Table of Contents

BS2_SetOsdpStandardDeviceSecurityKey

[+ 2.9.1] `CoreStation40` Set security key for OSDP devices.
The device identifier can send the identifier of the master device or the identifier of the slave device, but the operation is slightly different.
If key information is sent with the identifier of the master device as a device identifier, the master device will replace its primary key value with a new set key value.
At this time, OSDP slave devices should not be connected, and if there are connected devices, BS_SDK_ERROR_FAIL_MASTER_SET_KEY error may occur.
The master device sends a new key to process secure communication each time the OSDP device is connected.
Sending the OSDP slave device identifier as the device identifier, the master device sends the key it is holding to the slave device.
In other words, this command instructs the master device to send the key it holds to the specified OSDP device. Therefore, no separate key delivery is required in this case. (=NULL)

Declaration

```
#include "BS_API.h"

int BS2_SetOsdpStandardDeviceSecurityKey(void* context, BS2_DEVICE_ID
masterOrSlaveId, const BS2OsdpStandardDeviceSecurityKey* key);
```

See BS2OsdpStandardDeviceSecurityKey Structure

Parameter

- [In] *context* : Context
- [In] *masterOrSlaveId* : Device Identifier
- [In] *key* : Security key information structure pointer

Return Value

If successfully done, BS_SDK_SUCCESS will be returned. If there is an error, the corresponding error code will be returned.

See Also

BS2_AddOsdpStandardDevice
BS2_GetOsdpStandardDevice
BS2_GetAvailableOsdpStandardDevice
BS2_UpdateOsdpStandardDevice
BS2_RemoveOsdpStandardDevice
BS2_GetOsdpStandardDeviceCapability
BS2_SetOsdpStandardDeviceSecurityKey

Sample Code(C++)

sample_bs2_setosdpstandarddevicesecuritykey.cpp

```cpp
int sdkResult = BS_SDK_SUCCESS;

bool useMaster = true;
BS2_DEVICE_ID id =
(BS2_DEVICE_ID)Utility::selectMasterOrSlaveID(deviceList, useMaster);
if (useMaster)
{
    BS2OsdpStandardDeviceSecurityKey key = { , };
    string keyInfo = Utility::getInput<string>("Please enter the OSDP
security key.");
    memcpy(key.key, keyInfo.c_str(), min(keyInfo.size(),
BS2_OSDP_STANDARD_KEY_SIZE));

    sdkResult = BS2_SetOsdpStandardDeviceSecurityKey(context_, id,
&key);
}
else
{
    sdkResult = BS2_SetOsdpStandardDeviceSecurityKey(context_, id,
NULL);
}

if (BS_SDK_SUCCESS != sdkResult)
    printf("BS2_SetOsdpStandardDeviceSecurityKey call failed: %d",
sdkResult);

return sdkResult;
```

Sample Code (C#)

sample_bs2_setosdpstandarddevicesecuritykey.cs

```csharp
UInt32 id = Util.GetInputMasterOrSlaveID(deviceID);
BS2ErrorCode result = BS2ErrorCode.BS_SDK_SUCCESS;
```

```
if (id == deviceID)      // set key to master
{
    BS2OsdpStandardDeviceSecurityKey keyInfo =
Util.AllocateStructure<BS2OsdpStandardDeviceSecurityKey>();

    Util.HighlightLine(">>>> Please enter the OSDP security key.",
"security key");
    Console.Write(">>>> ");
    string keyString = Console.ReadLine();
    byte[] buff = Encoding.UTF8.GetBytes(keyString);

    Array.Clear(keyInfo.key, ,
BS2Environment.BS2_OSDP_STANDARD_KEY_SIZE);
    Array.Copy(buff, , keyInfo.key, , keyString.Length);

    IntPtr ptrKey =
Marshal.AllocHGlobal(Marshal.SizeOf(typeof(BS2OsdpStandardDeviceSecurit
yKey)));
    Marshal.StructureToPtr(keyInfo, ptrKey, false);

    result =
(BS2ErrorCode)API.BS2_SetOsdpStandardDeviceSecurityKey(sdkContext, id,
ptrKey);

    Marshal.FreeHGlobal(ptrKey);
}
else
{
    result =
(BS2ErrorCode)API.BS2_SetOsdpStandardDeviceSecurityKey(sdkContext, id,
IntPtr.Zero);
}

if (result != BS2ErrorCode.BS_SDK_SUCCESS)
    Console.WriteLine("Got error({0}).", result);
else
    Console.WriteLine("Call success.");

return result;
```