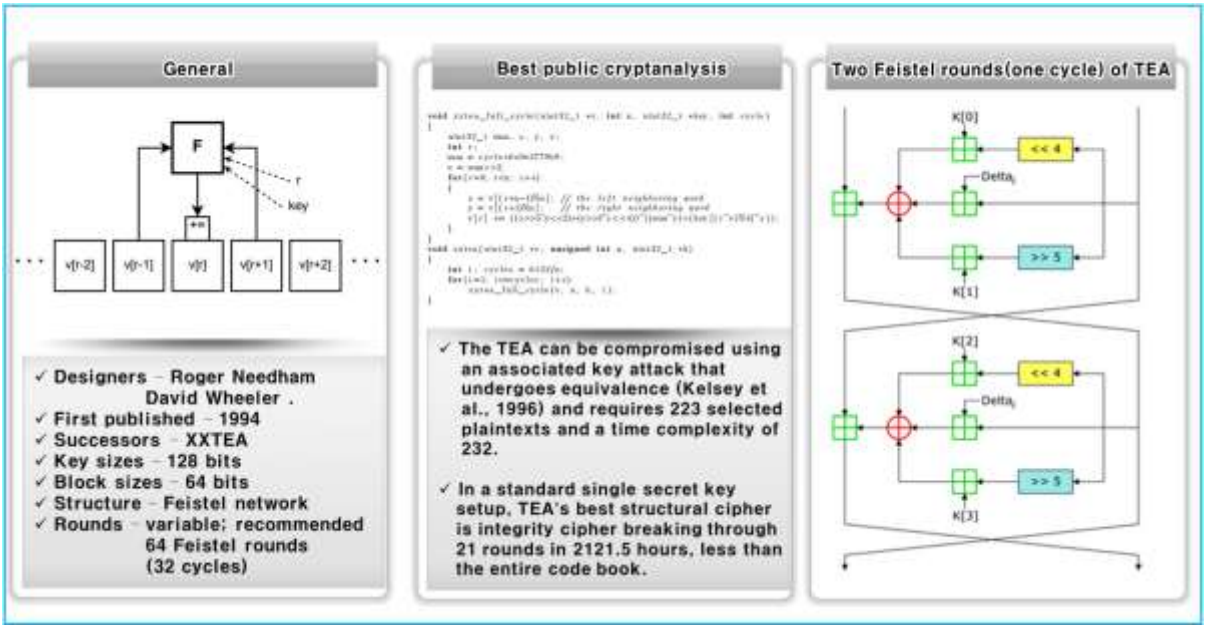
BaroCRYPT Guide(Oracle,Tibero)

Index

1. BaroCRYPT

1.1 BaroCRYPT Overview

The BaroCRYPT solution is a lightweight and fastest encryption algorithm based on the XXTEA (Extended Extended Tiny Encryption Algorithm), a compact and easy-to-implement block encryption algorithm using the Feistel cipher.



1.2 BaroCRYPT Features/Benefits

Based on the XXTEA (aka Corrected Block TEA) encryption algorithm, the BaroCRYPT solution is an optimal solution capable of quickly encrypting and decrypting data even under extreme constraints such as legacy hardware systems (embedded) with a minimum amount of usable RAM. Is as follows.

- ■ It is a small and easy-to-implement block encryption algorithm based on the Feistel cipher, which is small in size, fast and easy to implement.
- ■ It is a small-sized algorithm based on the Feistel cipher, and has high encryption strength compared to its size.
- ■ Although the size of the algorithm is small, it is the fastest and safest algorithm in existence.
- ■ Compared to other block encryption algorithms, it is easy to implement, easy to apply to environments with large hardware specification constraints, and freely used.
- ■ It is a block encryption algorithm that encrypts 64 bits (8 bytes) and uses a 128 bit (16 byte) key.
- ■ Corrected Block TEA (XXTEA) is a block cipher algorithm originally designed to correct the weakness of Block TEA
- ■ Provides free customizing and convenience of interlocking development with various application programs. (API integration in Java and C languages)
- ■ TO_ENCRYPTS (encryption) and TO_DECRYPTS (decryption) functions are provided for easy use in

SQL statements.

※ What is a Feistel Cipher?

It is a repetitive block cipher in which the ciphertext is encrypted from the plaintext while repeating the same substitution and substitution. It is a cipher similar to the Data Encryption Standard (DES). The other halves do an exclusive OR (XOR) and then swap each other. Do this process in the same pattern for each permutation, but do not exchange each other in the last permutation. The subkey used during encryption is reversed during decryption.

2. BaroCRYPT Integration API

2.1 Preparations before using the interlocking API

Since the BaroCRYPT module is written based on Java (barocrypt.jar) and C (libbarocrypt.so), the latest JDK 6.x or higher must be installed, and the environment settings for using the Java module are as follows.

① Java enbirenment environment settings(.profile)

```
export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.131.x86_64
export
CLASSPATH=$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/classes12.jar:$JAVA_HOME/lib/barocrypt.jar
```

② Java version check

```
> java -version
java version "1.7.0_131"
OpenJDK Runtime Environment (rhel-2.6.9.0.el5_11-x86_64 u131-b00)
OpenJDK 64-Bit Server VM (build 24.131-b00, mixed mode)
```

2.2 BaroCRYPT Integration API

1) C Module (libbarocrypt.so)

The symmetric key (64 bytes) used for field or data encryption/decryption is fixed inside the program, and in order to use the shared object (libbarocrypt.so), the directory (/home/baropam/crypt) where the shared object file exists must be set as Library must be set in the path.

```
For Linux: export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/baropam/crypt
For HP-UX: export SHLIB_PATH=$SHLIB_PATH:/home/baropam/crypt
For AIX:   export LIBPATH=$LIBPATH:/home/baropam/crypt
```

barocrypt.h, the header file for BaroCRYPT, is as follows

```
#ifndef BAROCRYPT_INCLUDED
#define BAROCRYPT_INCLUDED

#include <stdlib.h>

#ifdef __cplusplus
extern "C" {
#endif

/**
 * Function: baro_encrypt
 * @data:    Data to be encrypted
 * Returns:  Encrypted data or %NULL on failure
```

```
 *
 * Caller is responsible for freeing the returned buffer.
 */
void * baro_encrypts(const void * data);

/**
 * Function: baro_decrypt
 * @data:    Data to be decrypted
 * Returns:  Decrypted data or %NULL on failure
 *
 * Caller is responsible for freeing the returned buffer.
 */
void * baro_decrypts(const void * data);

#ifdef __cplusplus
}
#endif

#endif
```

① baro_encrypts function

- NAME
  baro_encrypts

- SYNOPSIS
  void * baro_encrypts(const void * data)

- DESCRIPTION
  A function that encrypts data.
  data: data to encrypt

- RETURN VALUES
  return encrypted data

② baro_decrypts function

- NAME
  baro_decrypts

- SYNOPSIS
  void * baro_decrypts(const void * data)

- DESCRIPTION
  A function to decrypt data.
  data: data to decrypt

- RETURN VALUES
  Return the decrypted data

③ Example of using data encryption/decryption

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

#include "barocrypt.h"

int main(int argc, char *argv[]) {
    const char *source_data = argv[1];
    const char *encrypt_data;
    const char *decrypt_data;

    encrypt_data = baro_encrypts(source_data );
    decrypt_data = baro_decrypts(encrypt_data);

    printf("Source   data = [%s]₩n", source_data );
    printf("encrypt data = [%s]₩n", encrypt_data);
    printf("decrypt data = [%s]₩n", decrypt_data);

    if (encrypt_data) free((void *)encrypt_data);
    if (decrypt_data) free((void *)decrypt_data);

    return 0;
}
```

2) Java Module (barocrypt.jar)

The symmetric key (64 bytes) used for field or data encryption is fixed inside the program, and to use the Java module (barocrypt.jar), you must download the directory (/home/baropam/crypt) where the barocrypt.jar file exists. The included Java module must be set in the class path.

```
export CLASSPATH=$CLASSPATH:/home/baropam/crypt/barocrypt.jar
```

① baro_encrypts function

- NAME
  baro_encrypts

- SYNOPSIS
  public static String baro_encrypts(String data)

- DESCRIPTION

A function that encrypts data.
data: data to encrypt

- RETURN VALUES
return encrypted data

② baro_decrypts 함수

- NAME
baro_decrypts

- SYNOPSIS
public static String baro_decrypts(String data)

- DESCRIPTION
A function to decrypt data.
data: data to decrypt

- RETURN VALUES
Return the decrypted data

③ Example of using data encryption/decryption

```
import barocrypt.barocrypt.*;

public static void main(String[] args) {
    try {
        String encrypt_data = baro_encrypts(args[0]       );
        String decrypt_data = baro_decrypts(encrypt_data);

        System.out.println("text              = [" + args[0]        + "]");
        System.out.println("encrypt_data = [" + encrypt_data + "]");
        System.out.println("decrypt_data = [" + decrypt_data + "]");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
    }
}
```

3) PowerBuider Module (barocrypt.dll)

The symmetric key (64 bytes) used for field or data encryption/decryption is fixed inside the program.

① Global External Function declaration

```
FUNCTION string baro_encrypts(string data) LIBRARY "barocrypt.dll" ALIAS FOR "baro_encrypts;ansi"
FUNCTION string  baro_decrypts(string data) LIBRARY "b.dll" ALIAS FOR "baro_decrypts;ansi"
```

② Example of using data encryption/decryption

```
string  ls_source_data = "qwerqwerqwer이종일qwerqwer";
string  ls_encrpt_data = "";
string  ls_decrpt_data = "";

ls_encrpt_data = baro_encrypts(ls_source_data)
MessageBox("Encryption", "encrpt_data = [" + ls_encrpt_data + "]")

ls_decrpt_data = baro_decrypts(ls_encrpt_data)
MessageBox("Decryption", "decrpt_data = [" + ls_decrpt_data + "]")

return
```

3. BaroCRYPT Integration API(DB)

3.1 What is an External Procedure?

There are cases where the functions provided by Oracle are not sufficient for complex formula calculations. In this case, after writing a complex function in a language such as C or JAVA, Oracle can improve the execution speed by passing parameters and receiving the result. In other words, after implementing something difficult or complex to implement in SQL using languages ??such as C, VB, or JAVA, it is called and used in SQL..

3.2 Advantages and Disadvantages

The advantage of using External Procedure is that Java or C can be reused. On the other hand, the disadvantage of using the External Procedure is that if the Session is not terminated, extProc is not an area where Oracle manages memory, but an O/S area. It remains, and there is no choice but to cause memory load of the O/S.

So, the method to reduce the memory of the O/S as much as possible is as follows.

① Adjust the number of sessions so that it does not exceed the memory limit of the O/S.

② Kill the old process among those created by extProc in the O/S. Even if you kill the process, it will be recreated if there is no big problem.

③ Eliminate the use of a function that calls an external procedure unnecessarily in an application, and minimize memory load by closing the session after use.

3.3 Java Module (barocrypts.java)

1) Java module Load

There are two ways to load a Java module into the Java space in Oracle as follows.

① Load only compiled sources

```
> loadjava -user baropam/baropam barocrypts.class
```

② Load up to source and compiled source

```
> loadjava -user baropam/baropam -resolve -v barocrypts.java
arguments: '-user' 'barocrypt/***' '-resolve' '-v' 'barocrypts.java'
creating: source barocrypts
loading : source barocrypts
created : CREATE$JAVA$LOB$TABLE
resolving: source barocrypts
```

```
Classes Loaded: 0
Resources Loaded: 0
Sources Loaded: 1
Published Interfaces: 0
Classes generated: 0
Classes skipped: 0
Synonyms Created: 0
Errors: 0

> loadjava -user baropam/baropam -force -resolve -verbose -synonym -grant public BaroClient.java
arguments: '-user' 'icam/***' '-force' '-resolve' '-verbose' '-synonym' '-grant' 'public' 'BaroClient.java'
creating: source BaroClient
loading : source BaroClient
granting: execute on source BaroClient to public
granting: execute on class BaroClient to public
resolving: source BaroClient
synonym : BaroClient
Classes Loaded: 0
Resources Loaded: 0
Sources Loaded: 1
Published Interfaces: 0
Classes generated: 0
Classes skipped: 0
Synonyms Created: 1
Errors: 0
```

2) Creation and confirmation of encryption stored function

After connecting to Oracle with sqlplus, create and check TO_ENCRYPTS (encryption function) and TO_DECRYPTS (decryption function) as follows.

```
> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 26 10:56:01 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE OR REPLACE FUNCTION TO_ENCRYPTS (input VARCHAR2) RETURN VARCHAR2 IS
  2          LANGUAGE JAVA
  3          NAME 'barocrypts.barocrypt_encrypts(java.lang.String) return java.lang.String';
  4  /

Function created.

SQL> CREATE OR REPLACE FUNCTION TO_DECRYPTS (input VARCHAR2) RETURN VARCHAR2 IS
  2          LANGUAGE JAVA
  3          NAME 'barocrypts.barocrypt_decrypts(java.lang.String) return java.lang.String';
  4  /
```

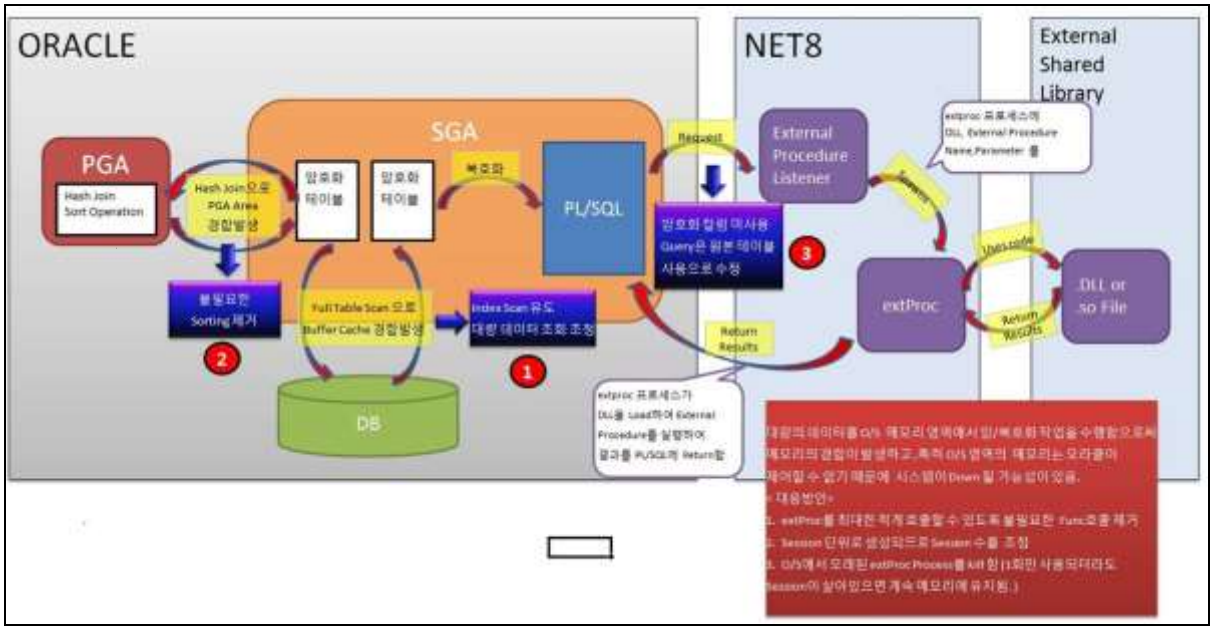
```
Function created.

SQL> commit;

Commit complete.

SQL> SELECT OBJECT_NAME, OBJECT_TYPE, STATUS FROM USER_OBJECTS WHERE
OBJECT_TYPE LIKE 'JAVA%';

OBJECT_NAME
--------------------------------------------------------------------------------
OBJECT_TYPE                             STATUS
----------------------------------- -------------
barocrypts
JAVA CLASS                              VALID

barocrypts
JAVA SOURCE                             VALID
```

3) Encryption function (TO_ENCRYPTS, TO_DECRYPTS) test

```
> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 12 11:34:24 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production

SQL> SELECT TO_ENCRYPTS('qwerqwerqwer0이종일qwerqwer') FROM DUAL;

TO_ENCRYPTS('QWERQWERQWER0이종일QWERQWER')
--------------------------------------------------------------------------------
BDx8KvL4xf0dHUf7LJl/edUWGwaJGGYtzYKhc5VvcHdnBArs

SQL> SELECT TO_DECRYPTS('BDx8KvL4xf0dHUf7LJl/edUWGwaJGGYtzYKhc5VvcHdnBArs') FROM DUAL;

TO_DECRYPTS('BDX8KVL4XF0DHUF7LJL/EDUWGWAJGGYTZYKHC5VVCHDNBARS')
--------------------------------------------------------------------------------
qwerqwerqwer0이종일qwerqwer
```

3.4 C Module (libbarocrypts.so)

Unlike Java modules, it is cumbersome to use external procedures in C modules, but the operation sequence of external procedures is as follows.



① The user makes a request to the DB for the function written in External Procedure in SQL.

② While parsing the corresponding statement in the Shared SQL Area, the external procedure is known and the NET8 Listener requests that the external procedure be interpreted because the user SQL called the external procedure.

③ Listener creates an extProc process again and transfers DLL, Procedure Name, and Parameter with External Procedure in O/S.

④ extProc finds the DLL file in the O/S and loads it into the memory of the O/S to process the result of the requested function.

⑤ The processed result is returned to SQL.

⑥ When the Session is terminated, extproc is automatically terminated.

1) Create C module library

The C module creates an encryption/decryption library (libcrypt_encrypts, libcrypt_decrypts) in Oracle as follows.

```
> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 12 14:13:40 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE OR REPLACE LIBRARY libcrypt_encrypts AS '/home/baropam/crypt/libbarocrypts.so' ;
  2  /
```

```
Library created.

SQL> CREATE OR REPLACE LIBRARY libcrypt_decrypts AS '/home/baropam/crypt/libbarocrypts.so' ;
  2  /

Library created.

SQL> commit;

Commit complete.
```

Note) If an "**ORA-01031: insufficient privileges**" error occurs while creating a library, it is caused because the account creating the library (baropam) does not have permission to create the library. In this case, you must connect as Oracle sysdba and grant the library creation authority to the account that creates the library.

```
SQL> create library to baropam ;
```

2) Creation and confirmation of encryption stored function

After connecting to Oracle with sqlplus, create and check TO_ENCRYPTS (encryption function) and TO_DECRYPTS (decryption function) as follows.

```
project:icam /usr/baropam> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 12 14:16:06 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE OR REPLACE FUNCTION TO_ENCRYPTS (input VARCHAR2) return VARCHAR2
        as external
        language C
        library libcrypt_encrypts
        name "baro_encrypts"
        parameters  (input  STRING);
  2     3     4     5     6     7  /

Function created.

SQL> CREATE OR REPLACE FUNCTION TO_DECRYPTS (input VARCHAR2) return VARCHAR2
        as external
        language C
        library libcrypt_decrypts
        name "baro_decrypts"
        parameters  (input  STRING);
  2     3     4     5     6     7  /

Function created.
```

```
SQL> commit;

Commit complete.
```

After connecting to Tibero with tsql, create and check TO_ENCRYPTS (encryption function) and TO_DECRYPTS (decryption function) as follows.

```
SQL> CREATE OR REPLACE FUNCTION TO_ENCRYPTS (input VARCHAR2) return VARCHAR2
        as language C
        library libcrypt_encrypts
        name "baro_encrypts"
        parameters   (input   STRING);
  2     3     4     5     6     7   /

Function created.

SQL> CREATE OR REPLACE FUNCTION TO_DECRYPTS (input VARCHAR2) return VARCHAR2
        as language C
        library libcrypt_decrypts
        name "baro_decrypts"
        parameters   (input   STRING);
  2     3     4     5     6     7   /

Function created.

SQL> commit;

Commit complete.
```

3) listener configuration

Before using the C module, you must configure the listener to include the path to the desired shared library. It is performed by setting the EXTPROC_DLLS environment variable. After logging in to the Oracle account, set the listener.ora file as follows.

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /oracle/app/oracle/product/11.2.0)
      (PROGRAM = extproc)
      (ENVS="EXTPROC_DLLS=ANY")
    )
    (SID_DESC =
      (SID_NAME = ODB)
      (ORACLE_HOME = /oracle/app/oracle/product/11.2.0)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
```

```
        (ADDRESS = (PROTOCOL = TCP)(HOST = project)(PORT = 1521))
    )
  )

ADR_BASE_LISTENER = /oracle/app/oracle
```

4) tnsnames.ora configuration

Before using the C module, log in to the Oracle account with the same values ??as the Key and SID set in the listener.ors file, and set EXTPROC_CONNECTION_DATA in the tnsnames.ora file as follows.

```
ODB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = project)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ODB)
    )
  )
EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
        (ADDRESS = (PROTOCOL = IPC)(HOST = project)(KEY = EXTPROC1521))
        (CONNECT_DATA = (SID = PLSExtProc))
)
```

After configuring the tnsnames.ora file, be sure to log in to your Oracle account and restart the listener.

```
> lsnrctl start | stop | status
```

5) Encryption function (TO_ENCRYPTS, TO_DECRYPTS) test

```
> sqlplus baropam/baropam

SQL*Plus: Release 11.2.0.1.0 Production on 화 9월 12 14:16:06 2017

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SELECT TO_ENCRYPTS('qwerqwerqwer0이종일qwerqwer') FROM DUAL;

TO_ENCRYPTS('QWERQWERQWER0이종일QWERQWER')
--------------------------------------------------------------------------------
LkrLD7uCXBnPZreic9NgsHgsDjWQG1QQL0w9UHndzy8=

SQL> SELECT TO_DECRYPTS('LkrLD7uCXBnPZreic9NgsHgsDjWQG1QQL0w9UHndzy8=') FROM DUAL;

TO_DECRYPTS('LKRLD7UCXBNPZREIC9NGSHGSDJWQG1QQL0W9UHNDZY8=')
--------------------------------------------------------------------------------
qwerqwerqwer0이종일qwerqwer
```

4. About BaroCRYPT

Company: Nurit Co., Ltd.
Registration Number: 258-87-00901
CEO: Jongil Lee
Tel: +8210-2771-4076(Technical support, sales inquiry)
email: mc529@nurit.co.kr
Address: #913, 15, Magokjungang 2-ro, Gangseo-gu, Seoul (Magok-dong, Magok Techno Tower 2)