```
**********************************************************
   61466 Thu Oct 17 11:32:55 2013
new/usr/src/lib/pkcs11/pkcs11_tpm/common/api_interface.c
4215 pkcs11_tpm botches shared library initialization, murders its friends
**********************************************************
______unchanged_portion_omitted_

1241 CK_RV
1242 C_GetFunctionList(CK_FUNCTION_LIST_PTR_PTR ppFunctionList)
1243 {
1244         _init();

1244         PK11_Functions.version.major = VERSION_MAJOR;
1245         PK11_Functions.version.minor = VERSION_MINOR;
1246         PK11_Functions.C_Initialize = C_Initialize;
1247         PK11_Functions.C_Finalize = C_Finalize;
1248         PK11_Functions.C_GetInfo = C_GetInfo;
1249         PK11_Functions.C_GetFunctionList = C_GetFunctionList;
1250         PK11_Functions.C_GetSlotList = C_GetSlotList;
1251         PK11_Functions.C_GetSlotInfo = C_GetSlotInfo;
1252         PK11_Functions.C_GetTokenInfo = C_GetTokenInfo;
1253         PK11_Functions.C_GetMechanismList = C_GetMechanismList;
1254         PK11_Functions.C_GetMechanismInfo = C_GetMechanismInfo;
1255         PK11_Functions.C_InitToken = C_InitToken;
1256         PK11_Functions.C_InitPIN = C_InitPIN;
1257         PK11_Functions.C_SetPIN = C_SetPIN;
1258         PK11_Functions.C_OpenSession = C_OpenSession;
1259         PK11_Functions.C_CloseSession = C_CloseSession;
1260         PK11_Functions.C_CloseAllSessions = C_CloseAllSessions;
1261         PK11_Functions.C_GetSessionInfo = C_GetSessionInfo;
1262         PK11_Functions.C_GetOperationState = C_GetOperationState;
1263         PK11_Functions.C_SetOperationState = C_SetOperationState;
1264         PK11_Functions.C_Login = C_Login;
1265         PK11_Functions.C_Logout = C_Logout;
1266         PK11_Functions.C_CreateObject = C_CreateObject;
1267         PK11_Functions.C_CopyObject = C_CopyObject;
1268         PK11_Functions.C_DestroyObject = C_DestroyObject;
1269         PK11_Functions.C_GetObjectSize = C_GetObjectSize;
1270         PK11_Functions.C_GetAttributeValue = C_GetAttributeValue;
1271         PK11_Functions.C_SetAttributeValue = C_SetAttributeValue;
1272         PK11_Functions.C_FindObjectsInit = C_FindObjectsInit;
1273         PK11_Functions.C_FindObjects = C_FindObjects;
1274         PK11_Functions.C_FindObjectsFinal = C_FindObjectsFinal;
1275         PK11_Functions.C_EncryptInit = C_EncryptInit;
1276         PK11_Functions.C_Encrypt = C_Encrypt;
1277         PK11_Functions.C_EncryptUpdate = C_EncryptUpdate;
1278         PK11_Functions.C_EncryptFinal = C_EncryptFinal;
1279         PK11_Functions.C_DecryptInit = C_DecryptInit;
1280         PK11_Functions.C_Decrypt = C_Decrypt;
1281         PK11_Functions.C_DecryptUpdate = C_DecryptUpdate;
1282         PK11_Functions.C_DecryptFinal = C_DecryptFinal;
1283         PK11_Functions.C_DigestInit = C_DigestInit;
1284         PK11_Functions.C_Digest = C_Digest;
1285         PK11_Functions.C_DigestUpdate = C_DigestUpdate;
1286         PK11_Functions.C_DigestKey = C_DigestKey;
1287         PK11_Functions.C_DigestFinal = C_DigestFinal;
1288         PK11_Functions.C_SignInit = C_SignInit;
1289         PK11_Functions.C_Sign = C_Sign;
1290         PK11_Functions.C_SignUpdate = C_SignUpdate;
1291         PK11_Functions.C_SignFinal = C_SignFinal;
1292         PK11_Functions.C_SignRecoverInit = C_SignRecoverInit;
1293         PK11_Functions.C_SignRecover = C_SignRecover;
1294         PK11_Functions.C_VerifyInit = C_VerifyInit;
1295         PK11_Functions.C_Verify = C_Verify;
1296         PK11_Functions.C_VerifyUpdate = C_VerifyUpdate;
1297         PK11_Functions.C_VerifyFinal = C_VerifyFinal;
```



```
1298         PK11_Functions.C_VerifyRecoverInit = C_VerifyRecoverInit;
1299         PK11_Functions.C_VerifyRecover = C_VerifyRecover;
1300         PK11_Functions.C_DigestEncryptUpdate = C_DigestEncryptUpdate;
1301         PK11_Functions.C_DecryptDigestUpdate = C_DecryptDigestUpdate;
1302         PK11_Functions.C_SignEncryptUpdate = C_SignEncryptUpdate;
1303         PK11_Functions.C_DecryptVerifyUpdate = C_DecryptVerifyUpdate;
1304         PK11_Functions.C_GenerateKey = C_GenerateKey;
1305         PK11_Functions.C_GenerateKeyPair = C_GenerateKeyPair;
1306         PK11_Functions.C_WrapKey = C_WrapKey;
1307         PK11_Functions.C_UnwrapKey = C_UnwrapKey;
1308         PK11_Functions.C_DeriveKey = C_DeriveKey;
1309         PK11_Functions.C_SeedRandom = C_SeedRandom;
1310         PK11_Functions.C_GenerateRandom = C_GenerateRandom;
1311         PK11_Functions.C_GetFunctionStatus = C_GetFunctionStatus;
1312         PK11_Functions.C_CancelFunction = C_CancelFunction;
1313         PK11_Functions.C_WaitForSlotEvent = C_WaitForSlotEvent;
1314         if (ppFunctionList) {
1315                 (*ppFunctionList) = &PK11_Functions;
1316                 return (CKR_OK);
1317         } else {
1318                 return (CKR_ARGUMENTS_BAD);
1319         }
1320 }
______unchanged_portion_omitted_
```

```
**********************************************************
   41747 Thu Oct 17 11:32:56 2013
new/usr/src/lib/pkcs11/pkcs11_tpm/common/tpmtok_int.h
4215 pkcs11_tpm botches shared library initialization, murders its friends
**********************************************************
______unchanged_portion_omitted_

490 typedef  struct token_specific_struct token_spec_t;

492 /*
493  * Global Variables
494  */
495 extern void copy_slot_info(CK_SLOT_ID, CK_SLOT_INFO_PTR);

497 extern struct messages err_msg[];

499 extern token_spec_t token_specific;
500 extern CK_BBOOL initialized;
501 extern char *card_function_names[];
502 extern char *total_function_names[];

504 extern MECH_LIST_ELEMENT mech_list[];
505 extern CK_ULONG mech_list_len;

507 extern pthread_mutex_t native_mutex;

509 extern void *xproclock;

511 extern pthread_mutex_t pkcs_mutex, obj_list_mutex,
512         sess_list_mutex, login_mutex;

514 extern DL_NODE *sess_list;
515 extern DL_NODE *sess_obj_list;
516 extern DL_NODE *publ_token_obj_list;
517 extern DL_NODE *priv_token_obj_list;
518 extern DL_NODE *object_map;

520 extern CK_BYTE so_pin_md5[MD5_DIGEST_LENGTH];
521 extern CK_BYTE user_pin_md5[MD5_DIGEST_LENGTH];

523 extern CK_BYTE default_user_pin_sha[SHA1_DIGEST_LENGTH];
524 extern CK_BYTE default_so_pin_sha[SHA1_DIGEST_LENGTH];
525 extern CK_BYTE default_so_pin_md5[MD5_DIGEST_LENGTH];

527 extern LW_SHM_TYPE *global_shm;

529 extern TOKEN_DATA *nv_token_data;

531 extern CK_ULONG next_object_handle;
532 extern CK_ULONG next_session_handle;

534 extern CK_STATE global_login_state;

536 extern CK_BYTE  ber_AlgIdRSAEncryption[];
537 extern CK_ULONG ber_AlgIdRSAEncryptionLen;
538 extern CK_BYTE  ber_rsaEncryption[];
539 extern CK_ULONG ber_rsaEncryptionLen;
540 extern CK_BYTE  ber_idDSA[];
541 extern CK_ULONG ber_idDSALen;

543 extern CK_BYTE ber_md5WithRSAEncryption[];
544 extern CK_ULONG ber_md5WithRSAEncryptionLen;
545 extern CK_BYTE ber_sha1WithRSAEncryption[];
546 extern CK_ULONG ber_sha1WithRSAEncryptionLen;
547 extern CK_BYTE ber_AlgMd5[];
548 extern CK_ULONG ber_AlgMd5Len;
```



```
549 extern CK_BYTE ber_AlgSha1[];
550 extern CK_ULONG ber_AlgSha1Len;

552 extern CK_C_INITIALIZE_ARGS cinit_args;

554 /*
555  * Function Prototypes
556  */
557 void *attach_shared_memory();
558 void  detach_shared_memory(char *);

560 int API_Initialized();
561 void Terminate_All_Process_Sessions();
562 int API_Register();
563 void API_UnRegister();

565 void CreateXProcLock(void *);
566 int XProcLock(void *);
567 int XProcUnLock(void *);

569 void _init(void);
569 void loginit();
570 void logterm();
571 void logit(int, char *, ...);
572 void AddToSessionList(Session_Struct_t *);
573 void RemoveFromSessionList(Session_Struct_t *);

575 int Valid_Session(Session_Struct_t *, ST_SESSION_T *);

577 CK_BBOOL pin_expired(CK_SESSION_INFO *, CK_FLAGS);
578 CK_BBOOL pin_locked(CK_SESSION_INFO *, CK_FLAGS);
579 void set_login_flags(CK_USER_TYPE, CK_FLAGS *);

581 extern void init_slot_info(TOKEN_DATA *);

583 CK_RV update_migration_data(TSS_HCONTEXT,
584         TSS_HKEY, TSS_HKEY, char *, char *, BYTE *, BYTE *);
585 CK_RV token_rng(TSS_HCONTEXT, CK_BYTE *, CK_ULONG);

587 TSS_RESULT set_public_modulus(TSS_HCONTEXT, TSS_HKEY,
588     unsigned long, unsigned char *);
589 TSS_RESULT open_tss_context(TSS_HCONTEXT *);
590 CK_RV token_get_tpm_info(TSS_HCONTEXT, TOKEN_DATA *);

592 CK_RV clock_set_default_attributes(TEMPLATE *);
593 CK_RV clock_check_required_attributes(TEMPLATE *, CK_ULONG);
594 CK_RV clock_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

596 CK_RV counter_set_default_attributes(TEMPLATE *);
597 CK_RV counter_check_required_attributes(TEMPLATE *, CK_ULONG);
598 CK_RV counter_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

600 CK_RV compute_next_token_obj_name(CK_BYTE *, CK_BYTE *);

602 CK_RV save_token_object(TSS_HCONTEXT, OBJECT *);
603 CK_RV save_public_token_object(OBJECT *);
604 CK_RV save_private_token_object(TSS_HCONTEXT, OBJECT *);

606 CK_RV load_public_token_objects(void);
607 CK_RV load_private_token_objects(TSS_HCONTEXT);

609 CK_RV reload_token_object(TSS_HCONTEXT, OBJECT *);

611 CK_RV delete_token_object(OBJECT *);

613 CK_RV init_token_data(TSS_HCONTEXT, TOKEN_DATA *);
```

```
614 CK_RV load_token_data(TSS_HCONTEXT, TOKEN_DATA *);
615 CK_RV save_token_data(TOKEN_DATA *);
616 void copy_slot_info(CK_SLOT_ID, CK_SLOT_INFO_PTR);

618 CK_RV compute_sha(CK_BYTE *, CK_ULONG_32, CK_BYTE *);

620 CK_RV parity_is_odd(CK_BYTE);

622 CK_RV build_attribute(CK_ATTRIBUTE_TYPE,
623         CK_BYTE *, CK_ULONG, CK_ATTRIBUTE **);

625 CK_RV add_pkcs_padding(CK_BYTE *, UINT32, UINT32, UINT32);

627 CK_RV strip_pkcs_padding(CK_BYTE *, UINT32, UINT32 *);

629 CK_RV remove_leading_zeros(CK_ATTRIBUTE *);

631 CK_RV rsa_pkcs_encrypt(
632         SESSION *,
633         CK_BBOOL,
634         ENCR_DECR_CONTEXT *,
635         CK_BYTE *,
636         CK_ULONG,
637         CK_BYTE *,
638         CK_ULONG *);

640 CK_RV rsa_pkcs_decrypt(SESSION *,
641         CK_BBOOL,
642         ENCR_DECR_CONTEXT *,
643         CK_BYTE *,
644         CK_ULONG,
645         CK_BYTE *,
646         CK_ULONG *);

648 CK_RV rsa_pkcs_sign(SESSION *,
649         CK_BBOOL,
650         SIGN_VERIFY_CONTEXT *,
651         CK_BYTE *,
652         CK_ULONG,
653         CK_BYTE *,
654         CK_ULONG *);

656 CK_RV rsa_pkcs_verify(SESSION *,
657         SIGN_VERIFY_CONTEXT *,
658         CK_BYTE *,
659         CK_ULONG,
660         CK_BYTE *,
661         CK_ULONG);

663 CK_RV rsa_pkcs_verify_recover(SESSION *,
664         CK_BBOOL,
665         SIGN_VERIFY_CONTEXT *,
666         CK_BYTE *,
667         CK_ULONG,
668         CK_BYTE *,
669         CK_ULONG *);

671 CK_RV rsa_hash_pkcs_sign(SESSION *,
672         CK_BBOOL,
673         SIGN_VERIFY_CONTEXT *,
674         CK_BYTE *,
675         CK_ULONG,
676         CK_BYTE *,
677         CK_ULONG *);

679 CK_RV rsa_hash_pkcs_verify(SESSION *,
```



```
680         SIGN_VERIFY_CONTEXT *,
681         CK_BYTE *,
682         CK_ULONG,
683         CK_BYTE *,
684         CK_ULONG);

686 CK_RV rsa_hash_pkcs_sign_update(SESSION *,
687         SIGN_VERIFY_CONTEXT *,
688         CK_BYTE *,
689         CK_ULONG);

691 CK_RV rsa_hash_pkcs_verify_update(SESSION *,
692         SIGN_VERIFY_CONTEXT *,
693         CK_BYTE *,
694         CK_ULONG);

696 CK_RV rsa_hash_pkcs_sign_final(SESSION *,
697         CK_BBOOL,
698         SIGN_VERIFY_CONTEXT *,
699         CK_BYTE *,
700         CK_ULONG *);

702 CK_RV rsa_hash_pkcs_verify_final(SESSION *,
703         SIGN_VERIFY_CONTEXT *,
704         CK_BYTE *,
705         CK_ULONG);


708 CK_RV ckm_rsa_key_pair_gen(TSS_HCONTEXT, TEMPLATE *, TEMPLATE *);

710 CK_RV sha1_hash(SESSION *, CK_BBOOL,
711         DIGEST_CONTEXT *,
712         CK_BYTE *, CK_ULONG,
713         CK_BYTE *, CK_ULONG *);

715 CK_RV sha1_hmac_sign(SESSION *, CK_BBOOL,
716         SIGN_VERIFY_CONTEXT *,
717         CK_BYTE *,
718         CK_ULONG,
719         CK_BYTE *,
720         CK_ULONG *);

722 CK_RV sha1_hmac_verify(SESSION *,
723         SIGN_VERIFY_CONTEXT *,
724         CK_BYTE *,
725         CK_ULONG,
726         CK_BYTE *,
727         CK_ULONG);

729 CK_RV md5_hash(SESSION *, CK_BBOOL,
730         DIGEST_CONTEXT *,
731         CK_BYTE *, CK_ULONG,
732         CK_BYTE *, CK_ULONG *);

734 CK_RV md5_hmac_sign(SESSION *, CK_BBOOL,
735         SIGN_VERIFY_CONTEXT *,
736         CK_BYTE *,
737         CK_ULONG,
738         CK_BYTE *,
739         CK_ULONG *);

741 CK_RV md5_hmac_verify(SESSION *,
742         SIGN_VERIFY_CONTEXT *,
743         CK_BYTE *,
744         CK_ULONG,
745         CK_BYTE *,
```

```
746         CK_ULONG);

748 DL_NODE *dlist_add_as_first(DL_NODE *, void *);
749 DL_NODE *dlist_add_as_last(DL_NODE *, void *);
750 DL_NODE *dlist_find(DL_NODE *, void *);
751 DL_NODE *dlist_get_first(DL_NODE *);
752 DL_NODE *dlist_get_last(DL_NODE *);
753         CK_ULONG dlist_length(DL_NODE *);
754 DL_NODE *dlist_next(DL_NODE *);
755 DL_NODE *dlist_prev(DL_NODE *);
756 void dlist_purge(DL_NODE *);
757 DL_NODE *dlist_remove_node(DL_NODE *, DL_NODE *);

759 CK_RV attach_shm(void);
760 CK_RV detach_shm(void);

762 // encryption manager routines
763 //
764 CK_RV encr_mgr_init(SESSION *,
765         ENCR_DECR_CONTEXT *,
766         CK_ULONG,
767         CK_MECHANISM *,
768         CK_OBJECT_HANDLE);

770 CK_RV encr_mgr_cleanup(ENCR_DECR_CONTEXT *);

772 CK_RV encr_mgr_encrypt(SESSION *, CK_BBOOL,
773         ENCR_DECR_CONTEXT *,
774         CK_BYTE *, CK_ULONG,
775         CK_BYTE *, CK_ULONG *);

777 CK_RV decr_mgr_init(SESSION *,
778         ENCR_DECR_CONTEXT *,
779         CK_ULONG,
780         CK_MECHANISM *,
781         CK_OBJECT_HANDLE);

783 CK_RV decr_mgr_cleanup(ENCR_DECR_CONTEXT *);

785 CK_RV decr_mgr_decrypt(SESSION *, CK_BBOOL,
786         ENCR_DECR_CONTEXT *,
787         CK_BYTE *, CK_ULONG,
788         CK_BYTE *, CK_ULONG *);

790 CK_RV digest_mgr_cleanup(DIGEST_CONTEXT *);

792 CK_RV digest_mgr_init(SESSION *,
793         DIGEST_CONTEXT *,
794         CK_MECHANISM *);

796 CK_RV digest_mgr_digest(SESSION *, CK_BBOOL,
797         DIGEST_CONTEXT *,
798         CK_BYTE *, CK_ULONG,
799         CK_BYTE *, CK_ULONG *);

801 CK_RV digest_mgr_digest_update(SESSION *,
802         DIGEST_CONTEXT *,
803         CK_BYTE *, CK_ULONG);

805 CK_RV digest_mgr_digest_key(SESSION *,
806         DIGEST_CONTEXT *,
807         CK_OBJECT_HANDLE);

809 CK_RV digest_mgr_digest_final(SESSION *,
810         DIGEST_CONTEXT *,
811         CK_BYTE *, CK_ULONG *);
```



```
813 CK_RV key_mgr_generate_key_pair(SESSION *,
814         CK_MECHANISM *,
815         CK_ATTRIBUTE *, CK_ULONG,
816         CK_ATTRIBUTE *, CK_ULONG,
817         CK_OBJECT_HANDLE *,
818         CK_OBJECT_HANDLE *);

820 CK_RV key_mgr_wrap_key(SESSION *,
821         CK_BBOOL,
822         CK_MECHANISM *,
823         CK_OBJECT_HANDLE,
824         CK_OBJECT_HANDLE,
825         CK_BYTE *,
826         CK_ULONG *);

828 CK_RV key_mgr_unwrap_key(SESSION *,
829         CK_MECHANISM *,
830         CK_ATTRIBUTE *,
831         CK_ULONG,
832         CK_BYTE *,
833         CK_ULONG,
834         CK_OBJECT_HANDLE,
835         CK_OBJECT_HANDLE *);

837 CK_RV sign_mgr_init(SESSION *,
838         SIGN_VERIFY_CONTEXT *,
839         CK_MECHANISM *,
840         CK_BBOOL,
841         CK_OBJECT_HANDLE);

843 CK_RV sign_mgr_cleanup(SIGN_VERIFY_CONTEXT *);

845 CK_RV sign_mgr_sign(SESSION *,
846         CK_BBOOL,
847         SIGN_VERIFY_CONTEXT *,
848         CK_BYTE *,
849         CK_ULONG,
850         CK_BYTE *,
851         CK_ULONG *);

853 CK_RV sign_mgr_sign_recover(SESSION *,
854         CK_BBOOL,
855         SIGN_VERIFY_CONTEXT *,
856         CK_BYTE *,
857         CK_ULONG,
858         CK_BYTE *,
859         CK_ULONG *);

861 CK_RV sign_mgr_sign_final(SESSION *,
862         CK_BBOOL,
863         SIGN_VERIFY_CONTEXT *,
864         CK_BYTE *,
865         CK_ULONG *);

867 CK_RV sign_mgr_sign_update(SESSION *,
868         SIGN_VERIFY_CONTEXT *,
869         CK_BYTE *,
870         CK_ULONG);

872 CK_RV verify_mgr_init(SESSION *,
873         SIGN_VERIFY_CONTEXT *,
874         CK_MECHANISM *,
875         CK_BBOOL,
876         CK_OBJECT_HANDLE);
```

```
878 CK_RV verify_mgr_cleanup(SIGN_VERIFY_CONTEXT *);

880 CK_RV verify_mgr_verify(SESSION *,
881         SIGN_VERIFY_CONTEXT *,
882         CK_BYTE *,
883         CK_ULONG,
884         CK_BYTE *,
885         CK_ULONG);

887 CK_RV verify_mgr_verify_recover(SESSION *,
888         CK_BBOOL,
889         SIGN_VERIFY_CONTEXT *,
890         CK_BYTE *,
891         CK_ULONG,
892         CK_BYTE *,
893         CK_ULONG *);

895 CK_RV verify_mgr_verify_update(SESSION *,
896         SIGN_VERIFY_CONTEXT *,
897         CK_BYTE *,
898         CK_ULONG);

900 CK_RV verify_mgr_verify_final(SESSION *,
901         SIGN_VERIFY_CONTEXT *,
902         CK_BYTE *,
903         CK_ULONG);


906 // session manager routines
907 //
908 CK_RV session_mgr_close_all_sessions(void);
909 CK_RV session_mgr_close_session(SESSION *);
910 SESSION *session_mgr_find(CK_SESSION_HANDLE);
911 CK_RV session_mgr_login_all(CK_USER_TYPE);
912 CK_RV session_mgr_logout_all(void);
913 CK_RV session_mgr_new(CK_ULONG, SESSION **);

915 CK_BBOOL session_mgr_readonly_exists(void);
916 CK_BBOOL session_mgr_so_session_exists(void);
917 CK_BBOOL session_mgr_user_session_exists(void);
918 CK_BBOOL session_mgr_public_session_exists(void);

920 CK_RV session_mgr_get_op_state(SESSION *, CK_BBOOL,
921         CK_BYTE *, CK_ULONG *);

923 CK_RV session_mgr_set_op_state(SESSION *,
924         CK_OBJECT_HANDLE, CK_OBJECT_HANDLE, CK_BYTE *);

926 CK_RV object_mgr_add(SESSION *,
927         CK_ATTRIBUTE *, CK_ULONG, CK_OBJECT_HANDLE *);

929 CK_RV object_mgr_add_to_map(SESSION *, OBJECT *, CK_OBJECT_HANDLE *);

931 CK_RV object_mgr_add_to_shm(OBJECT *);
932 CK_RV object_mgr_del_from_shm(OBJECT *);

934 CK_RV object_mgr_copy(SESSION *,
935         CK_ATTRIBUTE *, CK_ULONG, CK_OBJECT_HANDLE,
936         CK_OBJECT_HANDLE *);

938 CK_RV object_mgr_create_final(SESSION *,
939         OBJECT *, CK_OBJECT_HANDLE *);

941 CK_RV object_mgr_create_skel(SESSION *,
942         CK_ATTRIBUTE *, CK_ULONG, CK_ULONG,
943         CK_ULONG, CK_ULONG, OBJECT **);
```



```
945 CK_RV object_mgr_destroy_object(SESSION *, CK_OBJECT_HANDLE);

947 CK_RV object_mgr_destroy_token_objects(TSS_HCONTEXT);

949 CK_RV object_mgr_find_in_map1(TSS_HCONTEXT, CK_OBJECT_HANDLE, OBJECT **);

951 CK_RV object_mgr_find_in_map2(TSS_HCONTEXT, OBJECT *, CK_OBJECT_HANDLE *);

953 CK_RV object_mgr_find_init(SESSION *, CK_ATTRIBUTE *, CK_ULONG);

955 CK_RV object_mgr_find_build_list(SESSION *,
956         CK_ATTRIBUTE *,
957         CK_ULONG,
958         DL_NODE *,
959         CK_BBOOL public_only);

961 CK_RV object_mgr_find_final(SESSION *);

963 CK_RV object_mgr_get_attribute_values(SESSION *,
964         CK_OBJECT_HANDLE,
965         CK_ATTRIBUTE *,
966         CK_ULONG);

968 CK_RV object_mgr_get_object_size(TSS_HCONTEXT, CK_OBJECT_HANDLE,
969         CK_ULONG *);

971 CK_BBOOL object_mgr_invalidate_handle1(CK_OBJECT_HANDLE handle);

973 CK_BBOOL object_mgr_invalidate_handle2(OBJECT *);

975 CK_BBOOL object_mgr_purge_session_objects(SESSION *, SESS_OBJ_TYPE);

977 CK_BBOOL object_mgr_purge_token_objects(TSS_HCONTEXT);

979 CK_BBOOL object_mgr_purge_private_token_objects(TSS_HCONTEXT);

981 CK_RV object_mgr_remove_from_map(CK_OBJECT_HANDLE);

983 CK_RV object_mgr_restore_obj(CK_BYTE *, OBJECT *);

985 CK_RV object_mgr_set_attribute_values(SESSION *,
986         CK_OBJECT_HANDLE,
987         CK_ATTRIBUTE *,
988         CK_ULONG);

990 CK_BBOOL object_mgr_purge_map(SESSION *, SESS_OBJ_TYPE);

992 CK_RV object_create(CK_ATTRIBUTE *, CK_ULONG, OBJECT **);

994 CK_RV object_create_skel(CK_ATTRIBUTE *,
995         CK_ULONG,
996         CK_ULONG,
997         CK_ULONG,
998         CK_ULONG,
999         OBJECT **);

1001 CK_RV object_copy(CK_ATTRIBUTE *,
1002         CK_ULONG,
1003         OBJECT *,
1004         OBJECT **);

1006 CK_RV object_flatten(OBJECT *,
1007         CK_BYTE **,
1008         CK_ULONG_32 *);
```

```
1010 CK_BBOOL object_free(OBJECT *);

1012 CK_RV object_get_attribute_values(OBJECT *,
1013         CK_ATTRIBUTE *,
1014         CK_ULONG);

1016 CK_ULONG object_get_size(OBJECT *);

1018 CK_RV object_restore(CK_BYTE *,
1019         OBJECT **,
1020         CK_BBOOL replace);

1022 CK_RV object_set_attribute_values(OBJECT *,
1023         CK_ATTRIBUTE *,
1024         CK_ULONG);

1026 CK_BBOOL object_is_modifiable(OBJECT *);
1027 CK_BBOOL object_is_private(OBJECT *);
1028 CK_BBOOL object_is_public(OBJECT *);
1029 CK_BBOOL object_is_token_object(OBJECT *);
1030 CK_BBOOL object_is_session_object(OBJECT *);

1032 CK_BBOOL is_attribute_defined(CK_ATTRIBUTE_TYPE);

1034 CK_RV template_add_attributes(TEMPLATE *,
1035         CK_ATTRIBUTE *, CK_ULONG);

1037 CK_RV template_add_default_attributes(TEMPLATE *,
1038         CK_ULONG,
1039         CK_ULONG,
1040         CK_ULONG);

1042 CK_BBOOL template_attribute_find(TEMPLATE *,
1043         CK_ATTRIBUTE_TYPE, CK_ATTRIBUTE **);

1045 void template_attribute_find_multiple(TEMPLATE *,
1046         ATTRIBUTE_PARSE_LIST *,
1047         CK_ULONG);

1049 CK_BBOOL template_check_exportability(TEMPLATE *, CK_ATTRIBUTE_TYPE type);

1051 CK_RV template_check_required_attributes(TEMPLATE *,
1052         CK_ULONG, CK_ULONG, CK_ULONG);

1054 CK_RV template_check_required_base_attributes(TEMPLATE *,
1055         CK_ULONG);

1057 CK_BBOOL template_compare(CK_ATTRIBUTE *,
1058         CK_ULONG, TEMPLATE *);

1060 CK_RV template_copy(TEMPLATE *, TEMPLATE *);

1062 CK_RV template_flatten(TEMPLATE *, CK_BYTE *);

1064 CK_RV template_free(TEMPLATE *);

1066 CK_BBOOL template_get_class(TEMPLATE *, CK_ULONG *, CK_ULONG *);

1068 CK_ULONG template_get_count(TEMPLATE *);

1070 CK_ULONG template_get_size(TEMPLATE *);
1071 CK_ULONG template_get_compressed_size(TEMPLATE *);

1073 CK_RV template_set_default_common_attributes(TEMPLATE *);

1075 CK_RV template_merge(TEMPLATE *, TEMPLATE **);
```



```
1077 CK_RV template_update_attribute(TEMPLATE *, CK_ATTRIBUTE *);

1079 CK_RV template_unflatten(TEMPLATE **, CK_BYTE *, CK_ULONG);

1081 CK_RV template_validate_attribute(TEMPLATE *,
1082         CK_ATTRIBUTE *, CK_ULONG, CK_ULONG, CK_ULONG);

1084 CK_RV template_validate_attributes(TEMPLATE *,
1085         CK_ULONG, CK_ULONG, CK_ULONG);

1087 CK_RV template_validate_base_attribute(TEMPLATE *,
1088         CK_ATTRIBUTE *, CK_ULONG);


1091 // DATA OBJECT ROUTINES
1092 //
1093 CK_RV data_object_check_required_attributes(TEMPLATE *, CK_ULONG);
1094 CK_RV data_object_set_default_attributes(TEMPLATE *, CK_ULONG);
1095 CK_RV data_object_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

1097 // CERTIFICATE ROUTINES
1098 CK_RV cert_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

1100 CK_RV cert_x509_check_required_attributes(TEMPLATE *, CK_ULONG);
1101 CK_RV cert_x509_set_default_attributes(TEMPLATE *, CK_ULONG);
1102 CK_RV cert_x509_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);
1103 CK_RV cert_vendor_check_required_attributes(TEMPLATE *, CK_ULONG);
1104 CK_RV cert_vendor_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

1106 //
1107 // KEY ROUTINES
1108 //
1109 CK_RV key_object_check_required_attributes(TEMPLATE *, CK_ULONG);
1110 CK_RV key_object_set_default_attributes(TEMPLATE *, CK_ULONG);
1111 CK_RV key_object_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

1113 CK_RV publ_key_check_required_attributes(TEMPLATE *, CK_ULONG);
1114 CK_RV publ_key_set_default_attributes(TEMPLATE *, CK_ULONG);
1115 CK_RV publ_key_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

1117 CK_RV priv_key_check_required_attributes(TEMPLATE *, CK_ULONG);
1118 CK_RV priv_key_set_default_attributes(TEMPLATE *, CK_ULONG);
1119 CK_RV priv_key_unwrap(TEMPLATE *, CK_ULONG, CK_BYTE *, CK_ULONG);
1120 CK_RV priv_key_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);

1122         CK_BBOOL secret_key_check_exportability(CK_ATTRIBUTE_TYPE type);
1123 CK_RV secret_key_check_required_attributes(TEMPLATE *, CK_ULONG);
1124 CK_RV secret_key_set_default_attributes(TEMPLATE *, CK_ULONG);
1125 CK_RV secret_key_unwrap(TEMPLATE *, CK_ULONG, CK_BYTE *, CK_ULONG,
1126         CK_BBOOL fromend);
1127 CK_RV secret_key_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *,
1128         CK_ULONG);

1130 // rsa routines
1131 //
1132 CK_RV rsa_publ_check_required_attributes(TEMPLATE *, CK_ULONG);
1133 CK_RV rsa_publ_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);
1134 CK_RV rsa_publ_set_default_attributes(TEMPLATE *, CK_ULONG);
1135         CK_BBOOL rsa_priv_check_exportability(CK_ATTRIBUTE_TYPE type);
1136 CK_RV rsa_priv_check_required_attributes(TEMPLATE *, CK_ULONG);
1137 CK_RV rsa_priv_set_default_attributes(TEMPLATE *, CK_ULONG);
1138 CK_RV rsa_priv_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);
1139 CK_RV rsa_priv_wrap_get_data(TEMPLATE *, CK_BBOOL, CK_BYTE **, CK_ULONG *);
1140 CK_RV rsa_priv_unwrap(TEMPLATE *, CK_BYTE *, CK_ULONG);
```

```
1142 // Generic secret key routines
1143 CK_RV generic_secret_check_required_attributes(TEMPLATE *, CK_ULONG);
1144 CK_RV generic_secret_set_default_attributes(TEMPLATE *, CK_ULONG);
1145 CK_RV generic_secret_validate_attribute(TEMPLATE *, CK_ATTRIBUTE *, CK_ULONG);
1146 CK_RV generic_secret_wrap_get_data(TEMPLATE *, CK_BBOOL,
1147         CK_BYTE **, CK_ULONG *);

1149 CK_RV generic_secret_unwrap(TEMPLATE *, CK_BYTE *, CK_ULONG, CK_BBOOL fromend);

1151 CK_RV tpm_encrypt_data(TSS_HCONTEXT,
1152         TSS_HKEY, CK_BYTE *, CK_ULONG, CK_BYTE *, CK_ULONG *);

1154 CK_RV tpm_decrypt_data(TSS_HCONTEXT,
1155         TSS_HKEY, CK_BYTE *, CK_ULONG, CK_BYTE *, CK_ULONG *);

1157 CK_ULONG ber_encode_INTEGER(CK_BBOOL,
1158         CK_BYTE **, CK_ULONG *, CK_BYTE *, CK_ULONG);

1160 CK_RV ber_decode_INTEGER(CK_BYTE *,
1161         CK_BYTE **, CK_ULONG *, CK_ULONG *);

1163 CK_RV ber_encode_OCTET_STRING(CK_BBOOL,
1164         CK_BYTE **, CK_ULONG *, CK_BYTE *, CK_ULONG);

1166 CK_RV ber_decode_OCTET_STRING(CK_BYTE *,
1167         CK_BYTE **, CK_ULONG *, CK_ULONG *);

1169 CK_RV ber_encode_SEQUENCE(CK_BBOOL,
1170         CK_BYTE **, CK_ULONG *, CK_BYTE *, CK_ULONG);

1172 CK_RV ber_decode_SEQUENCE(CK_BYTE *,
1173         CK_BYTE **, CK_ULONG *, CK_ULONG *);

1175 CK_RV ber_encode_PrivateKeyInfo(CK_BBOOL,
1176         CK_BYTE **, CK_ULONG *, CK_BYTE *,
1177         CK_ULONG, CK_BYTE *, CK_ULONG);

1179 CK_RV ber_decode_PrivateKeyInfo(CK_BYTE *,
1180         CK_ULONG, CK_BYTE **, CK_ULONG *, CK_BYTE **);

1182 CK_RV ber_encode_RSAPrivateKey(CK_BBOOL,
1183         CK_BYTE **, CK_ULONG *, CK_ATTRIBUTE *,
1184         CK_ATTRIBUTE *, CK_ATTRIBUTE *, CK_ATTRIBUTE *,
1185         CK_ATTRIBUTE *, CK_ATTRIBUTE *, CK_ATTRIBUTE *,
1186         CK_ATTRIBUTE *);

1188 CK_RV ber_decode_RSAPrivateKey(CK_BYTE *,
1189         CK_ULONG, CK_ATTRIBUTE **, CK_ATTRIBUTE **,
1190         CK_ATTRIBUTE **, CK_ATTRIBUTE **, CK_ATTRIBUTE **,
1191         CK_ATTRIBUTE **, CK_ATTRIBUTE **, CK_ATTRIBUTE **);


1194 CK_RV ber_encode_DSAPrivateKey(CK_BBOOL,
1195         CK_BYTE **, CK_ULONG *, CK_ATTRIBUTE *,
1196         CK_ATTRIBUTE *, CK_ATTRIBUTE *, CK_ATTRIBUTE *);

1198 CK_RV ber_decode_DSAPrivateKey(CK_BYTE *,
1199         CK_ULONG, CK_ATTRIBUTE **, CK_ATTRIBUTE **,
1200         CK_ATTRIBUTE **, CK_ATTRIBUTE **);

1202 #define APPID   "TPM_STDLL"

1204 /* log to stdout */
1205 #define LogMessage(dest, priority, layer, fmt, ...) \
1206         (void) fprintf(dest, "%s %s %s:%d " fmt "\n", (char *)priority, \
1207                 (char *)layer, (char *)__FILE__,\
```



```
1208                 (int)__LINE__, __VA_ARGS__);

1210 #define LogMessage1(dest, priority, layer, data) \
1211         (void) fprintf(dest, "%s %s %s:%d %s\n", priority, layer, __FILE__, \
1212         __LINE__, data);

1214 /* Debug logging */
1215 #ifdef DEBUG
1216 #define LogDebug(fmt, ...) LogMessage(stdout, "LOG_DEBUG", APPID, \
1217         fmt, __VA_ARGS__)

1219 #define LogDebug1(data) LogMessage1(stdout, "LOG_DEBUG", APPID, data)

1221 /* Error logging */
1222 #define LogError(fmt, ...) LogMessage(stderr, "LOG_ERR", APPID,\
1223         "ERROR: " fmt, __VA_ARGS__)

1225 #define LogError1(data) LogMessage1(stderr, "LOG_ERR", APPID,\
1226         "ERROR: " data)

1228 /* Warn logging */
1229 #define LogWarn(fmt, ...) LogMessage(stdout, "LOG_WARNING", APPID,\
1230         "WARNING: " fmt, __VA_ARGS__)

1232 #define LogWarn1(data) LogMessage1(stdout, "LOG_WARNING", APPID,\
1233         "WARNING: " data)

1235 /* Info Logging */
1236 #define LogInfo(fmt, ...) LogMessage(stdout, "LOG_INFO", APPID,\
1237         fmt, __VA_ARGS__)

1239 #define LogInfo1(data) LogMessage1(stdout, "LOG_INFO", APPID, data)

1241 #define st_err_log(...) LogMessage(stderr, "ST MSG", APPID,\
1242         "", __VA_ARGS__)
1243 #else
1244 #define LogDebug(...)
1245 #define LogDebug1(...)
1246 #define LogBlob(...)
1247 #define LogError(...)
1248 #define LogError1(...)
1249 #define LogWarn(...)
1250 #define LogWarn1(...)
1251 #define LogInfo(...)
1252 #define LogInfo1(...)
1253 #define st_err_log(...)
1254 #endif

1256 /*
1257  * CK_FUNCTION_LIST is a structure holding a Cryptoki spec
1258  * version and pointers of appropriate types to all the
1259  * Cryptoki functions
1260  */

1262 /* CK_FUNCTION_LIST is new for v2.0 */

1264 typedef CK_RV
1265         (CK_PTR ST_C_Initialize)
1266         (void *ppFunctionList, CK_SLOT_ID slotID, CK_CHAR_PTR pCorrelator);
1267 typedef CK_RV
1268         (CK_PTR  ST_C_Finalize)
1269         (CK_VOID_PTR pReserved);
1270 typedef CK_RV
1271         (CK_PTR  ST_C_Terminate)();
1272 typedef CK_RV
1273         (CK_PTR  ST_C_GetInfo)
```

```
1274         (CK_INFO_PTR pInfo);
1275 typedef CK_RV
1276         (CK_PTR  ST_C_GetFunctionList)
1277         (CK_FUNCTION_LIST_PTR_PTR ppFunctionList);
1278 typedef CK_RV
1279         (CK_PTR  ST_C_GetSlotList)
1280         (CK_BBOOL tokenPresent, CK_SLOT_ID_PTR pSlotList,
1281         CK_ULONG_PTR pusCount);
1282 typedef CK_RV
1283         (CK_PTR  ST_C_GetSlotInfo)
1284         (CK_SLOT_ID slotID, CK_SLOT_INFO_PTR pInfo);
1285 typedef CK_RV
1286         (CK_PTR  ST_C_GetTokenInfo)
1287         (CK_SLOT_ID slotID, CK_TOKEN_INFO_PTR pInfo);
1288 typedef CK_RV
1289         (CK_PTR  ST_C_GetMechanismList)
1290         (CK_SLOT_ID slotID, CK_MECHANISM_TYPE_PTR pMechanismList,
1291         CK_ULONG_PTR pusCount);
1292 typedef CK_RV
1293         (CK_PTR  ST_C_GetMechanismInfo)
1294         (CK_SLOT_ID slotID, CK_MECHANISM_TYPE type,
1295         CK_MECHANISM_INFO_PTR pInfo);
1296 typedef CK_RV
1297         (CK_PTR  ST_C_InitToken)
1298         (CK_SLOT_ID slotID, CK_CHAR_PTR pPin, CK_ULONG usPinLen,
1299         CK_CHAR_PTR pLabel);
1300 typedef CK_RV
1301         (CK_PTR  ST_C_InitPIN)
1302         (ST_SESSION_T hSession, CK_CHAR_PTR pPin,
1303         CK_ULONG usPinLen);
1304 typedef CK_RV
1305         (CK_PTR  ST_C_SetPIN)
1306         (ST_SESSION_T hSession, CK_CHAR_PTR pOldPin,
1307         CK_ULONG usOldLen, CK_CHAR_PTR pNewPin,
1308         CK_ULONG usNewLen);

1310 typedef CK_RV
1311         (CK_PTR  ST_C_OpenSession)
1312         (CK_SLOT_ID slotID, CK_FLAGS flags,
1313         CK_SESSION_HANDLE_PTR phSession);

1315 typedef CK_RV
1316         (CK_PTR  ST_C_CloseSession)
1317         (ST_SESSION_T hSession);
1318 typedef CK_RV
1319         (CK_PTR  ST_C_CloseAllSessions)
1320         (CK_SLOT_ID slotID);
1321 typedef CK_RV
1322         (CK_PTR  ST_C_GetSessionInfo)
1323         (ST_SESSION_T hSession, CK_SESSION_INFO_PTR pInfo);
1324 typedef CK_RV
1325         (CK_PTR  ST_C_GetOperationState)
1326         (ST_SESSION_T hSession, CK_BYTE_PTR pOperationState,
1327         CK_ULONG_PTR pulOperationStateLen);
1328 typedef CK_RV
1329         (CK_PTR  ST_C_SetOperationState)
1330         (ST_SESSION_T hSession, CK_BYTE_PTR pOperationState,
1331         CK_ULONG ulOperationStateLen,
1332         CK_OBJECT_HANDLE hEncryptionKey,
1333         CK_OBJECT_HANDLE hAuthenticationKey);
1334 typedef CK_RV
1335         (CK_PTR  ST_C_Login)(ST_SESSION_T hSession,
1336         CK_USER_TYPE userType, CK_CHAR_PTR pPin,
1337         CK_ULONG usPinLen);
1338 typedef CK_RV
1339         (CK_PTR  ST_C_Logout)(ST_SESSION_T hSession);
```



```
1340 typedef CK_RV
1341         (CK_PTR  ST_C_CreateObject)
1342         (ST_SESSION_T hSession, CK_ATTRIBUTE_PTR pTemplate,
1343         CK_ULONG usCount, CK_OBJECT_HANDLE_PTR phObject);

1345 typedef CK_RV
1346         (CK_PTR  ST_C_CopyObject)
1347         (ST_SESSION_T hSession, CK_OBJECT_HANDLE hObject,
1348         CK_ATTRIBUTE_PTR pTemplate, CK_ULONG usCount,
1349         CK_OBJECT_HANDLE_PTR phNewObject);
1350 typedef CK_RV
1351         (CK_PTR  ST_C_DestroyObject)
1352         (ST_SESSION_T hSession, CK_OBJECT_HANDLE hObject);
1353 typedef CK_RV
1354         (CK_PTR  ST_C_GetObjectSize)
1355         (ST_SESSION_T hSession, CK_OBJECT_HANDLE hObject,
1356         CK_ULONG_PTR pusSize);
1357 typedef CK_RV
1358         (CK_PTR  ST_C_GetAttributeValue)
1359         (ST_SESSION_T hSession, CK_OBJECT_HANDLE hObject,
1360         CK_ATTRIBUTE_PTR pTemplate, CK_ULONG usCount);
1361 typedef CK_RV
1362         (CK_PTR  ST_C_SetAttributeValue)
1363         (ST_SESSION_T hSession, CK_OBJECT_HANDLE hObject,
1364         CK_ATTRIBUTE_PTR pTemplate, CK_ULONG usCount);
1365 typedef CK_RV
1366         (CK_PTR  ST_C_FindObjectsInit)
1367         (ST_SESSION_T hSession, CK_ATTRIBUTE_PTR pTemplate,
1368         CK_ULONG usCount);
1369 typedef CK_RV
1370         (CK_PTR  ST_C_FindObjects)
1371         (ST_SESSION_T hSession,
1372         CK_OBJECT_HANDLE_PTR phObject, CK_ULONG usMaxObjectCount,
1373         CK_ULONG_PTR pusObjectCount);
1374 typedef CK_RV
1375         (CK_PTR  ST_C_FindObjectsFinal)
1376         (ST_SESSION_T hSession);
1377 typedef CK_RV
1378         (CK_PTR  ST_C_EncryptInit)
1379         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1380         CK_OBJECT_HANDLE hKey);
1381 typedef CK_RV
1382         (CK_PTR  ST_C_Encrypt)
1383         (ST_SESSION_T hSession, CK_BYTE_PTR pData,
1384         CK_ULONG usDataLen, CK_BYTE_PTR pEncryptedData,
1385         CK_ULONG_PTR pusEncryptedDataLen);
1386 typedef CK_RV
1387         (CK_PTR  ST_C_EncryptUpdate)
1388         (ST_SESSION_T hSession, CK_BYTE_PTR pPart,
1389         CK_ULONG usPartLen, CK_BYTE_PTR pEncryptedPart,
1390         CK_ULONG_PTR pusEncryptedPartLen);
1391 typedef CK_RV
1392         (CK_PTR  ST_C_EncryptFinal)
1393         (ST_SESSION_T hSession,
1394         CK_BYTE_PTR pLastEncryptedPart,
1395         CK_ULONG_PTR pusLastEncryptedPartLen);
1396 typedef CK_RV
1397         (CK_PTR  ST_C_DecryptInit)
1398         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1399         CK_OBJECT_HANDLE hKey);
1400 typedef CK_RV
1401         (CK_PTR  ST_C_Decrypt)
1402         (ST_SESSION_T hSession, CK_BYTE_PTR pEncryptedData,
1403         CK_ULONG usEncryptedDataLen, CK_BYTE_PTR pData,
1404         CK_ULONG_PTR pusDataLen);
1405 typedef CK_RV
```

```
1406         (CK_PTR  ST_C_DecryptUpdate)
1407         (ST_SESSION_T hSession, CK_BYTE_PTR pEncryptedPart,
1408         CK_ULONG usEncryptedPartLen, CK_BYTE_PTR pPart,
1409         CK_ULONG_PTR pusPartLen);
1410 typedef CK_RV
1411         (CK_PTR  ST_C_DecryptFinal)
1412         (ST_SESSION_T hSession, CK_BYTE_PTR pLastPart,
1413         CK_ULONG_PTR pusLastPartLen);
1414 typedef CK_RV
1415         (CK_PTR  ST_C_DigestInit)
1416         (ST_SESSION_T hSession,
1417         CK_MECHANISM_PTR pMechanism);
1418 typedef CK_RV
1419         (CK_PTR  ST_C_Digest)
1420         (ST_SESSION_T hSession, CK_BYTE_PTR pData,
1421         CK_ULONG usDataLen, CK_BYTE_PTR pDigest,
1422         CK_ULONG_PTR pusDigestLen);
1423 typedef CK_RV
1424         (CK_PTR  ST_C_DigestUpdate)
1425         (ST_SESSION_T hSession, CK_BYTE_PTR pPart,
1426         CK_ULONG usPartLen);
1427 typedef CK_RV
1428         (CK_PTR  ST_C_DigestKey)
1429         (ST_SESSION_T hSession, CK_OBJECT_HANDLE hKey);
1430 typedef CK_RV
1431         (CK_PTR  ST_C_DigestFinal)
1432         (ST_SESSION_T hSession, CK_BYTE_PTR pDigest,
1433         CK_ULONG_PTR pusDigestLen);
1434 typedef CK_RV
1435         (CK_PTR  ST_C_SignInit)
1436         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1437         CK_OBJECT_HANDLE hKey);
1438 typedef CK_RV
1439         (CK_PTR  ST_C_Sign)
1440         (ST_SESSION_T hSession, CK_BYTE_PTR pData,
1441         CK_ULONG usDataLen, CK_BYTE_PTR pSignature,
1442         CK_ULONG_PTR pusSignatureLen);
1443 typedef CK_RV
1444         (CK_PTR  ST_C_SignUpdate)
1445         (ST_SESSION_T hSession, CK_BYTE_PTR pPart,
1446         CK_ULONG usPartLen);
1447 typedef CK_RV
1448         (CK_PTR  ST_C_SignFinal)
1449         (ST_SESSION_T hSession, CK_BYTE_PTR pSignature,
1450         CK_ULONG_PTR pusSignatureLen);
1451 typedef CK_RV
1452         (CK_PTR  ST_C_SignRecoverInit)
1453         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1454         CK_OBJECT_HANDLE hKey);
1455 typedef CK_RV
1456         (CK_PTR  ST_C_SignRecover)
1457         (ST_SESSION_T hSession, CK_BYTE_PTR pData,
1458         CK_ULONG usDataLen, CK_BYTE_PTR pSignature,
1459         CK_ULONG_PTR pusSignatureLen);
1460 typedef CK_RV
1461         (CK_PTR  ST_C_VerifyInit)
1462         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1463         CK_OBJECT_HANDLE hKey);
1464 typedef CK_RV
1465         (CK_PTR  ST_C_Verify)
1466         (ST_SESSION_T hSession, CK_BYTE_PTR pData,
1467         CK_ULONG usDataLen, CK_BYTE_PTR pSignature,
1468         CK_ULONG usSignatureLen);
1469 typedef CK_RV
1470         (CK_PTR  ST_C_VerifyUpdate)
1471         (ST_SESSION_T hSession, CK_BYTE_PTR pPart,
```



```
1472         CK_ULONG usPartLen);
1473 typedef CK_RV
1474         (CK_PTR  ST_C_VerifyFinal)
1475         (ST_SESSION_T hSession, CK_BYTE_PTR pSignature,
1476         CK_ULONG usSignatureLen);
1477 typedef CK_RV
1478         (CK_PTR  ST_C_VerifyRecoverInit)
1479         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1480         CK_OBJECT_HANDLE hKey);
1481 typedef CK_RV
1482         (CK_PTR  ST_C_VerifyRecover)
1483         (ST_SESSION_T hSession, CK_BYTE_PTR pSignature,
1484         CK_ULONG usSignatureLen, CK_BYTE_PTR pData,
1485         CK_ULONG_PTR pusDataLen);
1486 typedef CK_RV
1487         (CK_PTR  ST_C_DigestEncryptUpdate)
1488         (ST_SESSION_T hSession, CK_BYTE_PTR pPart,
1489         CK_ULONG ulPartLen, CK_BYTE_PTR pEncryptedPart,
1490         CK_ULONG_PTR pulEncryptedPartLen);
1491 typedef CK_RV
1492         (CK_PTR  ST_C_DecryptDigestUpdate)
1493         (ST_SESSION_T hSession, CK_BYTE_PTR pEncryptedPart,
1494         CK_ULONG ulEncryptedPartLen, CK_BYTE_PTR pPart,
1495         CK_ULONG_PTR pulPartLen);
1496 typedef CK_RV
1497         (CK_PTR  ST_C_SignEncryptUpdate)
1498         (ST_SESSION_T hSession, CK_BYTE_PTR pPart,
1499         CK_ULONG ulPartLen, CK_BYTE_PTR pEncryptedPart,
1500         CK_ULONG_PTR pulEncryptedPartLen);
1501 typedef CK_RV
1502         (CK_PTR  ST_C_DecryptVerifyUpdate)
1503         (ST_SESSION_T hSession, CK_BYTE_PTR pEncryptedPart,
1504         CK_ULONG ulEncryptedPartLen, CK_BYTE_PTR pPart,
1505         CK_ULONG_PTR pulPartLen);
1506 typedef CK_RV
1507         (CK_PTR  ST_C_GenerateKey)
1508         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1509         CK_ATTRIBUTE_PTR pTemplate, CK_ULONG usCount,
1510         CK_OBJECT_HANDLE_PTR phKey);
1511 typedef CK_RV
1512         (CK_PTR  ST_C_GenerateKeyPair)
1513         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1514         CK_ATTRIBUTE_PTR pPublicKeyTemplate,
1515         CK_ULONG usPublicKeyAttributeCount,
1516         CK_ATTRIBUTE_PTR pPrivateKeyTemplate,
1517         CK_ULONG usPrivateKeyAttributeCount,
1518         CK_OBJECT_HANDLE_PTR phPrivateKey,
1519         CK_OBJECT_HANDLE_PTR phPublicKey);
1520 typedef CK_RV
1521         (CK_PTR  ST_C_WrapKey)
1522         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1523         CK_OBJECT_HANDLE hWrappingKey, CK_OBJECT_HANDLE hKey,
1524         CK_BYTE_PTR pWrappedKey, CK_ULONG_PTR pusWrappedKeyLen);
1525 typedef CK_RV
1526         (CK_PTR  ST_C_UnwrapKey)
1527         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1528         CK_OBJECT_HANDLE hUnwrappingKey, CK_BYTE_PTR pWrappedKey,
1529         CK_ULONG usWrappedKeyLen, CK_ATTRIBUTE_PTR pTemplate,
1530         CK_ULONG usAttributeCount, CK_OBJECT_HANDLE_PTR phKey);
1531 typedef CK_RV
1532         (CK_PTR  ST_C_DeriveKey)
1533         (ST_SESSION_T hSession, CK_MECHANISM_PTR pMechanism,
1534         CK_OBJECT_HANDLE hBaseKey, CK_ATTRIBUTE_PTR pTemplate,
1535         CK_ULONG usAttributeCount, CK_OBJECT_HANDLE_PTR phKey);
1536 typedef CK_RV
1537         (CK_PTR  ST_C_SeedRandom)
```

```
1538         (ST_SESSION_T hSession, CK_BYTE_PTR pSeed,
1539         CK_ULONG usSeedLen);
1540 typedef CK_RV
1541         (CK_PTR  ST_C_GenerateRandom)
1542         (ST_SESSION_T hSession, CK_BYTE_PTR pRandomData,
1543         CK_ULONG usRandomLen);
1544 typedef CK_RV
1545         (CK_PTR  ST_C_GetFunctionStatus)
1546         (ST_SESSION_T hSession);
1547 typedef CK_RV
1548         (CK_PTR  ST_C_CancelFunction)
1549         (ST_SESSION_T hSession);
1550 typedef CK_RV
1551         (CK_PTR  ST_Notify)
1552         (ST_SESSION_T hSession, CK_NOTIFICATION event,
1553         CK_VOID_PTR pApplication);
1554 typedef CK_RV
1555         (CK_PTR  ST_C_WaitForSlotEvent)
1556         (CK_FLAGS flags, CK_SLOT_ID_PTR pSlot,
1557         CK_VOID_PTR pReserved);



1561 struct ST_FCN_LIST {

1563         ST_C_Initialize ST_Initialize;
1564         ST_C_Finalize ST_Finalize;

1566         ST_C_GetTokenInfo ST_GetTokenInfo;
1567         ST_C_GetMechanismList ST_GetMechanismList;
1568         ST_C_GetMechanismInfo ST_GetMechanismInfo;
1569         ST_C_InitToken ST_InitToken;
1570         ST_C_InitPIN ST_InitPIN;
1571         ST_C_SetPIN ST_SetPIN;

1573         ST_C_OpenSession ST_OpenSession;
1574         ST_C_CloseSession ST_CloseSession;
1575         ST_C_GetSessionInfo ST_GetSessionInfo;
1576         ST_C_GetOperationState ST_GetOperationState;
1577         ST_C_SetOperationState ST_SetOperationState;
1578         ST_C_Login ST_Login;
1579         ST_C_Logout ST_Logout;

1581         ST_C_CreateObject ST_CreateObject;
1582         ST_C_CopyObject ST_CopyObject;
1583         ST_C_DestroyObject ST_DestroyObject;
1584         ST_C_GetObjectSize ST_GetObjectSize;
1585         ST_C_GetAttributeValue ST_GetAttributeValue;
1586         ST_C_SetAttributeValue ST_SetAttributeValue;
1587         ST_C_FindObjectsInit ST_FindObjectsInit;
1588         ST_C_FindObjects ST_FindObjects;
1589         ST_C_FindObjectsFinal ST_FindObjectsFinal;


1592         ST_C_EncryptInit ST_EncryptInit;
1593         ST_C_Encrypt ST_Encrypt;
1594         ST_C_EncryptUpdate ST_EncryptUpdate;
1595         ST_C_EncryptFinal ST_EncryptFinal;
1596         ST_C_DecryptInit ST_DecryptInit;
1597         ST_C_Decrypt ST_Decrypt;
1598         ST_C_DecryptUpdate ST_DecryptUpdate;
1599         ST_C_DecryptFinal ST_DecryptFinal;
1600         ST_C_DigestInit ST_DigestInit;
1601         ST_C_Digest ST_Digest;
1602         ST_C_DigestUpdate ST_DigestUpdate;
1603         ST_C_DigestKey ST_DigestKey;
```



```
1604         ST_C_DigestFinal ST_DigestFinal;
1605         ST_C_SignInit ST_SignInit;
1606         ST_C_Sign ST_Sign;
1607         ST_C_SignUpdate ST_SignUpdate;
1608         ST_C_SignFinal ST_SignFinal;
1609         ST_C_SignRecoverInit ST_SignRecoverInit;
1610         ST_C_SignRecover ST_SignRecover;
1611         ST_C_VerifyInit ST_VerifyInit;
1612         ST_C_Verify ST_Verify;
1613         ST_C_VerifyUpdate ST_VerifyUpdate;
1614         ST_C_VerifyFinal ST_VerifyFinal;
1615         ST_C_VerifyRecoverInit ST_VerifyRecoverInit;
1616         ST_C_VerifyRecover ST_VerifyRecover;
1617         ST_C_DigestEncryptUpdate ST_DigestEncryptUpdate;
1618         ST_C_DecryptDigestUpdate ST_DecryptDigestUpdate;
1619         ST_C_SignEncryptUpdate ST_SignEncryptUpdate;
1620         ST_C_DecryptVerifyUpdate ST_DecryptVerifyUpdate;
1621         ST_C_GenerateKey ST_GenerateKey;
1622         ST_C_GenerateKeyPair ST_GenerateKeyPair;
1623         ST_C_WrapKey ST_WrapKey;
1624         ST_C_UnwrapKey ST_UnwrapKey;
1625         ST_C_DeriveKey ST_DeriveKey;
1626         ST_C_SeedRandom ST_SeedRandom;
1627         ST_C_GenerateRandom ST_GenerateRandom;
1628         ST_C_GetFunctionStatus ST_GetFunctionStatus;
1629         ST_C_CancelFunction ST_CancelFunction;
1630 };
```

*unchanged_portion_omitted_*