

BaroCRYPT(SQL Server) 가이드

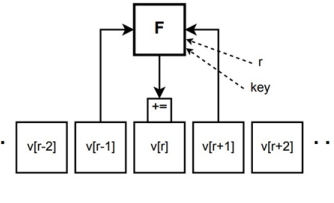
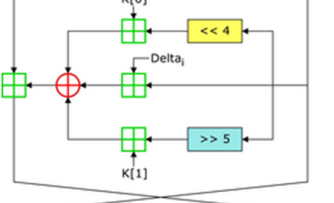
목차

목차.....	0
1. BaroCRYPT	1
1.1 BaroCRYPT 개요.....	1
1.2 BaroCRYPT 특/장점.....	1
2. BaroCRYPT 연동 API	3
2.1 연동 API 구성	3
2.2 연동 API 함수	4
3. BaroCRYPT 연동 API (DB)	7
3.1 Stored Function 생성	7
3.2 Stored Function 테스트	7
4. About BaroCRYPT.....	9

1. BaroCRYPT

1.1 BaroCRYPT 개요

BaroCRYPT 솔루션은 Feistel 암호를 사용하여 크기가 작고 구현이 쉬운 블록 암호화 알고리즘인 XXTEA (Extended Extended Tiny Encryption Algorithm)를 기반으로 하는 가볍고 가장 빠른 암호화 알고리즘이다.

General	Best public cryptanalysis	Two Feistel rounds(one cycle) of TEA
	<pre> void xxtea_full_cycle(uint32_t *v, int n, uint32_t *key, int cycle) { uint32_t sum, z, y, e; int r; sum = cycle * 0x9e3779b9; e = sum >> 2; for (r=0; r<n; r++) { z = v[(r-1)%n]; // the left neighboring word y = v[(r+1)%n]; // the right neighboring word v[r] += ((z >> 5 ^ y << 2) + (y >> 7 ^ z << 4)) ^ ((sum ^ y) + (key[(r^e)%4] ^ z)); } } void xxtea(uint32_t *v, unsigned int n, uint32_t *k) { int i, cycles = 6+52/n; for (i=1; i<=cycles; i++) xxtea_full_cycle(v, n, k, i); } </pre>	
<ul style="list-style-type: none"> ✓ Designers - Roger Needham David Wheeler . ✓ First published - 1994 ✓ Successors - XXTEA ✓ Key sizes - 128 bits ✓ Block sizes - 64 bits ✓ Structure - Feistel network ✓ Rounds - variable; recommended 64 Feistel rounds (32 cycles) 	<ul style="list-style-type: none"> ✓ TEA는 동등한 키 (Kelsey et al., 1996)를 겪고 223 개의 선택된 평문과 232의 시간 복잡성을 요구하는 관련 키 공격을 사용하여 손상 될 수 있음. ✓ 표준 단일 비밀 키 설정에서 TEA의 최상의 구조적 암호 해독은 전체 Code book 보다 적은 2121.5 시간에 21 라운드를 돌파하는 무결성 암호 해독임. 	

1.2 BaroCRYPT 특/장점

BaroCRYPT 솔루션은 XXTEA(일명 Corrected Block TEA) 암호화 알고리즘을 기반으로 사용 가능한 RAM의 양이 최소인 레거시 하드웨어 시스템(임베디드)과 같이 극한의 제약이 있는 상황에서도 빠르게 데이터 암호화를 실행 가능한 최적의 솔루션으로 특징점은 다음과 같다.

- 작고 구현이 쉬운 블록 암호화 알고리즘으로 페이스텔 암호를 기반으로 하여 크기가 작고 빠르면서 구현이 쉬움
- 페이스텔 암호를 기반으로 한 작은 크기의 알고리즘으로 그 크기에 비해서 암호화 강도가 높음
- 알고리즘의 크기는 작지만 현존하는 가장 빠르고 안전한 알고리즘
- 다른 블록 암호화 알고리즘에 비해 구현이 용이하고 하드웨어 사양 제약 조건이 큰 환경에 적용이 용이하며 자유롭게 사용
- 64 bit(8byte)를 암호화하는 블록 암호화 알고리즘으로 128 bit(16byte) 키를 사용
- Corrected Block TEA(XXTEA)는 원래 Block TEA의 약점을 수정하기 위해 고안된 블록 암호화 알고리즘
- 자유로운 Customizing 및 다양한 응용프로그램과 연동 개발의 편의성 제공 (Java, C 언어로 된 API 연동)
- SQL 문장에서 쉽게 사용할 수 있도록 TO_ENCRYPTS(암호화), TO_DECRYPTS(복호화) 함수 제공

※ Feistel 암호란?

동일한 대치와 치환을 반복하면서 암호문이 평문으로부터 암호화되는 반복 블록 암호로 데이터 암호화 표준(DES)과 유사한 암호로서, 평문을 반씩 2개 블록으로 나누어 한쪽은 서브 키를 사용한 기능 F로 치환하고 그 결과를 다른 반쪽에서 배타적 논리합(XOR)한 다음 서로 교환한다. 이러한 과정을 각 치환마다 동일

한 패턴으로 하되 마지막 치환에서는 서로 교환하지 않는다. 암호화시 사용된 서브키는 복호화시 역으로 사용된다.

2. BaroCRYPT 연동 API

2.1 연동 API 구성

BaroPAM 관련 Dynamic linking library는 데이터에 대한 암호화하는데 사용된다.

API구분	설명	비고
barokey.h libbarokey.dll	BaroPAM 관련 C++ dll 버전임. (.NET Framework 4.0 기반으로 컴파일 함)	
libcrypto-1_1-x64.dll libssl-1_1-x64.dll	Open SSL 관련 dll	

참고) BaroPAM의 암호화 dll을 사용하기 위해서는 반드시 "C:\Windows\System32" 디렉토리에 위치해야 한다.

BaroPAM의 암호화에 대한 header 파일인 다음과 같다.

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 연동 API 함수

1) BARO_ENCRYPTA 함수

- NAME
BARO_ENCRYPTA
- SYNOPSIS
char * BARO_ENCRYPTA(const void * data)
- DESCRIPTION
데이터를 암호화하는 함수
data: 암호화할 데이터
- RETURN VALUES
암호화한 데이터를 반환

2) BARO_DECRYPTA 함수

- NAME
BARO_DECRYPTA
- SYNOPSIS
char * BARO_DECRYPTA(const void * data)
- DESCRIPTION
데이터를 복호화 하는 함수
data: 복호화할 데이터
- RETURN VALUES
복호화한 데이터를 반환

참고) Java 라이브러리 모듈인 barokey.jar를 사용하는 경우

작업순서)

① ikvbin-7.2.4630.5.zip 다운로드

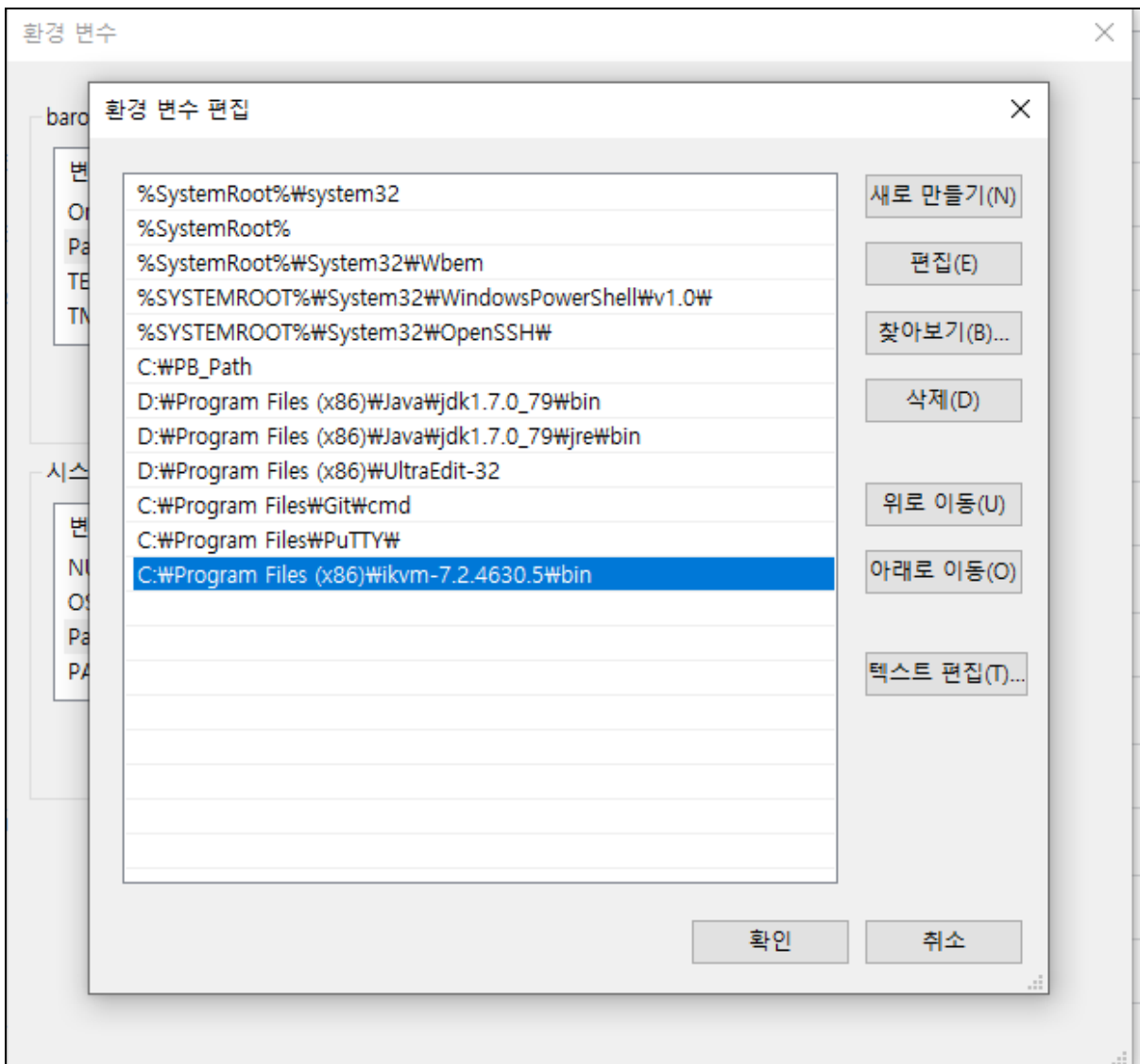
ikvm은 Java 모듈(jar)을 C#에서 사용할 수 있도록 dll로 변환해주는 툴임.

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

② ikvmbin-7.2.4630.5.zip 압축 풀음

ikvmbin-7.2.4630.5.zip를 "C:\Program Files (x86)\W" 디렉토리에 압축을 풀음.

③ 환경변수(PATH) 설정



④ barokey.java 컴파일(java version "1.7.0_79"에서 컴파일 함)

```
C:\work\etc> javac barokey.java
```

⑤ barokey.jar 파일 생성

```
C:\work\etc> jar cf barokey.jar barokey.class
```

⑥ barokey.jar 파일을 ikvm을 통해서 barokey.dll 파일로 변환

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

⑦ barokey.dll 파일을 dll을 사용하기 위해서는 반드시 "C:\Windows\System32" 디렉토리에 위치해야 한다.

3. BaroCRYPT 연동 API (DB)

3.1 Stored Function 생성

SQL 창에 즉 SQL Server 매니저 질의모드에서 TO_ENCRYPTS(암호화 함수), TO_DECRYPTS(복호화 함수)를 다음과 같이 생성한다.

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

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

참고) Java 라이브러리 모듈인 barokey.jar를 사용하는 경우

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

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

3.2 Stored Function 테스트

SQL 창에 즉 SQL Server 매니저 질의모드에서 생성한 TO_ENCRYPTS(암호화 함수), TO_DECRYPTS(복호화 함수)를 테스트한다.

```
SELECT TO_ENCRYPTS('qwerqwerqwer이종일qwerqwer');

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

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


TO_DECRYPTS('BDX8KVL4XF0DHUF7LJL/EDUWGNAJGGYTZYKHC5VVCHDNBARS')

qwer qwer qwer 이종일 qwer qwer

4. About BaroCRYPT



Version 1.0 - Official Release - 2016.12.1
Copyright © Nurit corp. All rights reserved.
<http://www.nurit.co.kr>

상호: 주식회사 누리아이티
등록번호: 258-87-00901
대표이사: 이종일
대표전화: 02-2665-0119(기술지원/영업문의)
이메일: mc529@nurit.co.kr
주소: 서울시 강서구 마곡중앙2로 15, 913호(마곡동, 마곡테크노타워2)