

## BaroPAM integration API(dll-en)

Nurit Co., Ltd.

Lee Jongil

June 1, 2023

### 1. Integration API configuration

BaroPAM related dynamic linking library is used to verify **one-time authentication keys**.

API class	Description	Etc
barokey.h libbarokey.dll	Header file and dll related to BaroPAM	
libcrypto-1_1-x64.dll libssl-1_1-x64.dll	Open SSL related dll	

Note) To use the BaroPAM related dll, it must be located in the "C:\Windows\System32" directory.

The header file Barokey.h for BaroPAM is as follows.

```
#ifndef _BAROKEY_API_H_
#define _BAROKEY_API_H_

#ifdef BAROPAMCORE_EXPORTS
#define BAROPAMCORE_API __declspec(dllexport)
#else
#define BAROPAMCORE_API __declspec(dllimport)
#endif

#ifdef __cplusplus
extern "C" {
#endif

BAROPAMCORE_API BOOL WINAPI BARO_ENCRYPT(const char* data, char* enc_result, unsigned long
buf_len);
BAROPAMCORE_API BOOL WINAPI BARO_DECRYPT(const char* data, char* dec_result, unsigned long
buf_len);
BAROPAMCORE_API BOOL WINAPI BARO_GENERATEKEY(const char* login_id, const char* phone_no,
const char* cycle_time, char* ota_key, unsigned long buf_len);
BAROPAMCORE_API BOOL WINAPI BARO_VERIFYKEY(const char* login_id, const char* phone_no,
const char* cycle_time, char* ota_key);
```

```

BAROPAMCORE_API char* WINAPI BARO_ENCRYPTA(const char* data);
BAROPAMCORE_API char* WINAPI BARO_DECRYPTA(const char* data);
BAROPAMCORE_API char* WINAPI BARO_GENERATEKEYA(const char* login_id, const char* phone_no,
const char* cycle_time);
BAROPAMCORE_API bool WINAPI BARO_VERIFYKEYA(const char* login_id, const char* phone_no,
const char* cycle_time, char* ota_key);

#ifdef __cplusplus
}
#endif

#endif // _BAROKEY_API_H_

```

## 2. Integration API function

### 1) BARO\_VERIFYKEY function

- NAME

BARO\_VERIFYKEY

- SYNOPSIS

```

BOOL BARO_VERIFYKEY(const char* login_id, const char* phone_no, const char* cycle_time,
char* ota_key)

```

- DESCRIPTION

A function that verifies whether the entered **one-time authentication key** is correct.

login\_id: Set the ID entered in the login-ID field of the login screen.

phone\_no: Login-ID set user's smart phone number only with numbers.

cycle\_time: Set the generation cycle (**3~60** seconds) of **one-time authentication key** specified for each user.

ota\_key: Set the **one-time authentication key** created and entered in the **BaroPAM** app on the login screen.

If the generation period of the smart phone number for each user and the **one-time authentication key** designated for each individual is different from the generator of the **one-time authentication key**, verification may fail because the **one-time authentication key** is different. You must match the information.

- RETURN VALUES

---

On success, **true** is returned, and on failure, **false** is returned.

### 3. Authentication key verification part

#### 1) Example of using authentication key verification module

Sample program) BARO\_VERIFYKEY function

```
#include <errno.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "barokey.h"

int main(int argc, char *argv[]) {
    const char *login_id = argv[1]; // Login-ID
    const char *phone_no = argv[2]; // Phone number
    const char *cycle_time = argv[3]; // Cycle time
    const char *key_method = argv[4]; // Key method
    char *ota_key = argv[5]; // Auth key

    // Authentication key verification
    BOOL bota_key = BARO_VERIFYKEY(login_id, phone_no, cycle_time, key_method, ota_key);

    // Authentication key verification (success)
    if (bota_key == true) {
        printf("Auth key success.\n");
    }
    // Authentication key verification (failure)
    } else {
        printf("Auth key failed.\n");
    }
}
```