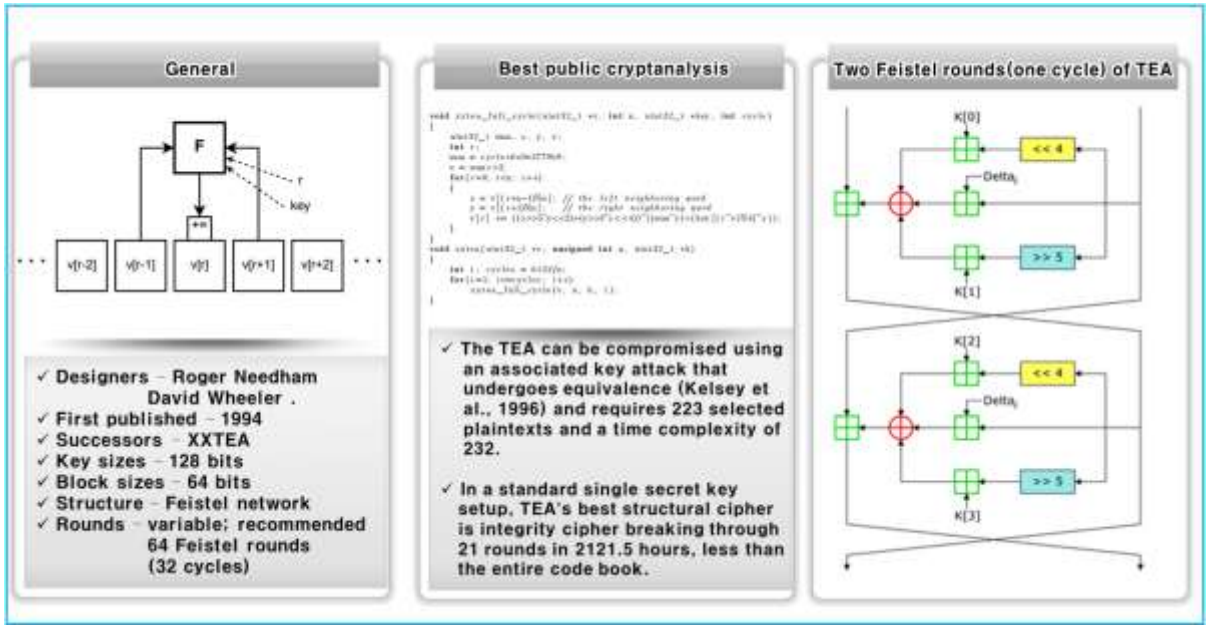
BaroCRYPT Guide(SQL Server)

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 Integration API configuration

Dynamic linking library related to BaroPAM is used for data encryption/decryption.

API class	Descrption	Etc
barokey.h libbarokey.dll	BaroPAM related C++ dll version. (Compiled based on .NET Framework 4.0)	
libcrypto-1_1-x64.dll libssl-1_1-x64.dll	Open SSL related dll	

Note) To use BaroPAM's encryption/decryption dll, it must be located in the "C:₩Windows₩System32" directory.

The header file for BaroPAM encryption/decryption is as follows.

barokey.h)

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

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

BAROPAMCORE_API wchar_t* BARO_ENCRYPTSW(const wchar_t* data);
BAROPAMCORE_API wchar_t* BARO_DECRYPTSW(const wchar_t* data);
BAROPAMCORE_API wchar_t* BARO_GENERATEKEYW(const wchar_t* login_id, const wchar_t* phone_no,
const wchar_t* cycle_time);
```

```
BAROPAMCORE_API bool BARO_VERIFYKEYW(const wchar_t* login_id, const wchar_t* phone_no, const wchar_t* cycle_time, wchar_t* totp);

#ifdef __cplusplus
}
#endif

#endif //_BAROKEY_API_H_
```

2.2 Integration API function

1) BARO_ENCRYPTA function

- NAME
  BARO_ENCRYPTA

- SYNOPSIS
  char * BARO_ENCRYPTA(const void * data)

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

- RETURN VALUES
  return encrypted data

2) BARO_DECRYPTA function

- NAME
  BARO_DECRYPTA

- SYNOPSIS
  char * BARO_DECRYPTA(const void * data)

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

- RETURN VALUES
  Return the decrypted data

Note) When using barokey.jar, a Java library module

Sequence of the work)

① ikvmbin-7.2.4630.5.zip download

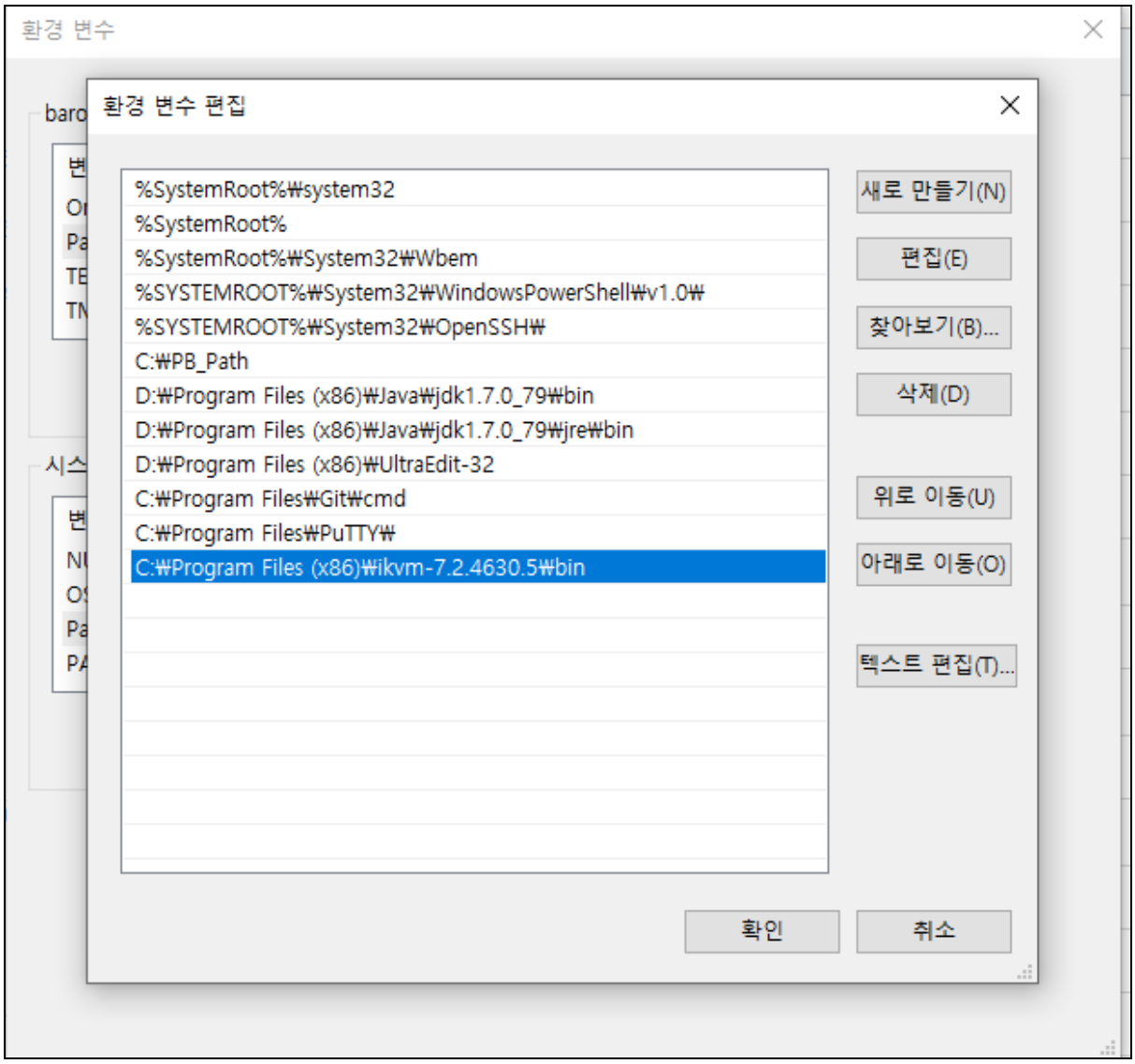
ikvm is a tool that converts a Java module (jar) into a dll for use in C#.

Download: https://osdn.net/projects/sfnet_ikvm/downloads/ikvm/7.2.4630.5/ikvmbin-7.2.4630.5.zip/

② Unzip ikvmbin-7.2.4630.5.zip

Unpack ikvmbin-7.2.4630.5.zip to the "C:₩Program Files (x86)₩" directory.

③ Setting the environment variable (PATH)



④ Compile barokey.java (compiled on java version "1.7.0_79")

```
C:₩workWetc> javac barokey.java
```

⑤ Create the barokey.jar file

```
C:₩workWetc> jar cf barokey.jar barokey.class
```

⑥ Convert barokey.jar file to barokey.dll file through ikvm

```
C:\work\etc> ikvmc barokey.jar
```

⑦ To use the barokey.dll file as a dll, it must be located in the "C:\Windows\System32" directory.

3. BaroCRYPT Integration API(DB)

3.1 Create Stored Function

Create Stored FunctionCreate TO_ENCRYPTS (encryption function) and TO_DECRYPTS (decryption function) in the SQL window, that is, in SQL Server Manager query mode, as follows.

```
CREATE FUNCTION TO_ENCRYPTS (
    @data nvarchar(2000)
) RETURNS nvarchar(2000)
AS EXTERNAL NAME libbarokey.[libbarokey.barokey].BARO_ENCRYPTA
GO

CREATE FUNCTION TO_DECRYPTS (
    @data nvarchar(2000)
) RETURNS nvarchar(2000)
AS EXTERNAL NAME libbarokey.[libbarokey.barokey].BARO_DECRYPTA
GO
```

Note) When using barokey.jar, a Java library module

```
CREATE FUNCTION TO_ENCRYPTS (
    @data nvarchar(2000)
) RETURNS nvarchar(2000)
AS EXTERNAL NAME barokey.[barokey.barokey].baro_encrypts
GO

CREATE FUNCTION TO_DECRYPTS (
    @data nvarchar(2000)
) RETURNS nvarchar(2000)
AS EXTERNAL NAME barokey.[barokey.barokey].baro_decrypts
GO
```

3.2 Stored Function tests

Test TO_ENCRYPTS (encryption function) and TO_DECRYPTS (decryption function) created in the SQL window, that is, SQL Server Manager query mode.

```
SELECT TO_ENCRYPTS('qwerqwerqwer이종일qwerqwer');

TO_ENCRYPTS('QWERQWERQWER이종일QWERQWER')
--------------------------------------------------------------------------------

BDx8KvL4xf0dHUf7LJl/edUWGwaJGGYtzYKhc5VvcHdnBArs

SELECT TO_DECRYPTS('BDx8KvL4xf0dHUf7LJl/edUWGwaJGGYtzYKhc5VvcHdnBArs');
```

```
TO_DECRYPTS('BDX8KVL4XFODHUF7LJL/EDUWGWAJGGYTZYKHC5VVCHDNBARS')
--------------------------------------------------------------------------------

qwerqwerqwer이종일qwerqwer
```

4. About BaroCRYPT

Version 1.0 - Official Release - 2016.12.1

http://www.nurit.co.kr

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)