TPM 2.0 System API Library
RELEASE DOCUMENT

Date:	7/14/2015
Contacts:	Will Arthur, Intel Corporation
Name:	Tpm 2.0 system API library
Require-ments:	TPM 2.0 devices or MS TPM 2.0 Simulator 01.22
	0.97 Release of TPM 2.0 Library and Test Code **Changes:** • **Fixed resource manager issues with leaving objects and session contexts in TPM memory. This was causing a 902 error on 2nd pass of PolicyTests. And it could have caused issues when error conditions occurred, because in those cases, the contexts weren't being evicted.** • **Changed TAB/RM into a separate executable (daemon).** • **Added code to save context in RM table when an object is context loaded.** • **Added code to get hierarchy from context when object is context loaded.** • **Fixed bug: if LoadContext fails when loading objects it should exit ResourceMgrSendTpmCommand immediately. Instead it was loading other objects and proceeding through the rest of ResourceMgrSendTpmCommand function.** • **Added targeted test to tpmclient.cpp to make sure that hierarchy is saved correctly for ContextLoad command.** • **Fixed issues with TCTI: opaque data shouldn't be defined in tss2_tcti.h file.** • **Fixed makefile issue: under Windows, it was using mkdir command instead of md.** • **Fixed issue with definition of TSS2_TCTI_POLL_HANDLE in tss2_tcti.h file.** • **Fixed bug: wasn't handling case for TPM errors correctly in CheckPassed.** • **Added code to print level-specific messages when errors occur.** • **Changed CheckOverflow to return SAPI error level for errors. Other levels of TSS that call this function will alter the error level field.** • **Fixed resource manager to properly handle EvictControl commands. Before, if a persistent object was needed, the RM would give a 0xc0002 error.**

	• **Fixed printf's in resource manager so that they only print the right # of characters.** • **Added test for EvictControl.Fixed TestEncryptDecryptSession to work with 1.22 simulator.** • **Fixed TestShutdown to work with 1.22 simulator.** • **Added code to check that TPM2B output parameters' size fields are set to 0 for following structures: TPM2B_ECC_POINT, TPM2B_PUBLIC, TPM2B_NV_PUBLIC, and TPM2B_CREATION_DATA.**
4/16/15	**0.96 Release of TPM 2.0 Library and Test Code** **Changes:** • **Added buffer overrun checks to all SAPI code.** • **Added buffer overrun checks to resource manager code.** • **Added code to Part 3 functions to properly handle null pointers for output parameters.** • **Auto-generated most of the SAPI code from the TPM 2.0 specification.**

1/26/15	**0.95.1 Release of TPM 2.0 Library and Test Code** **Changes:** • **Fixed bug in CreatePrimary and Create: for one-call and decrypt session case, they were copying first parameter from incorrect pointer.** • **For CopyCreationDataOut, CopyECCPointOut, CopyNvPublicOut, CopyPublicOut added placeholder for return code if size != 0 when called. To be filled in when TSS WG decides on error code.** • **Fixed bugs in CopySensitiveCreateIn and CopySensitiveIn: they shouldn't look at the size.** • **Fixed bugs in CopyECCPointIn, CopyNvPublicIn, CopyPublicIn, CopySensitiveIn, and CopySensitiveCreateIn: not handling NULL outpul parameters correctly.** • **Changes all instances of calls to ExecuteFinish to a timeout that works for all cases including communicating with the simulator over the network.** • **Fixed call to LoadExternal in TestUnseal--needed to pass in a NULL pointer for the inSensitive parameter.** • **Fixed bug in CreatePrimary: not passing correct pointer for inSensitive.** • **Fixed timeouts for all ExecuteFinish calls in test application.** • **Fixed bugs in RM: cases where I wasn't handling errors and then parsing data that hadn't been received. Caused seg faults under Linux.** • **Fixed timeout for async Startup test.** • **Fixed SocketReceiveTpmResponse for blocking case.** • **Fixed bug in ExecuteFinish: BAD_SEQUENCE error generated early in function was getting overwritten by INSUFFICENT_RESPONSE error.** • **Fixed bug in ExecuteFinish: it was always setting timeout to 0 instead of TSS2_TCTI_TIMEOUT_BLOCK.** • **Fixed bug in resource manager: error level for non-TPM errors was getting overwritten with resource manager error level.**

	• **Replace Implementation.h with implementation.h.**
	• **Changed name of TPMB.h tpmb.h**
	  • **Added code to dynamically work around simulator 1.19 bugs:**
	  ○ **GetCapability with bad property returns different error code.**
	  ○ **Shutdown with bad value for shutdownValue causes TPM to go into failure mode.**
	• **Fixed overlap in error codes: TSS2_BASE_RC_NOT_SUPPORTED and TSS2_BASE_RC_BAD_TCTI_STRUCTURE had same value.**
	• **Cleaned up all app level error codes.**
	• **Added code to RM and simDriver to support timeout on receive calls.**
	• **Added code to properly handle TPM errors in ExecuteFinish. Previously it was ignoring these errors, which meant that the rest of the _Complete call would try to unmarshall non-existent response data. Added test case for this.**
	• **Added support for cancel commands and tests for this.**
	• **Added help text for command line options.**
	• **Fixed bug with ordering of -startAuthSessionTest command line parameter: if it was not the last option, tpmclient would fail.**
	• **Added code to reset dictionary attacks to start of tpmclient tests: this works around an issue where the simulator doesn't seem to completely clear the dictionary attack counter.**
	• **Added support for TCTI setLocality to resource manager and sim driver and made test app use this.**
	• **Added RM tests.**
	• **Added code to RM to evict contexts for objects, sequences, and sessions whose handles are returned by commands.**
	• **Fixed bugs related to ContextLoad.**
	• **Added code to properly support ContextSave.**
	• **Fixed bug in EvictContext: it was updating lastSessionSequenceNum even if the ContextSave command failed.**
	• **Added proper error code levels to all RM errors.**
	• **Added code to LoadContext function to output TPM formatted error codes.**
	• **For Create and Load commands, added proper handling of errors if parent handle not found.**
	• **Fixed handling of RM errors that occur during command send.**
	• **Fixed bug in simDriver init function. A second TCTI context being initialized was re-initing the whole driver.**

	• **Added tests for bad session handle, both in handle area and in authorization area.**
	• **Updated to latest 1.19 header files.**
	• **Fixed bugs in resource manager:**
	  ○ **FindOldestSession wasn't working correctly—it was just finding the first one.**
	  ○ **HandleGap needed to un-gap all the session contexts from the older interval. It wasn't doing that.**
	• **Fixed bug in handling of command line options—specifying none would cause program to error out.**
	• **Fixed issues in cleanup of TestStartAuthSession test. It was leaving some sessions alive.**
	• **Added command line option to run the StartAuthSession tests by themselves.**
	• **Updated copyright notices on all files.**
	• **Added support for command line control of debug message levels.**
	• **Changed test app to use linked list of session structures instead of fixed array. This fixed a host of issues.**
	• **Added new error level for resource manager for errors received from TPM from commands sent by RM.**
	• **Added error return for insufficiently sized response to ExecuteFinish function.**
	• **Fixed bugs in Certify, CertifyCreation, Commit, Create, CreatePrimary, and GetCapability: if null used for return parameters, the function would fail.**
	• **Added gap support to resource manager.**
	• **Added support to resource manager for kicking out oldest session if max sessions have been started and a new one is being created.**
	• **Added getCap calls to RM init function for getting max sessions and gap limit.**
	• **Added code to teardown the RM.**
	• **Added test for session gapping.**
	• **Fixed bug in SimpleHmacOrPolicyTest where it was re-creating the global sysContext causing failures in later tests because the context was too small.**
	• **Fixed a bug in ExecuteFinish. If response is too small, code was just using the command buffer as the response buffer instead of returning an error.**
	• **Added code to proactively detect MAX_ACTIVE_SESSIONS.**
	• **Fixed some places in test app where I wasn't deleting entries from the sessions table.**

	• **Added SAPI library subproject to test app project. This allows a one-touch build in Visual Studio.** • **Removed 'extern "C"' statement from resourcemgr.c file. Not needed and causes problems with some compilers.** • **Removed unneeded includes from resource manager source.** • **Added changes to return error codes from TAB/RM and layers underneath in a response byte stream.** • **Fixed build warnings related to size mismatch of connectionId.** • **Changed TeardownSysContext to zero out freed context pointer.** • **This helps prevent double free errors.** • **Fixed bug in EncryptDecryptXOR: wasn't setting the size of the outputData buffer.**

10/17/14	**0.95 Release of TPM 2.0 Library and Test Code** **Changes:** • **Added support for Shutdown/Startup and effects on saved contexts.** • **Added support for stClear bit objects. On a TPM Restart, objects with this bit set will be removed from the TAB/RM entry list.** • **Fixed intermittent access violation bug with GetSetDecryptParamTests function. I was reading off the end of the nvWrite buffer.** • **Added TCTI teardown function.** • **Fixed bug in Tss2_Sys_GetContextSize function: it was getting the requested size only, not the requested size plus the context blob's size. Problem was an associativity issue with ternary conditional ?: operator.** • **Removed most instances of sysContext in tpmclient.cpp. Now most tests use the global one.** • **Re-architected TAB/RM:** • **Added TAB functionality.** • **Changed RM from reactive mode to proactive mode. Now instead of reacting to error codes from the TPM that indicate no enough slots, it guarantees that the TPM is always ready for each command (all slots freed after execution of each command).** • **Added TCTI layer below RM to talk to driver. This allows making calls into the SAPI library from the RM without recursing into the RM again. With the separate TCTI context, the RM can route SAPI calls to talk directly to the driver. This fixed the virtual/real handle mess that was occurring with recursively entering the RM.** • **Added function pointers to TAB/RM for functions that might need to be different based on the environment that TAB/RM is running in: malloc, free, printf.** • **Replaced the fixed length arrays of RM structures with linked list structures and appropriate functions.**

	• **Fixed some cases of using pointers before checking that they're not NULL.** • **Fixed bugs in marshalling/unmarshalling routines and added some missing unmarshalling functions.** • **Fixed hash sequence test.** • **Fixed bugs in CopyCapabilityDataOut function for algorithms.** • **Fixed bug with ExecuteAsync: passed in BE size to transmit call. Needs to be host-endian.** • **Added and corrected error codes to match latest SAPI spec.** • **Removed pack pragma from header files for external interfaces.** • **Added MAX_NV_BUFFER_SIZE and used for max size of MAX_NV_BUFFER_2B.** • **Changed on bit fields in TPM2 data structures to unsigned int. Previously the compiler was generating incorrect code because these were int bit fields.** • **Cleaned up TestHash function.** • **Added code to TestHash to calculate and validate a hash.** • **Added code to TestHash to force a flush of an active sequence and then use it to finish the hash calculation.** • **Added code to SimpleHMACTest to read the NV index back.** • **Added SimpleHMACOrPolicyTest function which helps illustrate the difference between HMAC and policy sessions.**

08/01/14	**0.93 Release of TPM 2.0 Library and Test Code** **Changes:** • **Fixed bad parameters on call to GetEncryptParam. This only failed on Linux systems.** • **Fixed minor build errors under Linux.** • **Added IsSession routine and fixed all instances in resource manager where a handle is checked for being a session handle (some were incorrect).** • **Eliminated unneeded code in TestPolicy.** • **Added RollNonces function and used for all tests for HMAC and policy sessions.** • **Changed how nonce's are setup after StartAuthSession. Before they were being inherently rolled in preparation for first command. Now the RollNonces routine will need to be called before the first command. This makes handling of the nonces consistent for all code that needs to roll them.** • **Added TCTI malformed response error code.** • **Added simple HMAC test.** • **Fixed bug in StartAuthSession: wasn't marshalling symmetric parameter properly if algorithm was TPM_ALG_XOR.** • **Fixed bug in SetDecryptParam: when inserting a decrypt param, the code wasn't updating the command size field.** • **Fixed bug in ExecuteFinish: wasn't returning TPM error code if no other errors had occurred.** • **Added test for session parameter encryption and decryption.** • **Fixed bug in KDFa function: if key size was zero, this was just returning success, but not generating a key. That behavior is specific to session key generation not to the underlying KDFa function. Upleveled that code into StartAuthSession function so that it only occurs in the session key generation case.** • **Changed NV attributes for all NV indices to add orderly attribute. This helps, but doesn't entirely relieve, NV wearout issues with the tests.** • **Removed an unused input parameter from ComputeCommandHmacs and CheckResponseHmacs.**

	• **Changed NV attributes for all NV indices to add orderly attribute. This helps, but doesn't entirely relieve, NV wearout issues with the tests.** • **Removed an unused input parameter from ComputeCommandHmacs and CheckResponseHmacs.** • **Added more descriptive error codes to StartAuthSession function.** • **Added TpmHashSequence function. Used this build password/PCR policy.** • **Added more policy tests:** **password/PCR** **authValue** **password** • **Fixed a bunch of resource manager issues. Many of these were exposed by the new policy tests.** • **Added code to flush context of session handles I'm not using.** • **Added GetTestResult functions (had missed these previously)** • **Updated resource manager to properly handle sessions. Before we were not swapping them in as needed.** • **Added tests for asyncronous and syncronous non-one call to Startup tests.** • **Added GetTestResult tests.** • **Added test to create a bunch of sessions. This test found some resource manager issues.** • **Updated readme.docx file. Now tests can run with V1.15 version of MS simulator.** • **Made test app work with MS simulator version 1.15. Had to add command to turn on NV. Before this change, when running against MS simulator, TPM2_Startup would fail with 0x923 error: "ERROR: WARNING, TPM_RC_NV_UNAVAILABLE: the command may require writing of NV and NV is not current accessible."** • **Changed NO_RESPONSE_RECEIVED error code to IO_ERROR.** • **Removed DRIVER_NOT_FOUND and DRIVERINFO_NOT_FOUND error codes.** • **Cleaned up defines for MS simulator commands.**

06/17/14	**0.92 Release of TPM 2.0 Library and Test Code** **Changes:** • **Fixed bugs in sockets send and receive code. Needed to account for actual bytes sent/received instead of assuming them. This was causing intermittent errors when looping continuously on the tests and running the tests remotely (on a different host system than the simulator was running on).** • **Fixed SAPI and test app builds to not fail if directories are already present. Suppressed error messages related to mkdir.** • **Turned on compiler warnings and fixed all issues when building under Ubuntu Linux.** • **Fixed error in readme.docx file. I was specifying the wrong version of the simulator.** • **Fixed error handling if sockets interface fails to connect.** • **Fixed build error: now I make directories that are needed.**
06/04/14	**0.91 Release of TPM 2.0 Library and Test Code** **Changes:** • **Added code optimized builds to System API library code** • **Added warning flags to compiler command lines.** • **Fixed all compiler warnings when built under Windows and Linux.**

05/28/14	**0.90 release of Tpm 2.0 library and test code** • **Ported existing functionality to latest SAPI spec.** • **Cleaned up and added comments to PasswordTest.** • **Added support for encrypt/decrypt sessions with one-call functions.** • **Added cleaned up and reorganized header files that comply with latest SAPI specification.** • **Added changes for supporting get/set encrypt/decrypt functions.** • **Added latest header file that corresponds to version 1.03 of TPM 2.0 specification.** • **Added debug display of command string for each command being run.** • **Added command line flag to slow down test display for demo purposes.** • **Fixed problem of hang when looping through tests. Sessions table was running out of entries because we weren't removing sessions that were closed.** • **Fixed issue with resource manager. All virtual handles had the high nibble set to 0xff. Now the high nibble is left intact so that applications can determine the type of the handle.** • **Added option to loop the tests continuously.** **NOTES:** **1. Testing is not comprehensive. See test code to see what's tested. Please report any bugs found so that fixes can be rolled out.** 2. **Range checks within SAPI code not yet implemented.** 3. **Still need to add support for separate debug and production builds. Production build will be optimized for code size.**
12/16/13	**0.82 release of Tpm 2.0 library and test code** **NOTE: HMAC and cpHash calculations are only supported for NV Read and NV Write commands currently. The system API changes to support this have been prototyped for these commands and are awaitingTSS approval before being ported to all the other commands.** • **Added support for building and running system API code and tests under Linux.** • **Added command line options for host name and port to test application.**

12/02/13	**0.81 release of Tpm 2.0 library and test code** **NOTE: HMAC and cpHash calculations are only supported for NV Read and NV Write commands currently. The system API changes to support this have been prototyped for these commands and are awaitingTSS approval before being ported to all the other commands.** • **Added support for TPM2_PolicyNvWritten command.** • **Altered tests to work with 1.01 simulator.** • **Fixed errors in readme.docx.**

11/19/13	**0.80 release of Tpm 2.0 library and test code** **NOTE: HMAC and cpHash calculations are only supported for NV Read and NV Write commands currently. The system API changes to support this have been prototyped for these commands and are awaitingTSS approval before being ported to all the other commands.** • **Fixed bugs in resource manager.** • **Fixed bugs with salted session tests.** • **Ported tests to work with 0.99 sim's version of support for bound sessions.** • **Fixed bugs in test code, with how key is generated for encrypting the salt for salted session tests.** • **Fixed a rather serious bug in HmacSessionTest: CopyNvPublicIn is called to copy a structure, but is had the side effect of modifying the first parameter. This function really wasn't designed to be used the way it is. Worked around the problem by resetting the pointer after calling CopyNvPublicIn. This problem showed up as a stack corruption issue that occurred during the 4th test. Basically the pointer moved enough after the first 3 tests to start corrupting other variables on the stack.** • **Added code to create a new session for reading/writing the NV index after it's first written. This tests the other case for bound sessions.** • **Automated runtime setup of key for salted tests.** • **Developed changes for NVRead/Write commands to use new 2-stage method for handling HMAC calculations.** • **Changed CopyPcrSelectionIn function so that it can be used by applications to generate policy hashes.** • **Fixed build error: changes in header files weren't causing TPM 2.0 library functions to be rebuilt.** • **Created CalcPHash helper function.** • **Changed HMAC session code to new architecture that doesn't use any helper function pointers.** • **Added routine to start policy sessions.** • **Added policy test code--not used currently.** • **Changed return code type form UINT32 to TPM_RC in tss_sysapi.h.** • **Changed "authHandle" to "sessionHandle" in sample code.** • **Debugged and fixed StartAuthSession2 function in test code.** • **Debugged and fixed first policy test.** • **Used new NvDefine function to help abstract some of the details of creating NV indices.** • **Used non-MS header file to build system API.** • **Cleaned up and reorganized files and directories.**

8/7/13	**0.67 release of Tpm 2.0 library and test code**
	
	NOTE: HMAC helper function callouts are only being done for NV Read and NV Write commands currently. The system API changes to support this are still being prototyped. After they are finalized, these changes will be extended to all functions that use sessions.
	
	New Features/Changes:
	• **Removed tis.h file. Not needed.**
	• **Updated headers with Intel license text.**
	• **Eliminated salted session test (because it doesn't work yet), and changed out.good file to match.**
	• **Plumbed in a resource mgr (doesn't actually do anything other than pass through at this time).**
	• **Split sockets driver into separate code module.**
	• **SALTED session test fixes:**
	  ○ **Fixed CopyRSAEncryptIon function--wasn't handling some cases correctly.**
	  ○ **Backed out change to make parameterSize passed to ComputeSessionHmacPtr function a UINT16. Needs to be UINT32.**
	• **For ComputeSessionHmacPtr, changed parameterSize to UINT16 to fix build warning.**
	• **Added BOUND and SALTED HMAC session tests. BOUND test works, but SALTED doesn't yet work.**
	• **Added code to delete an entity from the entity table.**
	• **Added code to work around an NV index anomaly with TPM simulator 0.98 and previous versions: after the first NV index write, the name changes. This causes the TPM's HMAC calculation to treat the index as if it's never the BOUND entity, even if it is. This is expected (but weird) behavior which will be fixed in 0.99 simulator.**
	• **Fixed bugs in KDFa().**
	• **Altered all APIs to use pointers to TPM input/output buffers.**

(cont)	• **Created two helper functions pointers for system API and used them for HMAC sessions.** • **Added support for HMAC session for NV read/write APIs.Added HMAC tests for unbounded/unsalted sessions.Fixed context save/restore functions.Created CopyNvPublicIn function and altered Tpm2_DefineSpace function to use it.** • **Created TpmHash function** • **Created TpmHandleToName function** • **Added HMAC tests for unbounded/unsalted sessions.** • **Fixed context save/restore functions.** • **Created CopyNvPublicIn function and altered Tpm2_DefineSpace function to use it.** • **Created TpmHash function** • **Created TpmHandleToName function** • **Documented helper function pointers in the system API header file.** • **Fixed formatting of prints of sized byte buffers in test app.** • **Added tests for TpmHandleToName function.** • **Fixed bug in TpmHmac function: needed to set size of result to 0 in case an error occurs.** • **Reorganized directories and moved files to make more logical sense.** • **Fixed bugs in CopySensitiveIn function: uninitialized size field, bad pointers, and incorrect increment of otherData at end of function.** • **Added functionality needed for KDFa function:** **ConcatSizedByteBuffer function** **CopySizedByteBuffer function** • **Added KDFa function in preparation for HMAC session test. Not tested yet.** • **Added LoadExternalHMACKey function. This function is called by TPM HMAC function.** • **Altered TpmHMAC function to call LoadExternalHMAC key function. This allows a better HMAC function pointer, one that complies with normal HMAC calling convention. Before it was TPM-specific.** • **Bumped up TPMBUF_LEN to 32k in tpmclient.cpp. This fixed overwriting problems during context save/restore function.** • **Fixed bugs in ContextLoad function: otherData wasn't initialized before it was used.** • **Fixed bug in Tpm20LoadExternal command: it wasn't properly marshaling the inPrivate data.**

	0.66 release of Tpm 2.0 library and test code **New Features/Changes** • **Added CertifyCreation function** • **Added EcEphemeral function** • **Added test for tspi_sys_TPM2_HashStart** • **Cleaned up for general TCG release**

4/10/13	***0.65 release of Tpm 2.0 library***
	
	New Features/Changes
	• ***All TPM 2.0 functions now supported.***
	• ***Limited testing done on following functions:***
	  • ***tspi_sys_TPM2_Startup***
	  • ***tspi_sys_Tpm2_SelfTest***
	  • ***tspi_sys_TPM2_GetCapability***
	  • ***tspi_sys_TPM2_Clear-tested***
	  • ***tspi_sys_TPM2_StartAuthSession***
	  • ***tspi_sys_TPM2_ClearControl***
	  • ***tspi_sys_TPM2_ChangeEPS***
	  • ***tspi_sys_TPM2_HierarchyChangeAuth***
	  • ***tspi_sys_TPM2_Extend***
	  • ***tspi_sys_TPM2_HashSequenceStart***
	  • ***tspi_sys_TPM2_SequenceUpdate***
	  • ***tspi_sys_TPM2_SequenceComplete***
	  • ***tspi_sys_TPM2_EventSequenceComplete***
	  • ***tspi_sys_TPM2_GetRandom***
	  • ***tspi_sys_TPM2_SaveState***
	  • ***tspi_sys_TPM2_PcrRead***
	  • ***tspi_sys_TPM2_NVRead***
	  • ***tspi_sys_TPM2_NVWrite***
	  • ***tspi_sys_TPM2_Unseal***
	  • ***tspi_sys_TPM2_PcrAllocate***
	  • ***tspi_sys_TPM2_DictionaryAttackLockReset***
	  • ***tspi_sys_TPM2_NV_Writelock***
	  • ***tspi_sys_TPM2_PolicyCommandCode***
	  • ***tspi_sys_TPM2_PolicyGetDigest***
	  • ***tspi_sys_TPM2_PolicyOr***
	  • ***tspi_sys_TPM2_PolicyRestart***
	  • ***tspi_sys_TPM2_LoadExternal***
	  • ***tspi_sys_TPM2_HierarchyControl***
	  • ***tspi_sys_TPM2_NV_UndefineSpace***
	  • ***tspi_sys_TPM2_Create***
	  •

(cont)	• ***tspi_sys_TPM2_Load*** • ***tspi_sys_TPM2_Quote*** • ***tspi_sys_TPM2_NV_ReadPublic*** • ***tspi_sys_TPM2_ChangePPS*** • ***tspi_sys_TPM2_NV_DefineSpace*** • ***tspi_sys_TPM2_PolicyLocality*** • ***tspi_sys_TPM2_PolicyPCR*** • ***tspi_sys_TPM2_CreatePrimary*** • ***tspi_sys_TPM2_Shutdown*** • ***tspi_sys_TPM2_PCR_Event*** • ***tspi_sys_TPM2_PolicyNV*** • ***tspi_sys_TPM2_NV_ReadLock*** • ***tspi_sys_TPM2_NV_UndefineSpaceSpecial*** • ***No testing done on all other 61 functions*** •
3/29/13	• **0.60 release of Tpm 2.0 library** **New Features/Changes:** • **First official release** • **Added changes to make it comply with TSS 2.0 system library API** • **Cleaned up and removed unneeded files.**