



PKCS #11 and Microsoft[®] CryptoAPI
Mechanisms for One-time Password Tokens

Tuesday, February 15, 2005

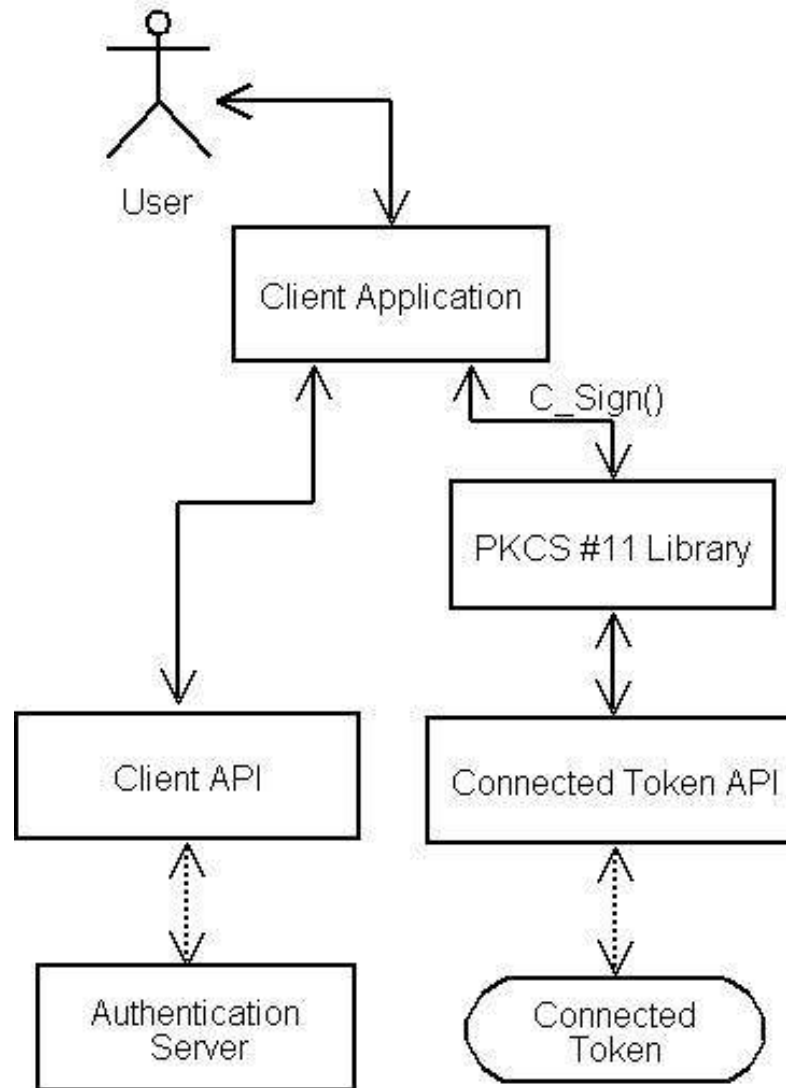
RSA Conference 2005

Document Objectives



- These two documents describe general PKCS #11 and CryptoAPI objects, procedures and mechanisms that can be used to retrieve and verify one-time passwords (OTPs) generated by OTP tokens
- Intended to meet the needs of vendors that wish to access connected OTP tokens in an interoperable manner
 - Eases the task of supporting multiple OTP token types
 - Enables a better user experience

PKCS #11 Principles of Operation



PKCS #11 OTP Key types



- OTP key type with a defined set of new, common, attributes
 - OTP Format (Hex, Decimal, ...)
 - OTP Length
 - PIN related: PIN Pad, Default PIN, ...
 - Challenge/Counter/Time-based
 - Service Name (Identifier)

PKCS #11 OTP Functions



- Retains existing v2.20 function set
- General approach is to use C_Sign and C_Verify
 - Follows existing PKCS #11 approach for HMAC algorithms

PKCS #11 OTP Mechanisms



- Defines three OTP mechanisms based on the foregoing
 - CKM_SECURID
 - CKM_SECURID_TRADITIONAL
 - CKM_SECURID_KEY_GEN
- Our hope is that definitions for other OTP algorithms will follow
- Defines additional key attributes for keys of type CKK_SECURID
- Mechanism parameters:

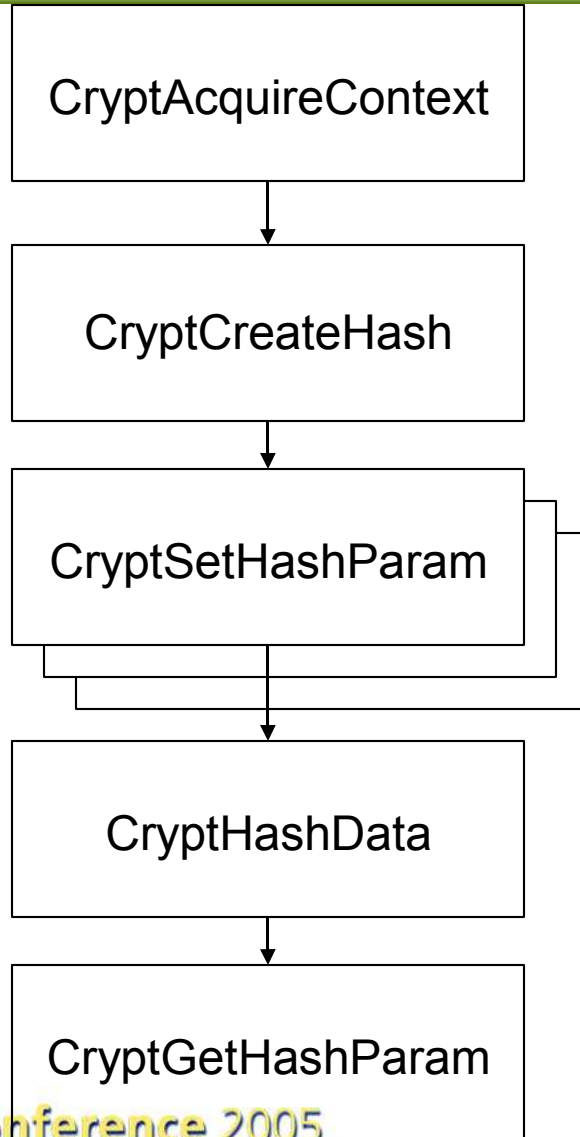
```
typedef struct CK_SECURID_PARAMS {
    CK_CHAR          utcTime[14];
    CK_FLAGS         flags;
    CK_UTF8CHAR_PTR pPIN;
    CK_UTF8CHAR_PTR pApplicationID;
    CK_VOID_PTR     pReserved} CK_SECURID_PARAMS;
```

SecurID OTP retrieval and validation



- Call C_SignInit with the mechanism parameters
- Provide a challenge (if any) in the call to C_Sign
- OTP returned as *pSignature*
- Validation with C_VerifyInit, C_Verify

CryptoAPI Principles of Operation



- CryptAcquireContext gives handle to key container
- CryptCreateHash gives handle to hash object, specifies OTP mechanism
- CryptSetHashParam sets various parameters
- CryptHashData generates the OTP
- CryptGetHashParam retrieves the OTP value
- Intent is to follow the existing HMAC approach for CryptoAPI

CryptoAPI Definitions



- Cryptographic Service Providers supporting the described mechanism are identified as being of type

PROV_OTP

- OTP algorithms will be of class

CALG_CLASS_OTP

- Keys will be created through CryptGenKey and CryptSetKeyParam, with parameters borrowed from the PKCS #11 sibling document

CryptoAPI OTP Algorithms



- Defines two OTP algorithms based on the foregoing
 - CALG_SECURID
 - CALG_SECURID_TRADITIONAL
- Our hope is that definitions for other OTP algorithms will follow
- For RSA SecurID, the document defines the parameters that may be used in calls to CryptSetHashParam
 - PIN may be set here, but it is preferable to set the CRYPT_USER_DATA flag in the call to CryptHashData