports(IR)".

Release 3.0.0.20

- Update to gcc-arm-none-eabi-6-2017-q1-update tool chain showed limitations on reading the fread with "rb" mode. All SA basic examples that were doing fread with "rb" failed as the file was getting read as ASCII file instead of binary file. Hence moved all the basic examples to be independent of fread operations to get the examples functional.
- Resolved IRs as listed at section as below:

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-3236	Major	SA BASIC EXMPLE ARM-LE hangs on K2E/K2L

Release 3.0.0.19

- Support for gcc-arm-none-eabi-6-2017-q1-update tool chain added
- Resolved IRs as listed as below:

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-3154	Major	SA Unit Test/SA BASIC EXMPLE ARM-LE: fails

Release 3.0.0.18

- Bug fixes in this release.

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-2349	Major	SA leaks failed authentication packet

- Resolved IRs as listed at section "Resolved Incident Reports(IR)".

Release 3.0.0.17

- Bug fixes in this release.
- Resolved IRs as listed as below:

IR Parent/ Child Number	Severity Level	IR Description

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-1481	Major	SA Unit test fails on K2G (NSS_LITE)
PRSDK-1779	Minor	Update PA Instance size change for SA Unit Test

Release 3.0.0.16

- Bug fixes in this release.
- Resolved IRs as listed as below.

IR Parent/ Child Number	Severity Level	IR Description
PLSDK-917	Major	SA firmware update for AES-GMAC does not include IV GMAC ESP payload authentication

Release 3.0.0.15

- Added a feature to select Air Cipher Engine over the default Encryption engine for algorithms such as AES_CTR. The new control bit breaks the backward compatibility and should be good as long as "*Sa_DataModeConfigParams_t*" structure is memset to zero before setting any parameters in the data mode configuration structure in older releases.
  - There are certain types of algorithms such as AES_CTR which both of these engines are capable of executing. By default the data mode selects to use the encryption engine for the algorithms such as AES_CTR. If Application needs to use air cipher engine, it can select the air cipher engine by setting this bit.
- Resolved IRs as listed as below:

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-79	Major	Match SA Block Count to Software Packet Instance Buffers
PRSDK-757	Major	Need fix in SA LLD to support data mode operations for AES using cipher engine only.

Release 3.0.0.14

- Resolved IPSec Replay window size firmware fixes
- Resolved IRs as listed below

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-634	Minor	IPSEC ESP replay issue

Release 3.0.0.13

- Support for K2G device added

- This feature introduces a lite SA library to be used for K2G device. The device that is under interest needs to be specified in the RTSC configuration file as below:

```
var Sa = xdc.useModule ('ti.drv.sa.Settings');
Sa.deviceType = "k2g";
```

- Resolved IRs as listed as below.

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00120945	Major	CountC update by SA Air Ciphering firmware
SDOCM00118020	Minor	Sa Core Dump feature for Air Cipher Specific packet information for Gen1 devices not supported
SDOCM00120785	Major	SA PKA test failed on ARM
SDOCM00120083	Minor	SA LLD: Document Run time Encryption Auth Key Change not supported for IPSec/AirCipher
SDOCM00120777	Minor	SA: Document internal Key storage feature not supported for KeyStone (C6678)
SDOCM00120822	Minor	RTOS: SA LLD docs folder includes manifest for older R2.0 version
SDOCM00118020	Minor	Sa Core Dump for Air Cipher specific packet information for Gen1 devices not supported
SDOCM00120945	Major	CountC update by SA Air Ciphering Firmware
SDOCM00120785	Major	SA PKA test failed on ARM

-

Release 3.0.0.12

- Support for ARM A15 RTOS library added

Release 3.0.0.11

- Support for Sa Debug feature added
  - This feature enables to get some deeper information in SASS when unintended results show up during SA operations. The feature enables to halt the PDSP firmware and/or collect internal debug information during Air Cipher case. Note that Air Cipher debug information log is not supported for NSS GEN1 devices.

    During Sa_create the control bit flags can be set to trigger PDSP halt during any system error conditions. When PDSP is halted, the core dump can be done (Please refer to Sa_CoreDump API for details.

    For NSS_GEN2, Air Cipher channels, few more debug logging is provided in SA firmware to analyze the data. This can be triggered by setting the control flag

sa_CONFIG_CTRL_BITMAP_TRIGGER_PKT_INFO_LOG during Sa_Create. Sa_CoreDump API can be used to dump the debug information. Please refer to Sa Unit Test application (running on DSP) which demonstrates the usage of Sa_CoreDump API.

Warning: sa_CONFIG_CTRL_BITMAP_TRIGGER_PKT_INFO_LOG control bit should never be enabled for non-Air Cipher cases, which may lead to unexpected results.

- Resolved IRs as listed below.

IR Parent/ Child Number	Severity Level	IR Description
115343	Major	SASS hangs due to invalid security context
116010	Major	SASS: Firmware should use constant IV size for 3GPP CMAC operation
117881	Major	SA firmware support error detect feature
115088	Minor	SA user space shared object libraries to have same shared object lib name (SONAME) across devices supported

Release 3.0.0.10

- This release includes the IPSEC Post-Processing function (Sa_chanReceiveData()) enhancements to process the inner-IP reassembled packet when the RA engine is enabled on the keystone2 NSS Gen2 devices such as K2L and K2E. The application should setup and include the new data structure Sa_RxPayloadInfo_t in the Sa_PktInfo_t when API Sa_chanReceiveData is invoked:
  - Sa_PktInfo_t. validBitMap |= sa_PKT_INFO_VALID_RX_PAYLOAD_INFO
  - Sa_RxPayloadInfo_t
    - ipOffset: Specify the offset to the outer IP in the packet in bytes
    - ipOffset2: Specify the offset to the inner IP in the packet in bytes (0: indicates there is no inner IP)

  The same change is also applied to the data structure salldIpsecEspTxRxInfo_t when the global ESP post-processing utility function salld_esp_post_proc_util is invoked.
- Resolved IRs as listed below.

IR Parent/ Child Number	Severity Level	IR Description
115089	Major	SA LLD: Enhance IPSEC Post-Processing functions to handle RA-based Inner IP reassembled packets

Release 3.0.0.9

- Resolved IRs as listed below.

IR Parent/ Child Number	Severity Level	IR Description
114357	Major	SA LLD: Enhance IPSEC Firmware tx processing to handle PASS Long Info

Release 3.0.0.8

- Enhanced SA LLD firmware images to be compatible with PASS which supports 64-bit system timestamp with variable size of PASS packet information.
- Enhanced data mode to support WiMax based AES CCM encryption data mode operation
- Resolved IRs as listed below.

IR Parent/ Child Number	Severity Level	IR Description
113998	Major	Update SA LLD to be compatible with PASS which supports 64-bit timestamp
113755	Major	SA firmware may overwrite the timestamp field of the CPPI descriptor unexpectedly
113207	Major	SA LLD has badly aligned structures

Release 3.0.0.7

There are no real changes at the LLD and firmware images. The example and test projects are updated to be compatible with the latest PA LLD 03.00.01.00.

Release 3.0.0.6

This release includes resolved IPs as listed below:

IR Parent/ Child Number	Severity Level	IR Description
108335	Minor	Four SA projects do not compile for simulator due to undefined identifiers
108539	Minor	Top level “make all” using devkit fails for SA, while executing the rule “tests”

Release 3.0.0.5

This release includes resolved IPs as listed below

IR Parent/ Child Number	Severity Level	IR Description
102646	Major	SA Examples do not use RM
106712	Major	SA Multi-core example crashed at K2L EVM

IR Parent/ Child Number	Severity Level	IR Description
107309	Major	SA_BasicExample_w3gpp_K2HBiosExampleProject does not work
107510	Major	SASS: Incorect CountC is inserted into the packet in LTE To-air direction
107516	Minor	SA LLD Unit Test enhancements: Decrypted packet verification
107896	Major	SA LLD: Snow3G LTE mode fails to cipher packets correctly (Snow3G and ZUC F8/F9 modes)

Alpha Release 3.0.0.4

This release provides a single library for all keystone2 devices. However, there are different sets of SASS firmware images for the first generation and second generation SASS respectively. The corresponding SASS firmware image files are stored at the following sub-directories.

- Kepler/Hawking: fw/v0
- Lamarr/Edison: fw/v1

The top layer header file salld.h is shared for both generations of SASS where certain advanced features are only applicable to the second generation SASS as documented at Updated Features section for Alpha release 3.0.0.2.

To create SA instance of the second generation SASS, the application should set the control bit sa_SIZE_CONFIG_SASS_GEN2 at data structure Sa_ChanSizeCfg_t when it invokes API Sa_getBufferReq(), Sa_create() and Sa_start().
.

Alpha Release 3.0.0.3

This release includes all the new features from SA LLD 2.0.1 and 2.0.3.

Alpha Release 3.0.0.2

This is the first official engineering drop of SA LLD for the second generation SASS (Security Accelerator Sub-System) on advanced Keystone2 devices. The supported feature list is compatible with SA LLD version 2.0.0.4 and all the existing APIs are 100% backward compatible.

The second generation SASS provides several feature enhancements. Overview of feature enhancements and APIs are highlighted below, please refer to SA LLD header file salld.h or SA LLD doxygen document for details. The SA unit test program under sa/test provides some examples and sample codes for all feature enhancements.

- Two sets of IPSEC processing Engines

  To increase the encryption/authentication capacity of IPSEC channels, the second generation SASS contains two sets of IPSEC processing engines. The application should make the following changes to use this feature enhancement:
  - Call Sa_downloadImage() to download the third image Sa_php3 in addition to the first two images Sa_php1 and Sa_php2.

- Specify one of the following IPSEC engine selection policy at parameter engSelMode of SA configuration structure Sa_Config_t
  - sa_EngSelMode_LOADBALANCED: SA will select the set of engines with lower channel utility (default)
  - sa_EngSelMode_ROUNDROBIN: A will use each set of engines interleaved in order for each channel created
  - sa_EngSelMode_USERSPECIFIED: User will specify engine set to use when creating new channel by setting parameter engSelect in Sa_ChanConfig_t.

- Increase IPSEC replay window size to 1024

The second generation SASS supports the IPSEC replay window size up to 1024 in stead of 128 as supported by the original SASS. The side effect of using large size replay window is that the size of corresponding security context will be increased by 128 when the replay window size exceeds 128.

- Support Air Ciphering algorithm ZUC F8/F9 and Sonw3G F9

The second generation SASS supports air ciphering/authentication algorithm ZUC F8/F9 and Snow3G F9. To select ZUC, set cipherMode to sa_CipherMode_ZUC_F8 and/or authMode to sa_AuthMode_ZUC_F9. To select Snow3G F9, set authMode to sa_AuthMode_SNOW3G_F9.

- Simultaneous Air Ciphering and Authentication

The first generation SASS only supports limited Air Ciphering authentication algorithm such as Kasumi-F9 and it is not able to perform air-ciphering and authentication in a single pass due to hardware limitation. All those restrictions have been removed in the second generation SASS.

- Local PKTDMA within Network sub-system (NSS)

The local PKTDMA can be used to deliver packets between SASS and PASS. A new parameter ctrlBitfield is added into the SASS Destination Information data structure Sa_DestInfo_t to enable this feature. To deliver SASS output packets to PASS through local PKTDMA, the application should set control bit sa_DEST_INFO_CTRL_USE_LOC_DMA of ctrlBitfield. To deliver PASS output packets to SASS through local PKTDMA, the application should set the destination type to pa_DEST_SASS_LOC_DMA in stead of pa_DEST_SASS at the PASS Routing information data structure paRouteInfo_t or paRouteInfo2_t.

Release 3.0.0.0 through Release 3.0.0.1

- Internal Releases

Release 2.0.2.0

- Example applications are enhanced to be linked with static libraries when there is no environment variable "USEDYNAMIC_LIB" set to "yes".
- When USEDYNAMIC_LIB=yes environment variable is set, the example applications are not forced to link with static libraries.

- Resolved IR as listed below

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00103233	Major	LLD makefiles need to support shared libraries
SDOCM00102566	Minor	Add an error check and return error if the required 3GPP ciphering modes are not enabled

Release 2.0.1.4

- Production release of SA LLD 2.0.1
- SA LLD 2.0.1 Migration Information
    - Multiprocess Support is added – Below two new elements are added to 'Sa_Config_t' config structure, which needs to be set to 'zero' if not used for multiprocessing.
        - void *instPoolBaseAddr : /**< Base address of the global shared memory pool from which global LLD instance & channel instance memory is allocated.*/
        - void *scPoolBaseAddr: /**< Base address of the global shared memory pool from which SA security context memory is allocated. This is a DMA'able memory */
    - Please memset the 'Sa_Config_t' structure to zero before initializing any configuration elements.
    - Updated new environment variables in armsetupenv.sh script to enable or disable the SA 3gpp support in the SA examples.
- Resolved IR as listed below

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00102453	Major	Enable Multiprocess support for SA LLD
SDOCM00102765	Major	SA LLD does not return Channel Stats when an IPSec tunnel is half duplex
SDOCM00101388	Minor	SA LLD examples are not restartable
SDOCM00102767	Minor	SA test makefiles use SA_INSTALL_PATH incorrectly
SDOCM00102769	Minor	SA test makefiles do not have support for building with SA_3GPP_SUPPORT
SDOCM00102842	Major	SA LLD library should be compiled with -mno-unaligned-access compiler flag for ARM User space

Release 2.0.1.3, Release 2.0.1.2

- Internal Release

Release 2.0.1.1

- Pre-release demonstrating multiprocess example in SA LLD.

Release 2.0.1.0

- Initial release (Pre-release) for Multiprocess support in SA LLD

Release 2.0.0.6

- Resolved IRs as listed below

IR Parent/Child Number	Severity Level	IR Description
SDOCM00100043	Major	SA LLD RN should indicate the Migration notes for ARM UserSpace LLD build from Keystone1
SDOCM00100931	Minor	ARM SaExample k2k makefile is missing from the SA LLD package
SDOCM00101341	Minor	SA LLD: System statistics not updated at certain scenario

Beta Release 2.0.0.5

- Resolved IRs as listed below.

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00099200	Major	sa example projects are not released with the bundle
SDOCM00100678	Major	Memory overwritten in function Sa_create() - SA LLD 2.00.00.04
SDOCM00099198	Major	SA Project Create does not have a Linux counter part
SDOCM00100410	Minor	Use by LLD makefiles of gcc instead of ld to perform library link fails with certain gcc versions

Alpha Release 2.0.0.4

- This release includes the following feature enhancements
  - Keystone2 support
  - Merged (Sync up) Keystone1 SA LLD 1.0.5.4 features. Refer to Keystone1 SA LLD 1.0.5.4 release notes for details.
- Resolved IRs as listed at section "Resolved Incident Reports (IR)".

NOTE: If you are using PDK Keystone2 1.0.0.7, please manually update the #define for "cslr_device.c" files under "ti\pdk_keystone2_1_00_00_07\packages\ti\csl\device\k2k\src" and "ti\pdk_keystone2_1_00_00_07\packages\ti\csl\device\k2h\src"as below.

#define CSL_NETCP_CFG_SA_CFG_REGS (0x02000000 + 0xC000)

This is fixed in next Keystone2 PDK releases.

Release 2.0.0.0 through Release 2.0.0.3

- Internal Releases