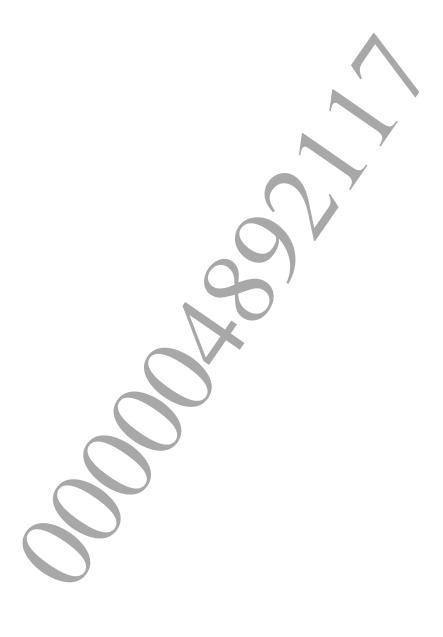


© 2011 Sony Computer Entertainment Inc. All Rights Reserved. SCE Confidential

### **Table of Contents**

1 Library Overview		3
		-
Rasic Usage Proce		1



# **Library Overview**

### Overview

libsha256 is a library that is used to generate a digest value using the SHA-256 Secure Hash Algorithm 256 format as defined by FIPS 180-2. It can be used to detect data corruption and prevent data tampering through the use of Keyed-Hashing for Message Authentication (HMAC).

### **Files**

The following files are required to use libsha256.

Description
Header file
Static link library file
Stub library file
weak import stub library file
PRX module file



## f 2 Using the Library

### **Basic Usage Procedure**

### (1) SHA-256 digest value computation (comprehensive)

No specific initialization is required to use libsha256.

```
SceUChar8 digest[SCE SHA256 DIGEST SIZE];
sceSha256Digest(plaintext, length, digest);
```

You can compute the digest value simply by calling the sceSha256Digest () function, as shown above.

#### (2) SHA-256 digest value computation (divided)

To compute a digest value for a large amount of data, the hash calculation can be broken up as shown below.

```
SceSha256Context sha;
SceUChar8 digest[SCE SHA256 DIGEST SIZE];
sceSha256BlockInit(&sha);
sceSha256BlockUpdate(&sha, plain1, len1);
sceSha256BlockUpdate(&sha, plain2, len2);
sceSha256BlockUpdate(&sha, plain3, len3);
                     Repeat an arbitrary number of times
sceSha256BlockResult(&sha, digest);
```

First, call the sceSha256BlockInit() function to initialize the SceSha256Context structure. Then, call the sceSha256BlockUpdate() function the desired number of times. Lastly, the digest value can be obtained by calling the sceSha256BlockResult () function.

