Integration of Indect Block Cipher into the OpenSSL library

Piotr Jurkiewicz
Department of Telecommunications
AGH University of Science and Technology

December 2012

Contents

1. Introduction

This document describes the process of integration the Indect Block Cipher with the OpenSSL library.

At the beginning the OpenSSL toolkit is introducted. Cryptographic feauters of the library are presented, with emphasis on symmetric encryption. Next, secure communication features provided by the library are described. Basics of SSL/TLS protocol are introduced.

Chapter 3 presents the details of integration of new cipher with the OpenSSL library. Modifications needed in the OpenSSL source code for enabling a new cipher are described. Moreover, process of implementation of new SSL/TLS ciphersuites is presented.

Chapter 4 presents the crucial part of work – the implementation of IBC algorithm. The cipher code was completely rewrited using low-level C language. Functions performing the key setup and actual encryption and decryption are presented in detail, with emphasis on performance issues.

After the chapters presenting the programming work, results of tests are shown. First, the compatibility test is presented. The test checks the backward binary compatibility of modified library with the original one. Next, the performance tests are presented. Performance of rewrited IBC code is compared with the performance of graphical application from D9.13. Finally, the usage of IBC cipher in SSL/TLS connections is tested.

There are three appendices to this document. Appendix A is the compatibility report proving binary compatibility of modified library. Appendix B is a report from *diffstat* program. It shows a number of changes made in each source file. Appendix C is a full diff file. It can be used to form a patch on openssl0.9.8v source.

2. OpenSSL library

OpenSSL is an open source toolkit, implementing the SSL and TLS protocols, as well as general purpose cryptographic functions. It is written in the C programming language. Versions are available for most operating systems, including Unix-like ones, Windows and MacOS. OpenSSL shared libraries are preinstalled in many Linux distributions. OpenSSL consist of three major components: libraries (*libcrypto* and *libssl*) and command line tool (*openssl*).

2.1 Cryptographic features

libcrypto library implements a wide range of cryptographic-related functions, including: symmetric encryption, public key cryptography, certificate handling and hash functions. The services provided by this library are used by the OpenSSL implementations of SSL/TLS protocols and openssl command line tool. This library can also be used in other applications. It has been used to implement OpenSSH, OpenPGP, and others. [1]

2.1.1 Symmetric encryption/decryption

OpenSSL library provides several built-in symmetric cipher algorithms. It includes AES, DES, Blowfish, IDEA and other algorithms. Most of the ciphers can work in block and stream modes. OpenSSL supports the following modes:

Block modes:
- ECB (Electronic Codebook)
- CBC (Cipher Block Chaining)

Stream modes:
- CFB-x (Cipher Feedback)
- OFB (Output Feedback)
- CTR (Counter Mode)

The CFB mode can operate on segments consists of 1 bit, 8 bits or equal to the block size of underlying block cipher.

The *libcrypto* library provides API which allow these ciphers to be used to encrypt or decrypt data in applications using the library. Moreover, the *openssl* command-line tool *enc* command can call crypto functions to encrypt or decrypt arbitrally files, using keys based on passwords or explicitly provided.

2.2 Secure communication features

The OpenSSL toolkit provides number of features useful in providing secure communication over Internet. The libssl library implements the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. The library can be used by developers for securing third-party applications. One of such application is OpenVPN.

2.2.1 SSL/TLS protocol

SSL (Secure Sockets Layer) and its successor, TLS (Transport Layer Security) are network protocols that provide secure communication over the Internet. They integrate the data cryptography functions, allowing client/server applications to communicate without danger of eavesdropping and tampering.

SSL/TLS is usually implemented on top of some transport protocol (e.g. TCP). It encapsulates the application data. One advantage of SSL/TLS is that it is application protocol independent. A higher level protocol can layer on top of the SSL/TLS protocol transparently.

The protocol itself is composed of two layers. At the lowest level, layered on top of transport protocol, is the Record Protocol. This protocol ensures confidentiality and integrity of transmitted data.

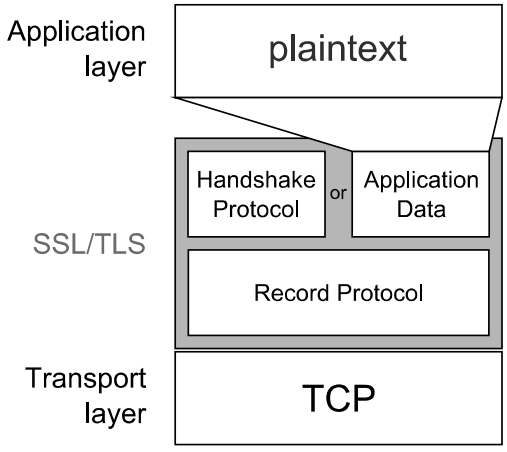

Figure 1: SSL/TLS protocol

Symmetric cryptography is used for data encryption (e.g. AES or IBC). The keys for this symmetric encryption are generated uniquely for each connection and are based on a secret negotiated by the Handshake Protocol. Encryption algorithm is chosen during the negotiation. The data can also be compressed.

Message transport includes a message integrity check using a keyed Message Authentication Code (MAC). Secure hash functions (e.g. SHA-1, MD5) can be used for MAC computations. The record protocol also fragments the data into blocks and numbers its sequence to protect against the data reordering. SSL/TLS supports many combinations of ciphers, authentication mechanisms and hashing algorithms.

The Record Protocol is used for encapsulation of higher level protocols: application protocol and Handshake Protocol. Handshake Protocol allows the server and client to authenticate each other and to negotiate cryptographic keys before the application transmits its data. This protocol provides authenticity of peers and the integrity of negotiation procedure.

For peer authentication, the Handshake Protocol uses X.509 certificates. This authentication is generally required for at least one of the peers, usually a server. Certificates are issued by certificate authorities. The certificate binds a particular public key to the entity the certificate identifies (e.g. a server). A certificate prevents the usage of fake public keys for impersonation.

Such authenticated server's public key is used by the SSL/TLS protocol for two purposes. Firstly, it is used to authenticate the server to the client. The client encrypts some random data using server's public key. The server decrypts that data using its own private key. Because the private key is kept secret, only the server can decrypt the data. If server does it successfully, client can be sure that talks with the genuine peer.

That random data is also used to establish a session key. This key will be used in the symmetric algorithms to encrypt the application data. As the random data is transmitted encrypted, only the server and the client will be able to derive the right session key. Thus, the SSL/TLS protocol combines benefits of asymmetric cryptography for authentication with the faster symmetric cryptography for the application data.

3. Integration of IBC with the OpenSSL library

Integrating a new cryptographic algorithm requires a number of modifications to the OpenSSL code. Several *libcrypto* subroutines, which implements individual functions, must know all available ciphers. New algorithm modes should be properly registered and calls to cipher routines should be added into the *libcrypto* code. Moreover, new SSL/TLS Ciphersuites, containing a new cipher, must be created in the *libssl*, in order to make it available for usage in secure communication. This chapter presents all the necessary modifications that make IBC routines fully available in all OpenSSL utilities, as well in SSL/TLS connections.

3.1 Modifications in apps/

The *openssl* program (command line tool) provides a rich variety of commands, each of which often has a wealth of options and arguments. These commands are implemented as separate programs in apps/ directory in the source tree. Each command program is implemented in separate source file, with its own **main()** function. Therefore, each program has its own command line arguments parsing code.

These programs, which make use of IBC, must have implemented code which parses arguments enabling IBC. This code part usually looks like:

```
#ifndef OPENSSL_NO_INDECT
        else if (!strcmp(*args,"-indect128"))
                cipher = EVP_indect_128_cbc();
        else if (!strcmp(*args,"-indect192"))
                cipher = EVP_indect_192_cbc();
        else if (!strcmp(*args,"-indect320"))
                cipher = EVP_indect_320_cbc();
#endif
```

Moreover, each program which can take IBC options must print these options in the usage message. Code responsible for this usually looks like:

```
#ifndef OPENSSL_NO_INDECT
        BIO_printf (bio_err, "-indect128, -indect192, -indect320\n");
        BIO_printf (bio_err, "                encrypt PEM output with cbc
                             indect\n");
#endif
```

Programs from apps/ which need to have implemented IBC arguments parsing are:

cms (cms.c) - CMS (Cryptographic Message Syntax) utility, lines: 229-236; 613-616
dsa (dsa.c) - DSA Data Management utility, lines: 90-92; 224-227
gendsa (gendsa.c) - Generation of DSA private key from parameters, lines: 160-167; 202-205
genrsa (genrsa.c) - Generation of RSA private key, lines: 183-190; 222-225
pkcs12 (pkcs12.c) - PKCS#12 Data Management, lines: 187-191; 344-347
rsa (rsa.c) - RSA key management, lines: 91-93; 229-232
smime (smime.c) - S/MIME mail processing, lines: 176-183; 454-457

3.1.1 apps/progs.h

This header file contains headers for all programs from apps directory. Moreover, it contains array which registers all standard commends (programs), message digest commands, and cipher commands, which are available in the *openssl* utility. The IBC cipher modes should be registered in order to make them available in the *openssl* utility (lines 192-208).

```
#ifndef OPENSSL_NO_INDECT
    {FUNC_TYPE_CIPHER,"indect-128-cbc",enc_main},
#endif
#ifndef OPENSSL_NO_INDECT
    {FUNC_TYPE_CIPHER,"indect-128-ecb",enc_main},
#endif
#ifndef OPENSSL_NO_INDECT
    {FUNC_TYPE_CIPHER,"indect-192-cbc",enc_main},
#endif
#ifndef OPENSSL_NO_INDECT
    {FUNC_TYPE_CIPHER,"indect-192-ecb",enc_main},
#endif
#ifndef OPENSSL_NO_INDECT
    {FUNC_TYPE_CIPHER,"indect-320-cbc",enc_main},
#endif
#ifndef OPENSSL_NO_INDECT
    {FUNC_TYPE_CIPHER,"indect-320-ecb",enc_main},
#endif
```

The progs.h file is generated automatically by progs.pl script. IBC should be listed in this script too. The IBC modes are defined in lines 65-67 and line 88.

3.1.2 apps/speed.c

openssl speed is a program used to test the performance of cryptographic algorithms. In order to make it able to test the IBC algorithm its code requires several modifications.
First, the IBC main header file should be included:

```
#ifndef OPENSSL_NO_INDECT
#include <openssl/indect.h>
#endif
```

Next, the constant defining number of available algorithms should be incremented (line 291):

```
#define ALGOR_NUM   28
```
to
```
#define ALGOR_NUM   31
```

After that, the names of IBC algorithms should be appended to array containing names of all available ciphers. It is important to append additional ciphers at the end of array.

```
static const char *names[ALGOR_NUM]={
  "md2","mdc2","md4","md5","hmac(md5)","sha1","rmd160","rc4",
  "des cbc","des ede3","idea cbc","seed cbc",
  "rc2 cbc","rc5-32/12 cbc","blowfish cbc","cast cbc",
  "aes-128 cbc","aes-192 cbc","aes-256 cbc",
  "camellia-128 cbc","camellia-192 cbc","camellia-256 cbc",
  "evp","sha256","sha512",
  "aes-128 ige","aes-192 ige","aes-256 ige",
  "indect-128 cbc","indect-192 cbc","indect-320 cbc"};
```

Next, keys used in the performance tests should be defined (lines 589-600):

```
#ifndef OPENSSL_NO_INDECT
    static const unsigned char ikey24[24]=
        {0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,
         0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,
         0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34};
    static const unsigned char ikey40[40]=
        {0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,
         0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,
         0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34,
         0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,
         0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34,0x56};
#endif
```

And key structures should be declared (lines 622-624):

```
#ifndef OPENSSL_NO_INDECT
    INDECT_KEY indect_ks1, indect_ks2, indect_ks3;
#endif
```

Next, constants defining numbers of IBC algorithms need to be defined (lines 653-655):

```
#define D_CBC_128_IBC   28
#define D_CBC_192_IBC   29
#define D_CBC_320_IBC   30
```

The speed program requires IBC options parsing code similar to other apps/ programs (lines 1004-1009 and 1094-1102):

```
#ifndef OPENSSL_NO_INDECT
if (strcmp(*argv,"indect-128-cbc") == 0) doit[D_CBC_128_IBC]=1;
else    if (strcmp(*argv,"indect-192-cbc") == 0) doit[D_CBC_192_IBC]=1;
else    if (strcmp(*argv,"indect-320-cbc") == 0) doit[D_CBC_320_IBC]=1;
else
#endif
...
#ifndef OPENSSL_NO_INDECT
if (strcmp(*argv,"indect") == 0)
{
doit[D_CBC_128_IBC]=1;
doit[D_CBC_192_IBC]=1;
doit[D_CBC_320_IBC]=1;
}
else
#endif
```

And error/usage message printing (lines 1237-1240 and 1284-1286):

```
#ifndef OPENSSL_NO_INDECT
BIO_printf(bio_err,"\n");
BIO_printf(bio_err,"indect-128-cbc indect-192-cbc indect-320-cbc ");
#endif
...
#ifndef OPENSSL_NO_INDECT
BIO_printf(bio_err,"indect ");
#endif
```

Next, the **speed** program prepares encryption keys for the performance test (lines 1396-1400):

```
#ifndef OPENSSL_NO_INDECT
    Indect_set_encrypt_key(key16,128,&indect_ks1);
    Indect_set_encrypt_key(ikey24,192,&indect_ks2);
    Indect_set_encrypt_key(ikey40,320,&indect_ks3);
#endif
```

And variables which stores performance results (lines 1467-1469 and 1507-1509):

```
    c[D_CBC_128_IBC][0]=count;
    c[D_CBC_192_IBC][0]=count;
    c[D_CBC_320_IBC][0]=count;
...

        c[D_CBC_128_IBC][i]=c[D_CBC_128_IBC][i-1]*l0/l1;
        c[D_CBC_192_IBC][i]=c[D_CBC_192_IBC][i-1]*l0/l1;
        c[D_CBC_320_IBC][i]=c[D_CBC_320_IBC][i-1]*l0/l1;
```

Finally, program executes the speed test (lines 1991-2035):

```
#ifndef OPENSSL_NO_INDECT
    if (doit[D_CBC_128_IBC])
        {
        for (j=0; j<SIZE_NUM; j++)
            {

print_message(names[D_CBC_128_IBC],c[D_CBC_128_IBC][j],lengths[j]);
            Time_F(START);
            for (count=0,run=1; COND(c[D_CBC_128_IBC][j]); count++)
                Indect_cbc_encrypt(buf,buf,
                        (unsigned long)lengths[j],&indect_ks1,
                        iv,INDECT_ENCRYPT);
            d=Time_F(STOP);
            print_result(D_CBC_128_IBC,j,count,d);
            }
        }
    if (doit[D_CBC_192_IBC])
        {
        for (j=0; j<SIZE_NUM; j++)
            {

print_message(names[D_CBC_192_IBC],c[D_CBC_192_IBC][j],lengths[j]);
            Time_F(START);
            for (count=0,run=1; COND(c[D_CBC_192_IBC][j]); count++)
                Indect_cbc_encrypt(buf,buf,
                        (unsigned long)lengths[j],&indect_ks2,
                        iv,INDECT_ENCRYPT);
            d=Time_F(STOP);
            print_result(D_CBC_192_IBC,j,count,d);
            }
        }
    if (doit[D_CBC_320_IBC])
        {
        for (j=0; j<SIZE_NUM; j++)
            {

print_message(names[D_CBC_320_IBC],c[D_CBC_320_IBC][j],lengths[j]);
            Time_F(START);
            for (count=0,run=1; COND(c[D_CBC_320_IBC][j]); count++)
```

```
                Indect_cbc_encrypt(buf,buf,
                        (unsigned long)lengths[j],&indect_ks3,
                        iv,INDECT_ENCRYPT);
            d=Time_F(STOP);
            print_result(D_CBC_320_IBC,j,count,d);
            }
        }

#endif
```

3.2 Modifications in crypto/evp/

The crypto/evp/ directory stores the code implementing EVP library. The EVP library provides a high-level interface to cryptographic functions. It requires several modifications in order to add new algorithm.

First, crypto/evp/c_allc.c file should be modified. It contains calls to functions registering all cipher algorithms. Calls registering the IBC algorithm need to be added (lines 225-250):

```
#ifndef OPENSSL_NO_INDECT
    EVP_add_cipher(EVP_indect_128_ecb());
    EVP_add_cipher(EVP_indect_128_cbc());
    EVP_add_cipher(EVP_indect_128_cfb());
    EVP_add_cipher(EVP_indect_128_cfb1());
    EVP_add_cipher(EVP_indect_128_cfb8());
    EVP_add_cipher(EVP_indect_128_ofb());
    EVP_add_cipher_alias(SN_indect_128_cbc,"INDECT128");
    EVP_add_cipher_alias(SN_indect_128_cbc,"indect128");
    EVP_add_cipher(EVP_indect_192_ecb());
    EVP_add_cipher(EVP_indect_192_cbc());
    EVP_add_cipher(EVP_indect_192_cfb());
    EVP_add_cipher(EVP_indect_192_cfb1());
    EVP_add_cipher(EVP_indect_192_cfb8());
    EVP_add_cipher(EVP_indect_192_ofb());
    EVP_add_cipher_alias(SN_indect_192_cbc,"INDECT192");
    EVP_add_cipher_alias(SN_indect_192_cbc,"indect192");
    EVP_add_cipher(EVP_indect_320_ecb());
    EVP_add_cipher(EVP_indect_320_cbc());
    EVP_add_cipher(EVP_indect_320_cfb());
    EVP_add_cipher(EVP_indect_320_cfb1());
    EVP_add_cipher(EVP_indect_320_cfb8());
    EVP_add_cipher(EVP_indect_320_ofb());
    EVP_add_cipher_alias(SN_indect_320_cbc,"INDECT320");
    EVP_add_cipher_alias(SN_indect_320_cbc,"indect320");
#endif
```

Second, the crypto/evp/evp.h header file should be modified. Value of constant variable defining the maximum key length should be modified (line 90). Originally it is 32 bytes (256 bits) – its needs to be changed to 40 bytes (320 bits) because IBC uses 320 bits key:

```
#define EVP_MAX_KEY_LENGTH      32 to #define EVP_MAX_KEY_LENGTH      40
```

Next modification required in crypto/evp/evp.h is to declare all IBC available modes (lines 818-840):

```
#ifndef OPENSSL_NO_INDECT
const EVP_CIPHER *EVP_indect_128_ecb(void);
const EVP_CIPHER *EVP_indect_128_cbc(void);
const EVP_CIPHER *EVP_indect_128_cfb1(void);
const EVP_CIPHER *EVP_indect_128_cfb8(void);
const EVP_CIPHER *EVP_indect_128_cfb128(void);
# define EVP_indect_128_cfb EVP_indect_128_cfb128
const EVP_CIPHER *EVP_indect_128_ofb(void);
const EVP_CIPHER *EVP_indect_192_ecb(void);
const EVP_CIPHER *EVP_indect_192_cbc(void);
const EVP_CIPHER *EVP_indect_192_cfb1(void);
const EVP_CIPHER *EVP_indect_192_cfb8(void);
const EVP_CIPHER *EVP_indect_192_cfb128(void);
# define EVP_indect_192_cfb EVP_indect_192_cfb128
const EVP_CIPHER *EVP_indect_192_ofb(void);
const EVP_CIPHER *EVP_indect_320_ecb(void);
const EVP_CIPHER *EVP_indect_320_cbc(void);
const EVP_CIPHER *EVP_indect_320_cfb1(void);
const EVP_CIPHER *EVP_indect_320_cfb8(void);
const EVP_CIPHER *EVP_indect_320_cfb128(void);
# define EVP_indect_320_cfb EVP_indect_320_cfb128
const EVP_CIPHER *EVP_indect_320_ofb(void);
#endif
```

crypto/evp/evp.h ends with error codes. It contains error codes related to IBC too. They are automatically generated by the script mkerr.pl and should not be modified manually. Similarly, the crypto/evp/evp_err.c file contains error codes. This file is also automatically generated, and should not be modified manually. In order to generate error codes in both files it is needed to call **make errors** command in the main source directory after applying all modifications in the source.

Next file required to modify is crypto/evp/evp_test.c. This file contains calls to the tests of registered ciphers. The call to disabling the test of IBC should be added (lines 427-433):

```
#ifdef OPENSSL_NO_INDECT
        if (strstr(cipher, "INDECT") == cipher)
        {
        fprintf(stdout, "Cipher disabled, skipping %s\n", cipher);
        continue;
        }
#endif
```

3.2.1 crypto/evp/e_indect.c

This file should be created manually. It contains calls from high level EVP functions to lower level functions implementing new algorithm, defined in the main header file (indect.h). The content of this file is listed below:

```
#include <openssl/opensslconf.h>
#ifndef OPENSSL_NO_INDECT
#include <openssl/evp.h>
#include <openssl/err.h>
#include <string.h>
#include <assert.h>
#include <openssl/indect.h>
#include "evp_locl.h"
```

```
static int indect_init_key(EVP_CIPHER_CTX *ctx, const unsigned char *key,
    const unsigned char *iv, int enc);

/* Indect subkey Structure */
typedef struct
    {
    INDECT_KEY ks;
    } EVP_INDECT_KEY;

/* Attribute operation for Indect */
#define data(ctx)   EVP_C_DATA(EVP_INDECT_KEY,ctx)

IMPLEMENT_BLOCK_CIPHER(indect_128, ks, Indect, EVP_INDECT_KEY,
    NID_indect_128, 16, 16, 16, 128,
    0, indect_init_key, NULL,
    EVP_CIPHER_set_asn1_iv,
    EVP_CIPHER_get_asn1_iv,
    NULL)
IMPLEMENT_BLOCK_CIPHER(indect_192, ks, Indect, EVP_INDECT_KEY,
    NID_indect_192, 16, 24, 16, 128,
    0, indect_init_key, NULL,
    EVP_CIPHER_set_asn1_iv,
    EVP_CIPHER_get_asn1_iv,
    NULL)
IMPLEMENT_BLOCK_CIPHER(indect_320, ks, Indect, EVP_INDECT_KEY,
    NID_indect_320, 16, 40, 16, 128,
    0, indect_init_key, NULL,
    EVP_CIPHER_set_asn1_iv,
    EVP_CIPHER_get_asn1_iv,
    NULL)

#define IMPLEMENT_INDECT_CFBR(ksize,cbits)
IMPLEMENT_CFBR(indect,Indect,EVP_INDECT_KEY,ks,ksize,cbits,32,0)

IMPLEMENT_INDECT_CFBR(128,1)
IMPLEMENT_INDECT_CFBR(192,1)
IMPLEMENT_INDECT_CFBR(320,1)

IMPLEMENT_INDECT_CFBR(128,8)
IMPLEMENT_INDECT_CFBR(192,8)
IMPLEMENT_INDECT_CFBR(320,8)



/* The subkey for Indect is generated. */
static int indect_init_key(EVP_CIPHER_CTX *ctx, const unsigned char *key,
    const unsigned char *iv, int enc)
    {
    int ret;

    if ((ctx->cipher->flags & EVP_CIPH_MODE) == EVP_CIPH_CFB_MODE
        || (ctx->cipher->flags & EVP_CIPH_MODE) == EVP_CIPH_OFB_MODE
        || enc)
        ret=Indect_set_encrypt_key(key, ctx->key_len * 8, ctx-
>cipher_data);
    else
        ret=Indect_set_decrypt_key(key, ctx->key_len * 8, ctx-
>cipher_data);

    if(ret < 0)
```

```
        {
        EVPerr(EVP_F_INDECT_INIT_KEY,EVP_R_INDECT_KEY_SETUP_FAILED);
        return 0;
        }

    return 1;
    }

#else
```

3.3 Modifications in crypto/objects/

The crypto/objects directory contains all information related to object model of the library. Files containing references to newly implemented IBC are obj_dat.h, obj_mac.c and obj_mac.num. These files are automatically generated by **make crypto/objects/obj_dat.h** command, basing on objects defined in objects.txt file.

objects.txt file contains definitions of all objects. The basic syntax for adding an object is as follows:

1 2 3 4 : shortName : Long Name

If the long name doesn't contain spaces, or no short name exists, the long name is used as basis for the base name in C. Otherwise, the short name is used.

Objects definitions containing references to IBC are located in objects.txt file (lines 1249-1280):

```
!Alias agh-indect 1 3 6 1 4 1 38980 666

agh-indect 1        : INDECT-128-ECB        : indect-128-ecb
agh-indect 2        : INDECT-128-CBC        : indect-128-cbc
!Cname indect-128-ofb128
agh-indect 3        : INDECT-128-OFB        : indect-128-ofb
!Cname indect-128-cfb128
agh-indect 4        : INDECT-128-CFB        : indect-128-cfb

agh-indect 21       : INDECT-192-ECB        : indect-192-ecb
agh-indect 22       : INDECT-192-CBC        : indect-192-cbc
!Cname indect-192-ofb128
agh-indect 23       : INDECT-192-OFB        : indect-192-ofb
!Cname indect-192-cfb128
agh-indect 24       : INDECT-192-CFB        : indect-192-cfb

agh-indect 41       : INDECT-320-ECB        : indect-320-ecb
agh-indect 42       : INDECT-320-CBC        : indect-320-cbc
!Cname indect-320-ofb128
agh-indect 43       : INDECT-320-OFB        : indect-320-ofb
!Cname indect-320-cfb128
agh-indect 44       : INDECT-320-CFB        : indect-320-cfb

# There are no OIDs for these Indect modes...

            : INDECT-128-CFB1       : indect-128-cfb1
            : INDECT-192-CFB1       : indect-192-cfb1
            : INDECT-320-CFB1       : indect-320-cfb1
```

```
: INDECT-128-CFB8       : indect-128-cfb8
: INDECT-192-CFB8       : indect-192-cfb8
: INDECT-320-CFB8       : indect-320-cfb8
```

It is important, that **make crypto/objects/obj_dat.h** command must be invoked after applying all source modification in order to update object definitions files from objects.txt.

3.4 Modifications in ssl/

In order to make new cipher available in SSL/TLS connections there must be several modifications applied in the *libssl* library. New ciphersuites, containing the new cipher must be created and registered.

Ciphersuites are named combinations of authentication, encryption, and message authentication code (MAC) algorithms used to negotiate the security settings for a network connection using the Transport Layer Security (TLS) or Secure Sockets Layer (SSL) network protocol. The structure and use of the ciphersuite concept is defined in the documents that define the protocol (RFC 5246 standard for TLS version 1.2).

When a TLS connection is established, a handshaking, known as the TLS Handshake Protocol, occurs. Within this handshake, a client hello (ClientHello) and a server hello (ServerHello) message are passed. First, the client sends a cipher suite list, a list of the cipher suites that it supports, in order of preference. Then the server replies with the cipher suite that it has selected from the client cipher suite list. [3]

The ciphersuites IDs are assigned by IANA and defined in several RFCs. There is a range of ciphersuites IDs reserved for private use (first byte of ciphersuite ID equals FF). The IBC ciphersuites use IDs from the private range. They are defined in the ssl/tls1.h header file (lines 233-247 and 363-376):

```
/* Indect ciphersuites from XXX (private) */
#define TLS1_CK_RSA_WITH_INDECT_128_CBC_SHA     0x0300FF41
#define TLS1_CK_DH_DSS_WITH_INDECT_128_CBC_SHA      0x0300FF42
#define TLS1_CK_DH_RSA_WITH_INDECT_128_CBC_SHA      0x0300FF43
#define TLS1_CK_DHE_DSS_WITH_INDECT_128_CBC_SHA       0x0300FF44
#define TLS1_CK_DHE_RSA_WITH_INDECT_128_CBC_SHA       0x0300FF45
#define TLS1_CK_ADH_WITH_INDECT_128_CBC_SHA     0x0300FF46

#define TLS1_CK_RSA_WITH_INDECT_320_CBC_SHA     0x0300FF84
#define TLS1_CK_DH_DSS_WITH_INDECT_320_CBC_SHA      0x0300FF85
#define TLS1_CK_DH_RSA_WITH_INDECT_320_CBC_SHA        0x0300FF86
#define TLS1_CK_DHE_DSS_WITH_INDECT_320_CBC_SHA       0x0300FF87
#define TLS1_CK_DHE_RSA_WITH_INDECT_320_CBC_SHA     0x0300FF88
#define TLS1_CK_ADH_WITH_INDECT_320_CBC_SHA     0x0300FF89

/* Indect ciphersuites from XXX (private) */
#define TLS1_TXT_RSA_WITH_INDECT_128_CBC_SHA        "INDECT128-SHA"
#define TLS1_TXT_DH_DSS_WITH_INDECT_128_CBC_SHA     "DH-DSS-INDECT128-SHA"
#define TLS1_TXT_DH_RSA_WITH_INDECT_128_CBC_SHA       "DH-RSA-INDECT128-
SHA"
#define TLS1_TXT_DHE_DSS_WITH_INDECT_128_CBC_SHA    "DHE-DSS-INDECT128-SHA"
#define TLS1_TXT_DHE_RSA_WITH_INDECT_128_CBC_SHA    "DHE-RSA-INDECT128-SHA"
#define TLS1_TXT_ADH_WITH_INDECT_128_CBC_SHA        "ADH-INDECT128-SHA"
```

```
#define TLS1_TXT_RSA_WITH_INDECT_320_CBC_SHA         "INDECT320-SHA"
#define TLS1_TXT_DH_DSS_WITH_INDECT_320_CBC_SHA      "DH-DSS-INDECT320-SHA"
#define TLS1_TXT_DH_RSA_WITH_INDECT_320_CBC_SHA        "DH-RSA-INDECT320-
SHA"
#define TLS1_TXT_DHE_DSS_WITH_INDECT_320_CBC_SHA     "DHE-DSS-INDECT320-SHA"
#define TLS1_TXT_DHE_RSA_WITH_INDECT_320_CBC_SHA     "DHE-RSA-INDECT320-SHA"
#define TLS1_TXT_ADH_WITH_INDECT_320_CBC_SHA         "ADH-INDECT320-SHA"
```

The ciphersuites defined above are implemented in the ssl/s3_lib.c file (lines 1606-1770). The structure implementing each ciphersuite looks like:

```
    /* Cipher FF41 */
    {
    1,
    TLS1_TXT_RSA_WITH_INDECT_128_CBC_SHA,
    TLS1_CK_RSA_WITH_INDECT_128_CBC_SHA,
    SSL_kRSA|SSL_aRSA|SSL_INDECT|SSL_SHA|SSL_TLSV1,
    SSL_NOT_EXP|SSL_HIGH,
    0,
    128,
    128,
    SSL_ALL_CIPHERS,
    SSL_ALL_STRENGTHS
    },
```

That structure contains all settings related to particular ciphersuite. There are 12 ciphersuites containing IBC cipher implemented.

Next file which needs to be modified is ssl/ssl.h header file. It contains several constant definitions. When implementing a new cipher, the constant specifying its name needs to be defined (line 289):

```
#define SSL_TXT_INDECT      "INDECT"
```

Furthermore, the ssl/ssl.h file contains `#define SSL_DEFAULT_CIPHER_LIST` constant. It defines the default ciphersuites order of preference. The cipher list consists of one or more cipher strings separated by colons. It can represent a list of cipher suites containing a certain algorithm, or cipher suites of a certain type. For example SHA1 represents all ciphers suites using the digest algorithm SHA1 and SSLv3 represents all SSL v3 algorithms

Each cipher string can be optionally preceded by the characters !, - or +. If **!** is used then the ciphers are permanently deleted from the list. The ciphers deleted can never reappear in the list even if they are explicitly stated.

If ! is used then the ciphers are permanently deleted from the list. The ciphers deleted can never reappear in the list even if they are explicitly stated.

If - is used then the ciphers are deleted from the list, but some or all of the ciphers can be added again by later options.

If + is used then the ciphers are moved to the end of the list. This option doesn't add any new ciphers it just moves matching existing ones.

If none of these characters is present then the string is just interpreted as a list of ciphers to be appended to the current preference list. If the list includes any ciphers already present they will be ignored: that is they will not be moved to the end of the list. Additionally the cipher string @STRENGTH can be used at any point to sort the current cipher list in order of encryption algorithm key length. [2]

The default cipher list constant is as follows (line 323):

```
#define SSL_DEFAULT_CIPHER_LIST "AES:ALL:!aNULL:!eNULL:+RC4:@STRENGTH"
```

This constant was not needed to be modified to enable preferred usage of IBC ciphersuites because the list is sorted in order of encryption algorithm key length. The IBC ciphersuite uses 320 bit key length, so it will be preferred over the AES ciphersuites (maximum 256 bit key length). It is worth to mention that modifying this constant results in breaking binary compatibility.

Next file which must be modified when implementing a new cipher is ssl/ssl_algs.c file. This file contains function `SSL_library_init(void)`. This function registers all ciphers and digests. Calls adding a new cipher need to be added (lines 91-94):

```
#ifndef OPENSSL_NO_INDECT
    EVP_add_cipher(EVP_indect_128_cbc());
    EVP_add_cipher(EVP_indect_320_cbc());
#endif
```

Several modifications must be applied in file ssl/ssl_ciph.c too. First, this file contains constants defining internal IDs for each encryption algorithm used in SSL/TLS. IDs for each key length for new algorithm need to be added (lines 136-137):

```
#define SSL_ENC_INDECT128_IDX   12
#define SSL_ENC_INDECT320_IDX   13
#define SSL_ENC_NUM_IDX     14
```

Moreover, `SSL_ENC_NUM_IDX` constant, which holds the number of all IDs needs to be incremented (from 12 to 14, because two IDs were added).

Next, the ssl/ssl_ciph.c file contains `SSL_CIPHER cipher_aliases[]` array. Alias for newly implemented algorithm needs to be added (line 207):

```
    {0,SSL_TXT_INDECT,0,SSL_INDECT, 0,0,0,0,SSL_ENC_MASK,0},
```

File contains also the implementation of `ssl_load_ciphers(void)` function. Calls loading a new cipher needs to be added in this function (lines 257-260):

```
    ssl_cipher_methods[SSL_ENC_INDECT128_IDX]=
      EVP_get_cipherbyname(SN_indect_128_cbc);
    ssl_cipher_methods[SSL_ENC_INDECT320_IDX]=
      EVP_get_cipherbyname(SN_indect_320_cbc);
```

Next the following code needs to be added in `ssl_cipher_get_evp()` function (lines 390-397):

```
    case SSL_INDECT:
        switch(c->alg_bits)
            {
        case 128: i=SSL_ENC_INDECT128_IDX; break;
        case 320: i=SSL_ENC_INDECT320_IDX; break;
        default: i=-1; break;
            }
        break;
```

This code part is responsible for selecting right key size. Similar code part needs to be added at the end of file, in function `SSL_CIPHER_description()` (lines 1238-1245). This function is responsible for printing information about use encryption algorithms.

```
    case SSL_INDECT:
        switch(cipher->strength_bits)
            {
        case 128: enc="Indect(128)"; break;
        case 320: enc="Indect(320)"; break;
        default: enc="Indect(?""?""?)"; break;
            }
        break;
```

The last file from *libssl* code which requires modifications is ssl/ssl_locl.h header file. Starting at line 249, this file contains definitions of bitmasks used to identify options by the library internals. Bitmask identifying a new cipher needs to be added (line 294):

```
#define SSL_INDECT              0x20000000L
```

It is important to update the overall encryption mask after adding a new mask. Encryption mask is defined at line 283:

```
#define SSL_ENC_MASK        0x1C3F8000L
```

It contains a mask which is a sum of the all masks defining the encryption algorithms. Because IBC mask is `0x20000000L`, the new encryption mask should be:

```
#define SSL_ENC_MASK        0x3C3F8000L
```

Adding new ciphers is limited by the mask space. After adding the IBC there is 2 bit left to go. Therefore, two more ciphers can be added.

3.5 Other modifications

After modifying the library it can be needed to update the version information. Version information is stored in crypto/opensslv.h heder file, lines 28-34. The line presented below is defining text constant storing the version information of modified library:

```
#define OPENSSL_VERSION_TEXT    "OpenSSL 0.9.8v 19 Apr 2012 + Indect 1.2 12 Aug 2012"
```

File util/libeay.num stores numbers of functions from *libcrypto* library. During adding a new cipher new functions were added. Therefore, it is needed to update this file. It is done automatically by command **make util/libeay.num**. This command should be invoked form the main directory after applying all modifications in the source.

Finally, the building scripts should be updated in order to properly build library with the new code. First, makefile.org should be updated (lines 138-146):

```
# dirs in crypto to build
SDIRS=  \
    objects \
    md2 md4 md5 sha mdc2 hmac ripemd \
    des aes rc2 rc4 rc5 idea bf cast camellia indect seed \
    bn ec rsa dsa ecdsa dh ecdh dso engine \
    buffer bio stack lhash rand err \
    evp asn1 pem x509 x509v3 conf txt_db pkcs7 pkcs12 comp ocsp ui krb5 \
    store cms pqueue jpake
```

Next, similar modification should be applied to the config script (lines 817-823):

```
for i in aes bf camellia indect cast des dh dsa ec hmac idea md2 md5 mdc2
rc2 rc4 rc5 ripemd rsa seed sha
do
  if [ ! -d crypto/$i ]
  then
    options="$options no-$i"
  fi
done
```

After applying all modifications presented in this chapter the new cipher should be available in the *libcrypto* functions as well in the SSL/TLS connection (*libssl* functions).

However, some repeating modifications were not described separately in the chapter. When implementing a new cipher refer to the diff file (Appendix C).

4. Implementation of IBC cipher

Implementations of particular cipher algorithms are part of the *libcrypto* library. Their code is located in *crypto/x* directories, where *x* is the name of individual algorithm. This chapter presents the content of the newly created *crypto/indect* directory, which contains definitions of IBC's functions and the code performing actual encryption and decryption. The code of IBC encryption/decryption was completely rewrited using fast, low-level C functions.

4.1 The main header file (indect.h)

Every cipher directory contains a main header file, named the same as a cipher directory. The main header file contains declarations of structures and functions. Names of functions consist of a prefix, usually the same as a name of the cipher. Functions prototypes are identical for every cipher, providing the interface to the underlying cipher.

In case of Indect Block Cipher, names of functions start with Indect_*.

```
int Indect_set_encrypt_key(const unsigned char *userKey, const int bits,
    INDECT_KEY *key);
int Indect_set_decrypt_key(const unsigned char *userKey, const int bits,
    INDECT_KEY *key);
```

The first two functions are responsible for setting up encryption/decryption options and generating internal keys based on settings and key provided by the user. These functions are implemented in *ibc_misc.c*. Functions take string, containing the encryption key provided by the user, as an input. The output of functions is stored in the `INDECT_KEY` type structure.

```
struct indect_key_st
    {
    unsigned char sbox[INDECT_SBOXES_MAXNR][256];
    unsigned char ptab[128][2];
    int bitLength;
    void (*enc)(const unsigned char *in, unsigned char *out, const unsigned
                char sbox[][256], const unsigned char ptab[][2]);
    void (*dec)(const unsigned char *in, unsigned char *out, const unsigned
                char sbox[][256], const unsigned char ptab[][2]);
    };
typedef struct indect_key_st INDECT_KEY;
```

This structure is cipher-dependent, so it can be different for every algorithm. It stores all the key information needed by the low level encryption/decryption functions. In case of Indect Block Cipher, the structure contains S-Boxes generated from key during the key initialization phase as well as permutation table, derived from P-Box. Because the IBC uses different encryption/decryption functions for each key length, the `INDECT_KEY` type structure stores also a pointer to the actual local low level encryption and decryption function. The pointer is set during the initialization phase.

```
void Indect_encrypt(const unsigned char *in, unsigned char *out,
                    const INDECT_KEY *key);
void Indect_decrypt(const unsigned char *in, unsigned char *out,
                    const INDECT_KEY *key);
```

`Indect_encrypt` and `Indect_decrypt` are functions called by upper level functions to encrypt a single block of data. They are implemented in *ibc_misc.c*. Because the IBC uses different encryption/decryption routines for each key length, these functions cannot directly implement block encryption/decryption. Instead, these functions invoke one of the actual encryption functions indirectly, by dereferencing function pointer argument stored in the key structure.

```
void Indect_encrypt(const unsigned char *in, unsigned char *out,
                    const INDECT_KEY *key)
    {
    key->enc(in, out, key->sbox, key->ptab);
    }
```

Next, the *indect.h* file contains declarations of functions implementing modes of operation. The source code of these functions is located in: *ibc_ecb.c*, *ibc_cbc.c*, *ibc_cfb.c*, *ibc_ofb.c* and *ibc_ctr.c*.

```
void Indect_ecb_encrypt(...);
void Indect_cbc_encrypt(...;
void Indect_cfb128_encrypt(...);
void Indect_cfb1_encrypt(...);
void Indect_cfb8_encrypt(...);
void Indect_cfbr_encrypt_block(...);
void Indect_ofb128_encrypt(...;
void Indect_ctr128_encrypt(...);
```

Functions do not encrypt/decrypt data itself. They only handle a data and prepare it for processing (partitioning into blocks, padding blocks). To encrypt/decrypt a single block they call `Indect_encrypt` and `Indect_decrypt` functions, described above. Thus, functions implementing block modes are mostly cipher invariant. In case of Indect Block Cipher, implementation of these functions has been borrowed from ciphers existing in OpenSSL and will not be discussed here.

4.2 Implementation of IBC (ibc_locl.h and indect.c)

The *ibc_locl.h* header file contains declarations of local functions, performing actual key setup and encryption/decryption. Functions interface does not depend on OpenSSL API. Thanks to that, the source code can be easily reused in other applications. Functions declared in *ibc_locl.c* are implemented in *indect.c* file.

4.2.1 Key setup

The first function is `indect_setup`. It takes a string containing key provided by the user as an input. The `const int bits` variable defines the key size and `const int enc` variable defines whether performing setup for encryption (1) or decryption (0).

```
void indect_setup(const unsigned char *key, const int bits, unsigned char
                  sbox[][256], unsigned char ptab[][2], const int enc);
```

The task of `indect_setup` function is to prepare key-based S-boxes and permutation table for the actual encryption/decryption functions.

```
sboxn = bits/64;
```

First, the function determines the total number of S-boxes to generate. This number depends on the actual key size. Each S-box is generated on base of 64 bits of the key. Next, it invokes a loop which generates one S-box in each iteration.

```
for (sboxi = 0; sboxi < sboxn; sboxi++) {
    ...
    rawkeylc = key[8*sboxi+0];
    mappedlc = (rawkeylc*255)/256;
    chosenlc[0] = chooselc(mappedlc, invalidlc_table);
    ...
```

In order to create a substitution box, it is required to use 8 LCs taken from the set of 255 possible ones. However these LCs must not be freely chosen, because it could result in a situation in which the matrix would lose its bijectivity. As it turns out, the linear combinations chosen to create a S-box have to be mutually non-linear.

At start of each iteration, the program reads the first 8 bits (1 byte) from the key. The first LC can be chosen from the full set (excluding the '0' linear combination). Therefore, the raw value (read from key) is mapped from 0-255 (256 values) range to 0-254 (255 values) range. The mapped number defines which LC **from set of right LCs** should be chosen. It is done by `chooselc` function.

```
unsigned char chooselc(unsigned char mappedlc, unsigned char
                       invalidlc_table[])
{
    unsigned char chosenlc;

    unsigned char currentlc = 0;
    int currentvalidlc = -1;

    while (1) {
        if (invalidlc_table[currentlc] == 0) ++currentvalidlc;
        if (currentvalidlc == mappedlc) {chosenlc = currentlc; break;}
        ++currentlc;
    }

    update_invalidlc_table(chosenlc, invalidlc_table);
    return chosenlc;

}
```

The function iterates through the set of all linear combinations and chooses the n-th right combination. **The numbering of right LCs starts with zero (the first right LC has number 0).** After choosing the combination, function calls `update_invalidlc_table`.

Function `update_invalidlc_table` iterates through the set of all linear combinations and set status "invalid" for the just chosen LC as well as for all LCs which could be created by XORing any number of already chosen LCs with selected one.

Next 7 LCs are chosen in similar way. The difference is that any next LC can be chosen from a smaller set of remaining combinations. Therefore, the raw number read from key is mapped to the smaller range, respectively: 0-253, 0-251, 0-247, 0-239, 0-223, 0-192 and 0-127 for the 8^{th} LC.

```
    ...
    rawkeylc = key[8*sboxi+7];
    mappedlc = (rawkeylc*128)/256;
    chosenlc[7] = chooselc(mappedlc, invalidlc_table);

    rawkeylc = key[8*sboxi+7];
    aes_inverted = rawkeylc & 1;
    ...
```

The last 8 bits from 64 bits defining each S-box determine the 8^{th} linear combination. It is worth to mention, that mapping from 0-255 to 0-127 effects in losing the value of last bit (dividing by two has the same effect as shifting right by 1 position). So, in fact, only 7 bits determine the linear combination. However, the last bit is used to determine whether use the standard AES S-box (0) or the inverted one (1) as a base for creating custom S-boxes.

```
    ...
    for (n = 0; n < 256; n++) {

        sbox[sboxi][n] = 0x00;

        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[0]) << 0;
        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[1]) << 1;
        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[2]) << 2;
        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[3]) << 3;
        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[4]) << 4;
        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[5]) << 5;
        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[6]) << 6;
        sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[7]) << 7;

    }
}
```

After finding all 8 linear combinations, the S-box is composed. Algorithm of S-boxes creating is described in D8.3 in detail and will not be discussed here. When S-box is composed, the loop goes to the next S-box, taking next 64 bits of key as an input.

After creating all S-boxes (which number depends on key length), the setup checks if it prepares them for decryption. If yes, generated S-boxes are inverted.

```
if (!enc)
{
    int n;
    unsigned char isbox[INDECT_SBOXES_MAXNR][256];
    for (sboxi = 0; sboxi < sboxn-1; sboxi++) for (n = 0; n < 256; n++)
                                        isbox[sboxi][sbox[sboxi][n]] = n;
    for (sboxi = 0; sboxi < sboxn-1; sboxi++) for (n = 0; n < 256; n++)
                                        sbox[sboxi][n] = isbox[sboxi][n];
}
```

The last S-box is not inverted. In fact, this is the P-box – it defines the permutation pattern. The P-box has size of 256, so it is feasible for operation on 256-bits data blocks. However, in

the OpenSSL implementation of IBC we were forced to use 128-bit blocks. It is due the binary compatibility issues. So, the P-box is first downsized to size of 128, and then it is (if applicable) inverted.

```
int newbit = 0;

for (bit=0; bit < 256; bit++) if (sbox[sboxn-1][bit] < 128)
{
    pbox[newbit] = sbox[sboxn-1][bit];
    newbit += 1;
}

if (!enc)
{
    int n;
    unsigned char ipbox[128];
    for (n = 0; n < 128; n++) ipbox[pbox[n]] = n;
    for (n = 0; n < 128; n++) pbox[n] = ipbox[n];
}
```

Finally, at the end of setup phase, the permutation table is generated. The permutation table specifies where each bit is moved during the permutation phase.

```
for (bit=0; bit < 128; bit++)
{
    ptab[bit][0] = pbox[bit]/8;
    ptab[bit][1] = pbox[bit]%8;
}

return;
```

The positions are calculated basing on the P-box. The first value stored in the table specifies into which byte should be particular bit moved. The second value is the remainder after dividing by eight. It specifies what is the exact position of bit in the destination byte (the amount of bit shift). The tests have shown that pre-calculating these values brings ~30% performance gain over calculating them each time during the permutation.

4.2.2 Encryption and decryption

There are 6 encryption and decryption functions defined in the *ibc_locl.h* header file. Each two of them corresponds to the different key length. Functions take string containing data as an input. They encrypt/decrypt the data and copy it into the memory location pointed by `unsigned char *out` argument.

Moreover, functions take S-boxes and permutation table prepared during the key setup as an input. They specify the encryption process. It is pretty straightforward.

```
unsigned char s[16];
memset(s,0,16);
for (bit=0; bit < 128; bit++)
        s[ptab[bit][0]] |=  ((in[bit/8] << (bit%8)) & 128) >> ptab[bit][1];
```

At start of encryption of each block, functions perform the initial permutation. First, it initializes `s[16]` variable, which stores the result of permutation, with zeros. Next, it invokes a loop. It processes one bit from the input string in each iteration. However, processor can only operate on bytes (groups of 8 bits). So, in fact, it reads 8 bits and masks the irrelevant bits rest out. Next, the bit is shifted for amount read from permutation table. After that, the whole byte (containing one bit read from input padded with zeros) is ORed in the proper position of block (which is read from permutation table too). This permutation process is repeated for every bit in 128-bit block.

```
for (round=0; round < 12; round++) {

    t[0] = sbox[0][s[0]];
    t[1] = sbox[0][s[1]];
    t[2] = sbox[0][s[2]];
    t[3] = sbox[0][s[3]];
    t[4] = sbox[1][s[4]];
    t[5] = sbox[1][s[5]];
    t[6] = sbox[1][s[6]];
    t[7] = sbox[1][s[7]];
    t[8] = sbox[2][s[8]];
    t[9] = sbox[2][s[9]];
    t[10] = sbox[2][s[10]];
    t[11] = sbox[2][s[11]];
    t[12] = sbox[3][s[12]];
    t[13] = sbox[3][s[13]];
    t[14] = sbox[3][s[14]];
    t[15] = sbox[3][s[15]];

    memset(s,0,16);
    for (bit=0; bit < 128; bit++)
         s[ptab[bit][0]] |=  ((t[bit/8] << (bit%8)) & 128) >> ptab[bit][1];
}
```

Next, the round loop starts. Number of rounds is different for each key size. For 128-bit key number of rounds equal 8. For 192-bit key it equals 10. For 320-bit key there are 12 rounds.

Each round consists of substitution and permutation. For substitution it uses S-boxes generated from key during the setup. Number of used S-boxes is different for each key size. For 128-bit key only one S-box is used. For 192-bit key there are two S-boxes. One half of the block (8 bytes) is substituted with one S-box, the second half with another. For 320-bit key there are four S-boxes used. Each S-box substitutes 4 bytes from block in that case.

For permutation it uses the same permutation table (generated from P-box) which is used in initial permutation. There is one permutation per round for each key size.

```
memcpy(out,s,16);
return;
}
```

After completing all rounds, function copies resulting ciphertext block into the memory location pointed by the `*out` pointer.

Decryption is a reversed algorithm of encryption. Decryption functions are built similar to the encryption functions and will not be discussed here.

5. Tests

This chapter presents various tests of IBC version of library. First, the compatibility test is presented. The test checks the backward binary compatibility of modified library with the original one. Next, the performance tests are presented. Performance of rewrited IBC code is compared with the performance of graphical application from D9.13. Finally, the usage of IBC cipher in SSL/TLS connections is tested.

5.1 Compatibility

OpenSSL is the most commonly used cryptographic library nowadays, especially in Linux environment. Also the OpenVPN toolkit uses OpenSSL as an cryptographic backend. Most applications use OpenSSL as a dynamic library.

In Windows environment libcrypto and libssl dynamic library (*.dll) files are usually distributed bundled with the application. The dll files are usually located in the application main directory. Their names are respectively *libeay32.dll* and *ssleay32.dll* for the 1.0 branch and *libeay32.dll* and *libssl32.dll* for the 0.9.8 branch.

In the Linux environment, however, most applications use system wide shared libraries, located in */lib* or similar system directory. Files are usually named *libcrypto.so.0.9.8* and *libssl.so.0.9.8* and respectively for the 1.0 branch.

One of the main reasons behind integrating the Indect Block Cipher into the OpenSSL toolkit was to enable the easiest possible way to apply IBC in existing applications. The most convenient way to do this would be to just replace existing shared library files with modified ones. However, it is possible only when the **binary compatibility** is ensured between the original and modified libraries.

Backward binary compatibility is the feature of new version of a library against an old version of the same library to guarantee that applications working with the old version keep working correctly with the new version **without recompilation**. Examples of breaking backward binary compatibility include changing data type size of a function parameter or changing the structure of a virtual table in a class. Such changes result in incorrect run-time behavior or even crash of applications that use the corresponding function or class. [4]

During the integration of IBC into the OpenSSL we put efforts toward keeping the binary interface compatible. Thanks to that, it is possible to deploy IBC on existing systems without any requirement to change code of applications using OpenSSL, or even without need for recompiling them against the modified library.

The binary compatibility has been tested with an automatic tool, called **abi-compliance-checker**. [5] It is a free, open source tool, available for Linux-like operating systems. The tool can check all the kinds of changes that cause backward binary compatibility problems. The test showed that full binary and source compatibility has been kept. Results of the test are presented in Appendix A.

5.2 Performance

Performance is a primary requirement for symmetric encryption nowadays. During the implementation of OpenSSL version of IBC we put efforts into optimizing the performance. Code of IBC was completely rewrited. Thanks that, we were able to gain a significant boost over the graphical application described in D9.13.

The first reason of such an improvement was migrating to the lower level programming language (C).

The second reason was rewriting the permutation code using only low-level bit instructions (bit shift, OR). It is worth to mention, that permutation is the most expensive operation in the algorithm. According to Amdahl's law, such large speedups can be achieved only by subsequently shaving off the largest cost factors in the task. As it was said earlier, only pre-calculating the amounts of bit shifts in permutation brings ~30% performance improvement.

Table 1: Performance comparison (3 GHz processor, one thread)

Key size	Rounds	Operation mode	Performance [Mbps]	
			GUI app	openssl implementation
128	8	CBC	2,709	63,624
192	10	CBC	2,370	52,122
320	12	CBC	2,116	44,122

5.3 SSL/TLS connectivity

New ciphersuites, containing IBC have been created in the *libssl* library. Thanks to that it is possible to use Indect Block Cipher to encrypt data transmitted in SSL/TLS connections.

The encryption algorithm used in a session is agreed between client and server during the handshake procedure. The client sends a Client Hello message to the server. Example of such a message is shown in **Figure 2**. The Client Hello message contains the list of cipher suites supported by the client. The list is ordered by preference (usually more secure ciphersuites are more preferred). IBC ciphersuites are identified as unknown by Wireshark software because its IDs are from the private range and Wireshark does not recognize them.

The server choses the first supported ciphersuite from the list, and sends its ID to the client in a Server Hello message. Example of a Server Hello message is show in **Figure 3**. In that case server chooses ciphersuite with ID 0xFF88. It corresponds to the `"DHE-RSA-INDECT320-SHA"` ciphersuite, which uses IBC cipher with 320 bits key to secure data transmitted between peers.

In order to successfully negotiate usage of IBC, both client and server must use modified version of library. If one of the peers use original SSL/TLS library (without IBC ciphersuites) the most preferred ciphersuite know to the both peers will be agreed (usually ciphersuite using AES algorithm). It means that it is possible to establish a secure connection between peers which are using modified and original libraries.

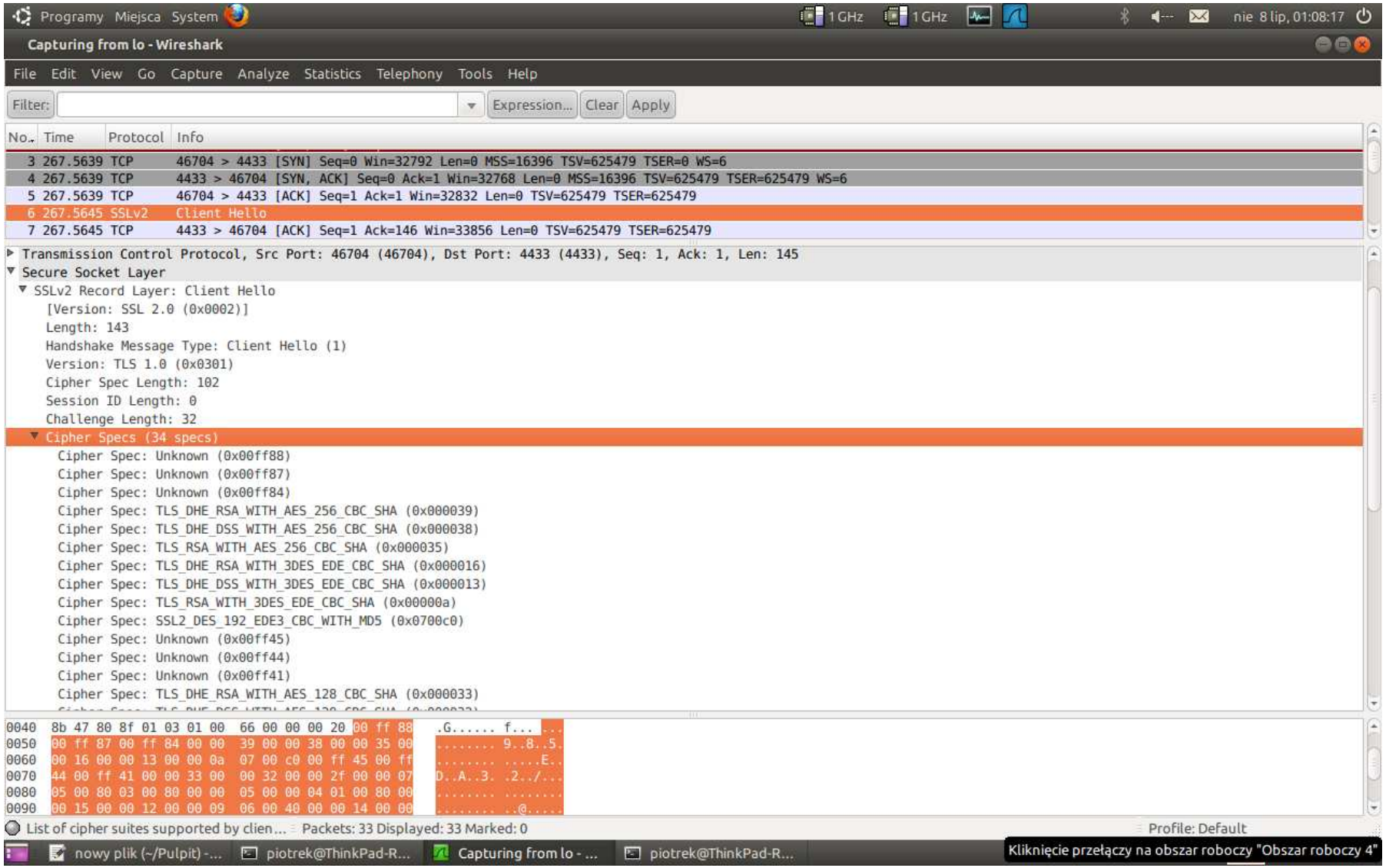

Figure 2: Client Hello message

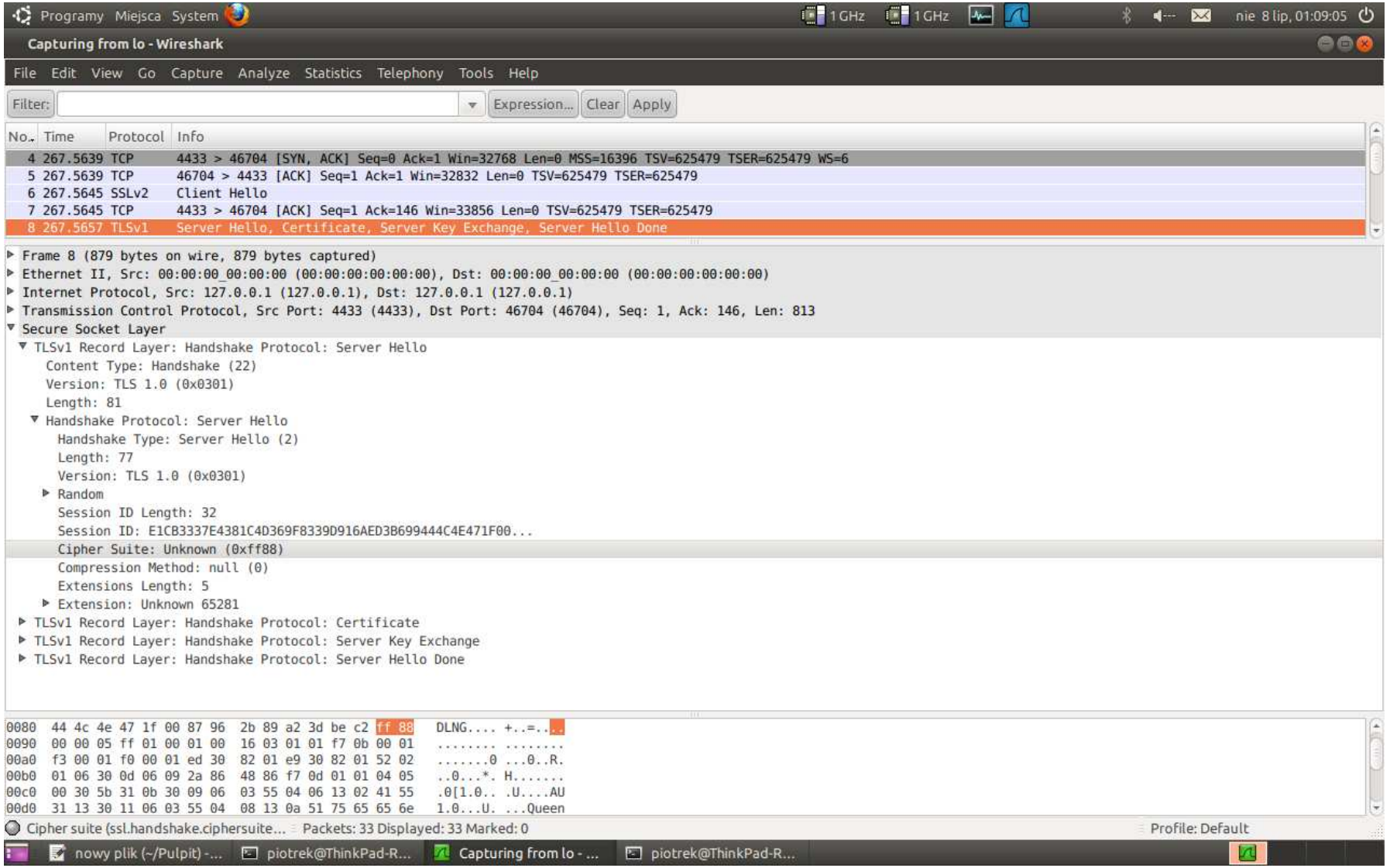

Figure 3: Server Hello message

6. Summary

Integrating a new cryptographic algorithm requires a number of modifications to the OpenSSL code. In particular, new SSL/TLS Ciphersuites, containing a new cipher, must be created, in order to make it available for usage in SSL/TLS protocol. All necesarly modifications are presented in Chapter 3. Overall, implementiation of IBC required more then 3000 changes in source code. The code modification statistics were presented in Appendix B.

The implementation of IBC algorithm itself was presented and discussed. The cipher code was completely rewrited using low-level C language. Thanks that, we were able to gain a significant performance boost over the graphical application described in D9.13. Results of performance tests are presented in Section 5.2. They show above 23x performance improvement.

Besides performance, compatibility requirements were important during the implementation. We were able to preserve full binary and source compatibility with the original version of OpenSSL. It is prooved in the compatibility report in Apeendix A. Thanks to that it is possible to use the most convienient way of deploying IBC libraries - replacing existing OpenSSL shared library files with modified ones.

Using the IBC enabled version of OpenSSL it is possible to establish secure connections over the Internet using Indect Block Cipher to encrypt transmitted data. If both client and server use modified library, the IBC is negotiated by default. SSL/TLS connectivity test is presented in Section 5.3.

Bibliography

1. OpenSSL Documents. *openssl.org.* [Online] http://www.openssl.org/docs/.

2. OpenSSL Documents: ciphers(1). *openssl.org.* [Online] http://www.openssl.org/docs/apps/ciphers.html.

3. Cipher suite. *wikipedia.org.* [Online] http://en.wikipedia.org/wiki/CipherSuite.

4. *Automated Verification of Shared Libraries for Backward Binary Compatibility.* Ponomarenko Andrey and Rubanov Vladimir. 2010. Proceedings of the 2010 Second International Conference on Advances in System Testing and Validation Lifecycle.

5. *ABI compliance checker.* [Online] http://ispras.linuxbase.org/index.php/ABI_compliance_checker.

Appendix A – Compatibility report

API compatibility report for the OpenSSL library between 0.9.8v and 0.9.8v + Indect 1.2 versions on x86

Binary Compatibility Source Compatibility

Test Info

Library Name	OpenSSL
Version #1	0.9.8v
Version #2	0.9.8v + Indect 1.2
CPU Type	x86
GCC Version	4.4.5
Subject	Binary Compatibility

Test Results

Total Header Files	2
Total Shared Libraries	2
Total Symbols / Types	2149 / 790
Verdict	**Compatible**

Problem Summary

	Severity	Count
Added Symbols	-	18
Removed Symbols	High	0
Problems with Data Types	High	0
	Medium	0
	Low	0
Problems with Symbols	High	0
	Medium	0
	Low	0
Problems with Constants	Low	0

Added Symbols (18)

evp.h, libcrypto.so.0.9.8
EVP_indect_128_cbc ( )

EVP_indect_128_cfb1 ( )
EVP_indect_128_cfb128 ( )
EVP_indect_128_cfb8 ( )
EVP_indect_128_ecb ( )
EVP_indect_128_ofb ( )
EVP_indect_192_cbc ( )
EVP_indect_192_cfb1 ( )
EVP_indect_192_cfb128 ( )
EVP_indect_192_cfb8 ( )
EVP_indect_192_ecb ( )
EVP_indect_192_ofb ( )
EVP_indect_320_cbc ( )
EVP_indect_320_cfb1 ( )
EVP_indect_320_cfb128 ( )
EVP_indect_320_cfb8 ( )
EVP_indect_320_ecb ( )
EVP_indect_320_ofb ( )

to the top

Header Files (2)

crypto.h
ssl.h

to the top

Shared Libraries (2)

libcrypto.so.0.9.8
libssl.so.0.9.8

to the top

Test Info

Library Name	OpenSSL
Version #1	0.9.8v
Version #2	0.9.8v + Indect 1.2
CPU Type	x86
GCC Version	4.4.5
Subject	Source Compatibility

Test Results

Total Header Files	2
Total Shared Libraries	2
Total Symbols / Types	2149 / 790
Verdict	**Compatible**

Problem Summary

	Severity	Count
Added Symbols	-	18
Removed Symbols	High	0
Problems with Data Types	High	0
	Medium	0
	Low	0
Problems with Symbols	High	0
	Medium	0
	Low	0
Problems with Constants	Low	0

Added Symbols (18)

evp.h
EVP_indect_128_cbc ( )
EVP_indect_128_cfb1 ( )
EVP_indect_128_cfb128 ( )
EVP_indect_128_cfb8 ( )
EVP_indect_128_ecb ( )
EVP_indect_128_ofb ( )
EVP_indect_192_cbc ( )
EVP_indect_192_cfb1 ( )
EVP_indect_192_cfb128 ( )
EVP_indect_192_cfb8 ( )
EVP_indect_192_ecb ( )

EVP_indect_192_ofb ( )
EVP_indect_320_cbc ( )
EVP_indect_320_cfb1 ( )
EVP_indect_320_cfb128 ( )
EVP_indect_320_cfb8 ( )
EVP_indect_320_ecb ( )
EVP_indect_320_ofb ( )

to the top

Header Files (2)

crypto.h
ssl.h

to the top

Shared Libraries (2)

libcrypto.so.0.9.8
libssl.so.0.9.8

to the top

Appendix B – Diffstat report

```
filename                     | number of changes

Makefile.org                 |    2
apps/cms.c                   |   12 +
apps/dsa.c                   |    7
apps/gendsa.c                |   12 +
apps/genrsa.c                |   12 +
apps/pkcs12.c                |    9
apps/progs.h                 |   18 +
apps/progs.pl                |    4
apps/rsa.c                   |    7
apps/smime.c                 |   12 +
apps/speed.c                 |  107 ++++++++-
config                       |    2
crypto/crypto-lib.com        |    6
crypto/evp/c_allc.c          |   27 ++
crypto/evp/evp.h             |   30 ++
crypto/evp/evp_err.c         |    4
crypto/evp/evp_test.c        |    7
crypto/evp/e_indect.c        |  136 ++++++++++
crypto/indect/ibc_cbc.c      |  132 ++++++++++
crypto/indect/ibc_cfb.c      |  227 ++++++++++++++++++
crypto/indect/ibc_ctr.c      |  143 ++++++++++
crypto/indect/ibc_ecb.c      |   74 ++++++
crypto/indect/ibc_locl.h     |  103 ++++++++
crypto/indect/ibc_misc.c     |  139 ++++++++++
crypto/indect/ibc_ofb.c      |  141 ++++++++++
crypto/indect/indect.c       |  529 ++++++++++++++++++++++++++++++++++++++++++++++
crypto/indect/indect.h       |  132 ++++++++++
crypto/install.com           |    3
crypto/objects/obj_dat.h     |  100 ++++++++
crypto/objects/obj_mac.h     |   86 +++++++
crypto/objects/obj_mac.num   |   18 +
crypto/objects/objects.txt   |   34 ++
crypto/openssl.h             |    4
include/openssl/evp.h        |   30 ++
include/openssl/indect.h     |  132 ++++++++++
include/openssl/obj_mac.h    |   86 +++++++
include/openssl/opensslv.h   |    4
include/openssl/ssl.h        |    1
include/openssl/tls1.h       |   30 ++
makevms.com                  |    5
ssl/s3_lib.c                 |  165 +++++++++++++
ssl/ssl.h                    |    1
ssl/ssl_algs.c               |    5
ssl/ssl_ciph.c               |   28 ++
ssl/ssl_locl.h               |    5
ssl/tls1.h                   |   30 ++
test/evp_test.c              |    7
util/libeay.num              |   30 ++
util/mk1mf.pl                |    8
util/mkdef.pl                |    7
util/mkfiles.pl              |    1

51 files changed, 2957 insertions(+), 33 deletions(-)
```

added files

Appendix C – Full diff file

```
diff -rupN openssl-0.9.8v-org//apps/cms.c openssl-0.9.8v//apps/cms.c
--- openssl-0.9.8v-org//apps/cms.c  2012-03-12 15:51:44.000000000 +0100
+++ openssl-0.9.8v//apps/cms.c  2012-05-09 23:23:06.000000000 +0200
@@ -226,6 +226,14 @@ int MAIN(int argc, char **argv)
        else if (!strcmp(*args,"-camellia256"))
                cipher = EVP_camellia_256_cbc();
 #endif
+#ifndef OPENSSL_NO_INDECT
+       else if (!strcmp(*args,"-indect128"))
+               cipher = EVP_indect_128_cbc();
+       else if (!strcmp(*args,"-indect192"))
+               cipher = EVP_indect_192_cbc();
+       else if (!strcmp(*args,"-indect320"))
+               cipher = EVP_indect_320_cbc();
+#endif
        else if (!strcmp (*args, "-debug_decrypt"))
                flags |= CMS_DEBUG_DECRYPT;
        else if (!strcmp (*args, "-text"))
@@ -602,6 +610,10 @@ int MAIN(int argc, char **argv)
        BIO_printf (bio_err, "-camellia128, -camellia192, -camellia256\n");
        BIO_printf (bio_err, "                encrypt PEM output with cbc camellia\n");
 #endif
+#ifndef OPENSSL_NO_INDECT
+       BIO_printf (bio_err, "-indect128, -indect192, -indect320\n");
+       BIO_printf (bio_err, "                encrypt PEM output with cbc indect\n");
+#endif
        BIO_printf (bio_err, "-nointern      don't search certificates in message for
signer\n");
        BIO_printf (bio_err, "-nosigs        don't verify message signature\n");
        BIO_printf (bio_err, "-noverify      don't verify signers certificate\n");
diff -rupN openssl-0.9.8v-org//apps/dsa.c openssl-0.9.8v//apps/dsa.c
--- openssl-0.9.8v-org//apps/dsa.c  2010-02-02 15:03:05.000000000 +0100
+++ openssl-0.9.8v//apps/dsa.c  2012-05-09 23:22:05.000000000 +0200
@@ -87,6 +87,9 @@
  * -camellia128 - encrypt output if PEM format
  * -camellia192 - encrypt output if PEM format
  * -camellia256 - encrypt output if PEM format
+ * -indect128 - encrypt output if PEM format
+ * -indect192 - encrypt output if PEM format
+ * -indect320 - encrypt output if PEM format
  * -seed        - encrypt output if PEM format
  * -text   - print a text version
  * -modulus    - print the DSA public key
@@ -218,6 +221,10 @@ bad:
        BIO_printf(bio_err," -camellia128, -camellia192, -camellia256\n");
        BIO_printf(bio_err,"                encrypt PEM output with cbc camellia\n");
 #endif
+#ifndef OPENSSL_NO_INDECT
+       BIO_printf(bio_err," -indect128, -indect192, -indect320\n");
+       BIO_printf(bio_err,"                encrypt PEM output with cbc indect\n");
+#endif
 #ifndef OPENSSL_NO_SEED
        BIO_printf(bio_err," -seed           encrypt PEM output with cbc seed\n");
 #endif
diff -rupN openssl-0.9.8v-org//apps/gendsa.c openssl-0.9.8v//apps/gendsa.c
--- openssl-0.9.8v-org//apps/gendsa.c   2010-06-12 15:18:44.000000000 +0200
+++ openssl-0.9.8v//apps/gendsa.c   2012-05-09 23:34:23.000000000 +0200
@@ -157,6 +157,14 @@ int MAIN(int argc, char **argv)
        else if (strcmp(*argv,"-camellia256") == 0)
            enc=EVP_camellia_256_cbc();
 #endif
+#ifndef OPENSSL_NO_INDECT
+       else if (strcmp(*argv,"-indect128") == 0)
+           enc=EVP_indect_128_cbc();
+       else if (strcmp(*argv,"-indect192") == 0)
+           enc=EVP_indect_192_cbc();
+       else if (strcmp(*argv,"-indect320") == 0)
+           enc=EVP_indect_320_cbc();
+#endif
        else if (**argv != '-' && dsaparams == NULL)
            {
            dsaparams = *argv;
```

```
@@ -191,6 +199,10 @@ bad:
        BIO_printf(bio_err," -camellia128, -camellia192, -camellia256\n");
        BIO_printf(bio_err,"                encrypt PEM output with cbc camellia\n");
 #endif
+#ifndef OPENSSL_NO_INDECT
+       BIO_printf(bio_err," -indect128, -indect192, -indect320\n");
+       BIO_printf(bio_err,"                encrypt PEM output with cbc indect\n");
+#endif
 #ifndef OPENSSL_NO_ENGINE
        BIO_printf(bio_err," -engine e - use engine e, possibly a hardware device.\n");
 #endif
diff -rupN openssl-0.9.8v-org//apps/genrsa.c openssl-0.9.8v//apps/genrsa.c
--- openssl-0.9.8v-org//apps/genrsa.c   2010-06-12 15:18:45.000000000 +0200
+++ openssl-0.9.8v//apps/genrsa.c   2012-05-09 23:24:41.000000000 +0200
@@ -180,6 +180,14 @@ int MAIN(int argc, char **argv)
        else if (strcmp(*argv,"-camellia256") == 0)
            enc=EVP_camellia_256_cbc();
 #endif
+#ifndef OPENSSL_NO_INDECT
+       else if (strcmp(*argv,"-indect128") == 0)
+           enc=EVP_indect_128_cbc();
+       else if (strcmp(*argv,"-indect192") == 0)
+           enc=EVP_indect_192_cbc();
+       else if (strcmp(*argv,"-indect320") == 0)
+           enc=EVP_indect_320_cbc();
+#endif
        else if (strcmp(*argv,"-passout") == 0)
            {
            if (--argc < 1) goto bad;
@@ -211,6 +219,10 @@ bad:
        BIO_printf(bio_err," -camellia128, -camellia192, -camellia256\n");
        BIO_printf(bio_err,"                encrypt PEM output with cbc camellia\n");
 #endif
+#ifndef OPENSSL_NO_INDECT
+       BIO_printf(bio_err," -indect128, -indect192, -indect320\n");
+       BIO_printf(bio_err,"                encrypt PEM output with cbc indect\n");
+#endif
        BIO_printf(bio_err," -out file       output the key to 'file\n");
        BIO_printf(bio_err," -passout arg    output file pass phrase source\n");
        BIO_printf(bio_err," -f4             use F4 (0x10001) for the E value\n");
diff -rupN openssl-0.9.8v-org//apps/pkcs12.c openssl-0.9.8v//apps/pkcs12.c
--- openssl-0.9.8v-org//apps/pkcs12.c   2011-03-13 19:23:24.000000000 +0100
+++ openssl-0.9.8v//apps/pkcs12.c   2012-05-09 23:23:33.000000000 +0200
@@ -184,6 +184,11 @@ int MAIN(int argc, char **argv)
        else if (!strcmp(*args,"-camellia192")) enc=EVP_camellia_192_cbc();
        else if (!strcmp(*args,"-camellia256")) enc=EVP_camellia_256_cbc();
 #endif
+#ifndef OPENSSL_NO_INDECT
+       else if (!strcmp(*args,"-indect128")) enc=EVP_indect_128_cbc();
+       else if (!strcmp(*args,"-indect192")) enc=EVP_indect_192_cbc();
+       else if (!strcmp(*args,"-indect320")) enc=EVP_indect_320_cbc();
+#endif
        else if (!strcmp (*args, "-noiter")) iter = 1;
        else if (!strcmp (*args, "-maciter"))
                    maciter = PKCS12_DEFAULT_ITER;
@@ -336,6 +341,10 @@ int MAIN(int argc, char **argv)
    BIO_printf (bio_err, "-camellia128, -camellia192, -camellia256\n");
    BIO_printf (bio_err, "              encrypt PEM output with cbc camellia\n");
 #endif
+#ifndef OPENSSL_NO_INDECT
+   BIO_printf (bio_err, "-indect128, -indect192, -indect320\n");
+   BIO_printf (bio_err, "              encrypt PEM output with cbc indect\n");
+#endif
    BIO_printf (bio_err, "-nodes        don't encrypt private keys\n");
    BIO_printf (bio_err, "-noiter       don't use encryption iteration\n");
    BIO_printf (bio_err, "-maciter      use MAC iteration\n");
diff -rupN openssl-0.9.8v-org//apps/progs.h openssl-0.9.8v//apps/progs.h
--- openssl-0.9.8v-org//apps/progs.h    2008-04-04 01:03:41.000000000 +0200
+++ openssl-0.9.8v//apps/progs.h    2012-05-09 23:30:13.000000000 +0200
@@ -188,6 +188,24 @@ FUNCTION functions[] = {
 #ifndef OPENSSL_NO_CAMELLIA
    {FUNC_TYPE_CIPHER,"camellia-256-ecb",enc_main},
 #endif
+#ifndef OPENSSL_NO_INDECT
+   {FUNC_TYPE_CIPHER,"indect-128-cbc",enc_main},
+#endif
+#ifndef OPENSSL_NO_INDECT
```

```
+   {FUNC_TYPE_CIPHER,"indect-128-ecb",enc_main},
+#endif
+#ifndef OPENSSL_NO_INDECT
+   {FUNC_TYPE_CIPHER,"indect-192-cbc",enc_main},
+#endif
+#ifndef OPENSSL_NO_INDECT
+   {FUNC_TYPE_CIPHER,"indect-192-ecb",enc_main},
+#endif
+#ifndef OPENSSL_NO_INDECT
+   {FUNC_TYPE_CIPHER,"indect-320-cbc",enc_main},
+#endif
+#ifndef OPENSSL_NO_INDECT
+   {FUNC_TYPE_CIPHER,"indect-320-ecb",enc_main},
+#endif
    {FUNC_TYPE_CIPHER,"base64",enc_main},
 #ifndef OPENSSL_NO_DES
    {FUNC_TYPE_CIPHER,"des",enc_main},
diff -rupN openssl-0.9.8v-org//apps/progs.pl openssl-0.9.8v//apps/progs.pl
--- openssl-0.9.8v-org//apps/progs.pl   2008-04-04 01:03:41.000000000 +0200
+++ openssl-0.9.8v//apps/progs.pl   2012-05-09 23:24:58.000000000 +0200
@@ -62,6 +62,9 @@ foreach (
    "camellia-128-cbc", "camellia-128-ecb",
    "camellia-192-cbc", "camellia-192-ecb",
    "camellia-256-cbc", "camellia-256-ecb",
+   "indect-128-cbc", "indect-128-ecb",
+   "indect-192-cbc", "indect-192-ecb",
+   "indect-320-cbc", "indect-320-ecb",
    "base64",
    "des", "des3", "desx", "idea", "seed", "rc4", "rc4-40",
    "rc2", "bf", "cast", "rc5",
@@ -82,6 +85,7 @@ foreach (
    if    ($_ =~ /des/)  { $t="#ifndef OPENSSL_NO_DES\n${t}#endif\n"; }
    elsif ($_ =~ /aes/)  { $t="#ifndef OPENSSL_NO_AES\n${t}#endif\n"; }
    elsif ($_ =~ /camellia/)  { $t="#ifndef OPENSSL_NO_CAMELLIA\n${t}#endif\n"; }
+   elsif ($_ =~ /indect/)  { $t="#ifndef OPENSSL_NO_INDECT\n${t}#endif\n"; }
    elsif ($_ =~ /idea/) { $t="#ifndef OPENSSL_NO_IDEA\n${t}#endif\n"; }
    elsif ($_ =~ /seed/) { $t="#ifndef OPENSSL_NO_SEED\n${t}#endif\n"; }
    elsif ($_ =~ /rc4/)  { $t="#ifndef OPENSSL_NO_RC4\n${t}#endif\n"; }
diff -rupN openssl-0.9.8v-org//apps/rsa.c openssl-0.9.8v//apps/rsa.c
--- openssl-0.9.8v-org//apps/rsa.c  2007-04-24 01:49:57.000000000 +0200
+++ openssl-0.9.8v//apps/rsa.c  2012-05-09 23:22:43.000000000 +0200
@@ -88,6 +88,9 @@
  * -camellia128 - encrypt output if PEM format
  * -camellia192 - encrypt output if PEM format
  * -camellia256 - encrypt output if PEM format
+ * -indect128 - encrypt output if PEM format
+ * -indect192 - encrypt output if PEM format
+ * -indect320 - encrypt output if PEM format
  * -text   - print a text version
  * -modulus    - print the RSA key modulus
  * -check  - verify key consistency
@@ -223,6 +226,10 @@ bad:
        BIO_printf(bio_err," -camellia128, -camellia192, -camellia256\n");
        BIO_printf(bio_err,"                encrypt PEM output with cbc camellia\n");
 #endif
+#ifndef OPENSSL_NO_INDECT
+       BIO_printf(bio_err," -indect128, -indect192, -indect320\n");
+       BIO_printf(bio_err,"                encrypt PEM output with cbc indect\n");
+#endif
        BIO_printf(bio_err," -text           print the key in text\n");
        BIO_printf(bio_err," -noout          don't print key out\n");
        BIO_printf(bio_err," -modulus        print the RSA key modulus\n");
diff -rupN openssl-0.9.8v-org//apps/smime.c openssl-0.9.8v//apps/smime.c
--- openssl-0.9.8v-org//apps/smime.c    2008-11-05 19:36:35.000000000 +0100
+++ openssl-0.9.8v//apps/smime.c    2012-05-09 23:24:21.000000000 +0200
@@ -173,6 +173,14 @@ int MAIN(int argc, char **argv)
        else if (!strcmp(*args,"-camellia256"))
                cipher = EVP_camellia_256_cbc();
 #endif
+#ifndef OPENSSL_NO_INDECT
+       else if (!strcmp(*args,"-indect128"))
+               cipher = EVP_indect_128_cbc();
+       else if (!strcmp(*args,"-indect192"))
+               cipher = EVP_indect_192_cbc();
+       else if (!strcmp(*args,"-indect320"))
+               cipher = EVP_indect_320_cbc();
+#endif
```

```
        else if (!strcmp (*args, "-text"))
                flags |= PKCS7_TEXT;
        else if (!strcmp (*args, "-nointern"))
@@ -443,6 +451,10 @@ int MAIN(int argc, char **argv)
        BIO_printf (bio_err, "-camellia128, -camellia192, -camellia256\n");
        BIO_printf (bio_err, "                encrypt PEM output with cbc camellia\n");
 #endif
+#ifndef OPENSSL_NO_INDECT
+       BIO_printf (bio_err, "-indect128, -indect192, -indect320\n");
+       BIO_printf (bio_err, "                encrypt PEM output with cbc indect\n");
+#endif
        BIO_printf (bio_err, "-nointern      don't search certificates in message for
signer\n");
        BIO_printf (bio_err, "-nosigs        don't verify message signature\n");
        BIO_printf (bio_err, "-noverify      don't verify signers certificate\n");
diff -rupN openssl-0.9.8v-org//apps/speed.c openssl-0.9.8v//apps/speed.c
--- openssl-0.9.8v-org//apps/speed.c    2010-07-08 03:23:25.000000000 +0200
+++ openssl-0.9.8v//apps/speed.c    2012-05-09 23:36:06.000000000 +0200
@@ -167,6 +167,9 @@
 #ifndef OPENSSL_NO_CAMELLIA
 #include <openssl/camellia.h>
 #endif
+#ifndef OPENSSL_NO_INDECT
+#include <openssl/indect.h>
+#endif
 #ifndef OPENSSL_NO_MD2
 #include <openssl/md2.h>
 #endif
@@ -285,7 +288,7 @@ static void print_result(int alg,int run
 static int do_multi(int multi);
 #endif

-#define ALGOR_NUM  28
+#define ALGOR_NUM  31
 #define SIZE_NUM   5
 #define RSA_NUM        4
 #define DSA_NUM        3
@@ -300,7 +303,8 @@ static const char *names[ALGOR_NUM]={
   "aes-128 cbc","aes-192 cbc","aes-256 cbc",
   "camellia-128 cbc","camellia-192 cbc","camellia-256 cbc",
   "evp","sha256","sha512",
-  "aes-128 ige","aes-192 ige","aes-256 ige"};
+  "aes-128 ige","aes-192 ige","aes-256 ige",
+  "indect-128 cbc","indect-192 cbc","indect-320 cbc"};
 static double results[ALGOR_NUM][SIZE_NUM];
 static int lengths[SIZE_NUM]={16,64,256,1024,8*1024};
 #ifndef OPENSSL_NO_RSA
@@ -582,6 +586,18 @@ int MAIN(int argc, char **argv)
         0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34,
         0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34,0x56};
 #endif
+#ifndef OPENSSL_NO_INDECT
+   static const unsigned char ikey24[24]=
+       {0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,
+        0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,
+        0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34};
+   static const unsigned char ikey40[40]=
+       {0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,
+        0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,
+        0x56,0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34,
+        0x12,0x34,0x56,0x78,0x9a,0xbc,0xde,0xf0,
+        0x78,0x9a,0xbc,0xde,0xf0,0x12,0x34,0x56};
+#endif
 #ifndef OPENSSL_NO_AES
 #define MAX_BLOCK_SIZE 128
 #else
@@ -603,6 +619,9 @@ int MAIN(int argc, char **argv)
 #ifndef OPENSSL_NO_CAMELLIA
    CAMELLIA_KEY camellia_ks1, camellia_ks2, camellia_ks3;
 #endif
+#ifndef OPENSSL_NO_INDECT
+   INDECT_KEY indect_ks1, indect_ks2, indect_ks3;
+#endif
 #define    D_MD2       0
 #define    D_MDC2      1
 #define    D_MD4       2
@@ -631,6 +650,9 @@ int MAIN(int argc, char **argv)
```

```
 #define D_IGE_128_AES   25
 #define D_IGE_192_AES   26
 #define D_IGE_256_AES   27
+#define D_CBC_128_IBC   28
+#define D_CBC_192_IBC   29
+#define D_CBC_320_IBC   30
     double d=0.0;
     long c[ALGOR_NUM][SIZE_NUM];
 #define    R_DSA_512   0
@@ -979,6 +1001,12 @@ int MAIN(int argc, char **argv)
         else    if (strcmp(*argv,"camellia-256-cbc") == 0) doit[D_CBC_256_CML]=1;
         else
 #endif
+#ifndef OPENSSL_NO_INDECT
+           if (strcmp(*argv,"indect-128-cbc") == 0) doit[D_CBC_128_IBC]=1;
+       else    if (strcmp(*argv,"indect-192-cbc") == 0) doit[D_CBC_192_IBC]=1;
+       else    if (strcmp(*argv,"indect-320-cbc") == 0) doit[D_CBC_320_IBC]=1;
+       else
+#endif
 #ifndef OPENSSL_NO_RSA
 #if 0 /* was: #ifdef RSAref */
            if (strcmp(*argv,"rsaref") == 0)
@@ -1063,6 +1091,15 @@ int MAIN(int argc, char **argv)
            }
         else
 #endif
+#ifndef OPENSSL_NO_INDECT
+           if (strcmp(*argv,"indect") == 0)
+           {
+           doit[D_CBC_128_IBC]=1;
+           doit[D_CBC_192_IBC]=1;
+           doit[D_CBC_320_IBC]=1;
+           }
+       else
+#endif
 #ifndef OPENSSL_NO_RSA
            if (strcmp(*argv,"rsa") == 0)
            {
@@ -1197,6 +1234,10 @@ int MAIN(int argc, char **argv)
           BIO_printf(bio_err,"\n");
           BIO_printf(bio_err,"camellia-128-cbc camellia-192-cbc camellia-256-cbc ");
 #endif
+#ifndef OPENSSL_NO_INDECT
+           BIO_printf(bio_err,"\n");
+           BIO_printf(bio_err,"indect-128-cbc indect-192-cbc indect-320-cbc ");
+#endif
 #ifndef OPENSSL_NO_RC4
           BIO_printf(bio_err,"rc4");
 #endif
@@ -1240,6 +1281,9 @@ int MAIN(int argc, char **argv)
 #ifndef OPENSSL_NO_CAMELLIA
           BIO_printf(bio_err,"camellia ");
 #endif
+#ifndef OPENSSL_NO_INDECT
+           BIO_printf(bio_err,"indect ");
+#endif
 #ifndef OPENSSL_NO_RSA
           BIO_printf(bio_err,"rsa      ");
 #endif
@@ -1249,7 +1293,8 @@ int MAIN(int argc, char **argv)
     !defined(OPENSSL_NO_RC2) || !defined(OPENSSL_NO_DES) || \
     !defined(OPENSSL_NO_RSA) || !defined(OPENSSL_NO_BF) || \
-    !defined(OPENSSL_NO_AES) || !defined(OPENSSL_NO_CAMELLIA)
+    !defined(OPENSSL_NO_AES) || !defined(OPENSSL_NO_CAMELLIA) || \
+    !defined(OPENSSL_NO_INDECT)
           BIO_printf(bio_err,"\n");
 #endif

@@ -1348,6 +1393,11 @@ int MAIN(int argc, char **argv)
     Camellia_set_key(ckey24,192,&camellia_ks2);
     Camellia_set_key(ckey32,256,&camellia_ks3);
 #endif
+#ifndef OPENSSL_NO_INDECT
+   Indect_set_encrypt_key(key16,128,&indect_ks1);
+   Indect_set_encrypt_key(ikey24,192,&indect_ks2);
+   Indect_set_encrypt_key(ikey40,320,&indect_ks3);
```

```
+#endif
 #ifndef OPENSSL_NO_IDEA
    idea_set_encrypt_key(key16,&idea_ks);
 #endif
@@ -1414,6 +1464,9 @@ int MAIN(int argc, char **argv)
    c[D_IGE_128_AES][0]=count;
    c[D_IGE_192_AES][0]=count;
    c[D_IGE_256_AES][0]=count;
+   c[D_CBC_128_IBC][0]=count;
+   c[D_CBC_192_IBC][0]=count;
+   c[D_CBC_320_IBC][0]=count;

    for (i=1; i<SIZE_NUM; i++)
        {
@@ -1451,6 +1504,9 @@ int MAIN(int argc, char **argv)
        c[D_IGE_128_AES][i]=c[D_IGE_128_AES][i-1]*l0/l1;
        c[D_IGE_192_AES][i]=c[D_IGE_192_AES][i-1]*l0/l1;
        c[D_IGE_256_AES][i]=c[D_IGE_256_AES][i-1]*l0/l1;
+       c[D_CBC_128_IBC][i]=c[D_CBC_128_IBC][i-1]*l0/l1;
+       c[D_CBC_192_IBC][i]=c[D_CBC_192_IBC][i-1]*l0/l1;
+       c[D_CBC_320_IBC][i]=c[D_CBC_320_IBC][i-1]*l0/l1;
        }
 #ifndef OPENSSL_NO_RSA
    rsa_c[R_RSA_512][0]=count/2000;
@@ -1931,6 +1987,51 @@ int MAIN(int argc, char **argv)
            }
        }

+#endif
+#ifndef OPENSSL_NO_INDECT
+   if (doit[D_CBC_128_IBC])
+       {
+       for (j=0; j<SIZE_NUM; j++)
+           {
+           print_message(names[D_CBC_128_IBC],c[D_CBC_128_IBC][j],lengths[j]);
+           Time_F(START);
+           for (count=0,run=1; COND(c[D_CBC_128_IBC][j]); count++)
+               Indect_cbc_encrypt(buf,buf,
+                       (unsigned long)lengths[j],&indect_ks1,
+                       iv,INDECT_ENCRYPT);
+           d=Time_F(STOP);
+           print_result(D_CBC_128_IBC,j,count,d);
+           }
+       }
+   if (doit[D_CBC_192_IBC])
+       {
+       for (j=0; j<SIZE_NUM; j++)
+           {
+           print_message(names[D_CBC_192_IBC],c[D_CBC_192_IBC][j],lengths[j]);
+           Time_F(START);
+           for (count=0,run=1; COND(c[D_CBC_192_IBC][j]); count++)
+               Indect_cbc_encrypt(buf,buf,
+                       (unsigned long)lengths[j],&indect_ks2,
+                       iv,INDECT_ENCRYPT);
+           d=Time_F(STOP);
+           print_result(D_CBC_192_IBC,j,count,d);
+           }
+       }
+   if (doit[D_CBC_320_IBC])
+       {
+       for (j=0; j<SIZE_NUM; j++)
+           {
+           print_message(names[D_CBC_320_IBC],c[D_CBC_320_IBC][j],lengths[j]);
+           Time_F(START);
+           for (count=0,run=1; COND(c[D_CBC_320_IBC][j]); count++)
+               Indect_cbc_encrypt(buf,buf,
+                       (unsigned long)lengths[j],&indect_ks3,
+                       iv,INDECT_ENCRYPT);
+           d=Time_F(STOP);
+           print_result(D_CBC_320_IBC,j,count,d);
+           }
+       }
+
 #endif
 #ifndef OPENSSL_NO_IDEA
    if (doit[D_CBC_IDEA])
diff -rupN openssl-0.9.8v-org//config openssl-0.9.8v//config
```

```
--- openssl-0.9.8v-org//config  2011-07-15 21:59:31.000000000 +0200
+++ openssl-0.9.8v//config  2012-04-20 20:43:36.000000000 +0200
@@ -814,7 +814,7 @@ case "$GUESSOS" in
   i386-*) options="$options 386" ;;
 esac

-for i in aes bf camellia cast des dh dsa ec hmac idea md2 md5 mdc2 rc2 rc4 rc5 ripemd rsa
seed sha
+for i in aes bf camellia indect cast des dh dsa ec hmac idea md2 md5 mdc2 rc2 rc4 rc5 ripemd
rsa seed sha
 do
   if [ ! -d crypto/$i ]
   then
diff -rupN openssl-0.9.8v-org//crypto/crypto-lib.com openssl-0.9.8v//crypto/crypto-lib.com
--- openssl-0.9.8v-org//crypto/crypto-lib.com   2010-03-25 15:45:22.000000000 +0100
+++ openssl-0.9.8v//crypto/crypto-lib.com   2012-04-20 20:48:34.000000000 +0200
@@ -83,7 +83,7 @@ $!
 $ ENCRYPT_TYPES = "Basic,"+ -
          "OBJECTS,"+ -
          "MD2,MD4,MD5,SHA,MDC2,HMAC,RIPEMD,"+ -
-         "DES,RC2,RC4,RC5,IDEA,BF,CAST,CAMELLIA,SEED,"+ -
+         "DES,RC2,RC4,RC5,IDEA,BF,CAST,CAMELLIA,INDECT,SEED,"+ -
          "BN,EC,RSA,DSA,ECDSA,DH,ECDH,DSO,ENGINE,AES,"+ -
          "BUFFER,BIO,STACK,LHASH,RAND,ERR,"+ -
          "EVP,EVP_2,ASN1,ASN1_2,PEM,X509,X509V3,"+ -
@@ -189,6 +189,8 @@ $ LIB_BF = "bf_skey,bf_ecb,bf_enc,bf_cfb
 $ LIB_CAST = "c_skey,c_ecb,c_enc,c_cfb64,c_ofb64"
 $ LIB_CAMELLIA = "camellia,cmll_misc,cmll_ecb,cmll_cbc,cmll_ofb,"+ -
    "cmll_cfb,cmll_ctr"
+$ LIB_INDECT = "indect,ibc_misc,ibc_ecb,ibc_cbc,ibc_ofb,"+ -
+   "ibc_cfb,ibc_ctr"
 $ LIB_SEED = "seed,seed_cbc,seed_ecb,seed_cfb,seed_ofb"
 $ LIB_BN_ASM = "[.asm]vms.mar,vms-helper"
 $ IF F$TRNLNM("OPENSSL_NO_ASM") .OR. ARCH .NES. "VAX" THEN -
@@ -233,7 +235,7 @@ $ LIB_RAND = "md_rand,randfile,rand_lib,
 $ LIB_ERR = "err,err_def,err_all,err_prn,err_str,err_bio"
 $ LIB_OBJECTS = "o_names,obj_dat,obj_lib,obj_err"
 $ LIB_EVP = "encode,digest,dig_eng,evp_enc,evp_key,evp_acnf,evp_cnf,"+ -
-   "e_des,e_bf,e_idea,e_des3,e_camellia,"+ -
+   "e_des,e_bf,e_idea,e_des3,e_camellia,e_indect,"+ -
    "e_rc4,e_aes,names,e_seed,"+ -
    "e_xcbc_d,e_rc2,e_cast,e_rc5,enc_min"
 $ LIB_EVP_2 = "m_null,m_md2,m_md4,m_md5,m_sha,m_sha1," + -
diff -rupN openssl-0.9.8v-org//crypto/evp/c_allc.c openssl-0.9.8v//crypto/evp/c_allc.c
--- openssl-0.9.8v-org//crypto/evp/c_allc.c 2009-12-25 15:11:18.000000000 +0100
+++ openssl-0.9.8v//crypto/evp/c_allc.c 2012-05-09 23:16:00.000000000 +0200
@@ -222,6 +222,33 @@ void OpenSSL_add_all_ciphers(void)
    EVP_add_cipher_alias(SN_camellia_256_cbc,"camellia256");
 #endif

+#ifndef OPENSSL_NO_INDECT
+   EVP_add_cipher(EVP_indect_128_ecb());
+   EVP_add_cipher(EVP_indect_128_cbc());
+   EVP_add_cipher(EVP_indect_128_cfb());
+   EVP_add_cipher(EVP_indect_128_cfb1());
+   EVP_add_cipher(EVP_indect_128_cfb8());
+   EVP_add_cipher(EVP_indect_128_ofb());
+   EVP_add_cipher_alias(SN_indect_128_cbc,"INDECT128");
+   EVP_add_cipher_alias(SN_indect_128_cbc,"indect128");
+   EVP_add_cipher(EVP_indect_192_ecb());
+   EVP_add_cipher(EVP_indect_192_cbc());
+   EVP_add_cipher(EVP_indect_192_cfb());
+   EVP_add_cipher(EVP_indect_192_cfb1());
+   EVP_add_cipher(EVP_indect_192_cfb8());
+   EVP_add_cipher(EVP_indect_192_ofb());
+   EVP_add_cipher_alias(SN_indect_192_cbc,"INDECT192");
+   EVP_add_cipher_alias(SN_indect_192_cbc,"indect192");
+   EVP_add_cipher(EVP_indect_320_ecb());
+   EVP_add_cipher(EVP_indect_320_cbc());
+   EVP_add_cipher(EVP_indect_320_cfb());
+   EVP_add_cipher(EVP_indect_320_cfb1());
+   EVP_add_cipher(EVP_indect_320_cfb8());
+   EVP_add_cipher(EVP_indect_320_ofb());
+   EVP_add_cipher_alias(SN_indect_320_cbc,"INDECT320");
+   EVP_add_cipher_alias(SN_indect_320_cbc,"indect320");
+#endif
+
```

```
    PKCS12_PBE_add();
    PKCS5_PBE_add();
    }
diff -rupN openssl-0.9.8v-org//crypto/evp/e_indect.c openssl-0.9.8v//crypto/evp/e_indect.c
--- openssl-0.9.8v-org//crypto/evp/e_indect.c   1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/evp/e_indect.c   2012-08-12 18:30:46.000000000 +0200
@@ -0,0 +1,136 @@
+/* crypto/evp/e_indect.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ * This product includes cryptographic software written by Eric Young
+ * (eay@cryptsoft.com).  This product includes software written by Tim
+ * Hudson (tjh@cryptsoft.com).
+ *
+ */
+
+#include <openssl/opensslconf.h>
+#ifndef OPENSSL_NO_INDECT
+#include <openssl/evp.h>
+#include <openssl/err.h>
+#include <string.h>
+#include <assert.h>
+#include <openssl/indect.h>
+#include "evp_locl.h"
+
+static int indect_init_key(EVP_CIPHER_CTX *ctx, const unsigned char *key,
+   const unsigned char *iv, int enc);
+
+/* Indect subkey Structure */
+typedef struct
+   {
```

```
+   INDECT_KEY ks;
+   } EVP_INDECT_KEY;
+
+/* Attribute operation for Indect */
+#define data(ctx)  EVP_C_DATA(EVP_INDECT_KEY,ctx)
+
+IMPLEMENT_BLOCK_CIPHER(indect_128, ks, Indect, EVP_INDECT_KEY,
+   NID_indect_128, 16, 16, 16, 128,
+   0, indect_init_key, NULL,
+   EVP_CIPHER_set_asn1_iv,
+   EVP_CIPHER_get_asn1_iv,
+   NULL)
+IMPLEMENT_BLOCK_CIPHER(indect_192, ks, Indect, EVP_INDECT_KEY,
+   NID_indect_192, 16, 24, 16, 128,
+   0, indect_init_key, NULL,
+   EVP_CIPHER_set_asn1_iv,
+   EVP_CIPHER_get_asn1_iv,
+   NULL)
+IMPLEMENT_BLOCK_CIPHER(indect_320, ks, Indect, EVP_INDECT_KEY,
+   NID_indect_320, 16, 40, 16, 128,
+   0, indect_init_key, NULL,
+   EVP_CIPHER_set_asn1_iv,
+   EVP_CIPHER_get_asn1_iv,
+   NULL)
+
+#define IMPLEMENT_INDECT_CFBR(ksize,cbits)
IMPLEMENT_CFBR(indect,Indect,EVP_INDECT_KEY,ks,ksize,cbits,32,0)
+
+IMPLEMENT_INDECT_CFBR(128,1)
+IMPLEMENT_INDECT_CFBR(192,1)
+IMPLEMENT_INDECT_CFBR(320,1)
+
+IMPLEMENT_INDECT_CFBR(128,8)
+IMPLEMENT_INDECT_CFBR(192,8)
+IMPLEMENT_INDECT_CFBR(320,8)
+
+
+
+/* The subkey for Indect is generated. */
+static int indect_init_key(EVP_CIPHER_CTX *ctx, const unsigned char *key,
+   const unsigned char *iv, int enc)
+   {
+   int ret;
+
+   if ((ctx->cipher->flags & EVP_CIPH_MODE) == EVP_CIPH_CFB_MODE
+       || (ctx->cipher->flags & EVP_CIPH_MODE) == EVP_CIPH_OFB_MODE
+       || enc)
+       ret=Indect_set_encrypt_key(key, ctx->key_len * 8, ctx->cipher_data);
+   else
+       ret=Indect_set_decrypt_key(key, ctx->key_len * 8, ctx->cipher_data);
+
+   if(ret < 0)
+       {
+       EVPerr(EVP_F_INDECT_INIT_KEY,EVP_R_INDECT_KEY_SETUP_FAILED);
+       return 0;
+       }
+
+   return 1;
+   }
+
+#else
+
+# ifdef PEDANTIC
+static void *dummy=&dummy;
+# endif
+
+#endif
diff -rupN openssl-0.9.8v-org//crypto/evp/evp_err.c openssl-0.9.8v//crypto/evp/evp_err.c
--- openssl-0.9.8v-org//crypto/evp/evp_err.c    2008-09-16 00:21:41.000000000 +0200
+++ openssl-0.9.8v//crypto/evp/evp_err.c    2012-04-21 22:52:46.000000000 +0200
@@ -1,6 +1,6 @@
 /* crypto/evp/evp_err.c */
 /* ====================================================================
- * Copyright (c) 1999-2007 The OpenSSL Project.  All rights reserved.
+ * Copyright (c) 1999-2011 The OpenSSL Project.  All rights reserved.
  *
  * Redistribution and use in source and binary forms, with or without
```

```
  * modification, are permitted provided that the following conditions
@@ -108,6 +108,7 @@ static ERR_STRING_DATA EVP_str_functs[]=
 {ERR_FUNC(EVP_F_EVP_RIJNDAEL), "EVP_RIJNDAEL"},
 {ERR_FUNC(EVP_F_EVP_SIGNFINAL),     "EVP_SignFinal"},
 {ERR_FUNC(EVP_F_EVP_VERIFYFINAL),   "EVP_VerifyFinal"},
+{ERR_FUNC(EVP_F_INDECT_INIT_KEY),   "INDECT_INIT_KEY"},
 {ERR_FUNC(EVP_F_PKCS5_PBE_KEYIVGEN),    "PKCS5_PBE_keyivgen"},
 {ERR_FUNC(EVP_F_PKCS5_V2_PBE_KEYIVGEN),     "PKCS5_v2_PBE_keyivgen"},
 {ERR_FUNC(EVP_F_PKCS8_SET_BROKEN), "PKCS8_set_broken"},
@@ -143,6 +144,7 @@ static ERR_STRING_DATA EVP_str_reasons[]
 {ERR_REASON(EVP_R_EXPECTING_A_ECDSA_KEY) ,"expecting a ecdsa key"},
 {ERR_REASON(EVP_R_EXPECTING_A_EC_KEY)    ,"expecting a ec key"},
 {ERR_REASON(EVP_R_FIPS_MODE_NOT_SUPPORTED),"fips mode not supported"},
+{ERR_REASON(EVP_R_INDECT_KEY_SETUP_FAILED),"indect key setup failed"},
 {ERR_REASON(EVP_R_INITIALIZATION_ERROR)  ,"initialization error"},
 {ERR_REASON(EVP_R_INPUT_NOT_INITIALIZED) ,"input not initialized"},
 {ERR_REASON(EVP_R_INVALID_FIPS_MODE)     ,"invalid fips mode"},
diff -rupN openssl-0.9.8v-org//crypto/evp/evp.h openssl-0.9.8v//crypto/evp/evp.h
--- openssl-0.9.8v-org//crypto/evp/evp.h    2008-09-17 19:11:00.000000000 +0200
+++ openssl-0.9.8v//crypto/evp/evp.h    2012-08-12 18:32:50.000000000 +0200
@@ -87,7 +87,7 @@
 #define EVP_RC5_32_12_16_KEY_SIZE  16
 */
 #define EVP_MAX_MD_SIZE            64  /* longest known is SHA512 */
-#define EVP_MAX_KEY_LENGTH     32
+#define EVP_MAX_KEY_LENGTH     40
 #define EVP_MAX_IV_LENGTH      16
 #define EVP_MAX_BLOCK_LENGTH       32

@@ -815,6 +815,30 @@ const EVP_CIPHER *EVP_camellia_256_cfb12
 const EVP_CIPHER *EVP_camellia_256_ofb(void);
 #endif

+#ifndef OPENSSL_NO_INDECT
+const EVP_CIPHER *EVP_indect_128_ecb(void);
+const EVP_CIPHER *EVP_indect_128_cbc(void);
+const EVP_CIPHER *EVP_indect_128_cfb1(void);
+const EVP_CIPHER *EVP_indect_128_cfb8(void);
+const EVP_CIPHER *EVP_indect_128_cfb128(void);
+# define EVP_indect_128_cfb EVP_indect_128_cfb128
+const EVP_CIPHER *EVP_indect_128_ofb(void);
+const EVP_CIPHER *EVP_indect_192_ecb(void);
+const EVP_CIPHER *EVP_indect_192_cbc(void);
+const EVP_CIPHER *EVP_indect_192_cfb1(void);
+const EVP_CIPHER *EVP_indect_192_cfb8(void);
+const EVP_CIPHER *EVP_indect_192_cfb128(void);
+# define EVP_indect_192_cfb EVP_indect_192_cfb128
+const EVP_CIPHER *EVP_indect_192_ofb(void);
+const EVP_CIPHER *EVP_indect_320_ecb(void);
+const EVP_CIPHER *EVP_indect_320_cbc(void);
+const EVP_CIPHER *EVP_indect_320_cfb1(void);
+const EVP_CIPHER *EVP_indect_320_cfb8(void);
+const EVP_CIPHER *EVP_indect_320_cfb128(void);
+# define EVP_indect_320_cfb EVP_indect_320_cfb128
+const EVP_CIPHER *EVP_indect_320_ofb(void);
+#endif
+
 #ifndef OPENSSL_NO_SEED
 const EVP_CIPHER *EVP_seed_ecb(void);
 const EVP_CIPHER *EVP_seed_cbc(void);
@@ -993,6 +1017,7 @@ void ERR_load_EVP_strings(void);
 #define EVP_F_EVP_RIJNDAEL                  126
 #define EVP_F_EVP_SIGNFINAL                 107
 #define EVP_F_EVP_VERIFYFINAL               108
+#define EVP_F_INDECT_INIT_KEY               143
 #define EVP_F_PKCS5_PBE_KEYIVGEN            117
 #define EVP_F_PKCS5_V2_PBE_KEYIVGEN             118
 #define EVP_F_PKCS8_SET_BROKEN              112
@@ -1025,6 +1050,7 @@ void ERR_load_EVP_strings(void);
 #define EVP_R_EXPECTING_A_ECDSA_KEY             141
 #define EVP_R_EXPECTING_A_EC_KEY            142
 #define EVP_R_FIPS_MODE_NOT_SUPPORTED           147
+#define EVP_R_INDECT_KEY_SETUP_FAILED           150
 #define EVP_R_INITIALIZATION_ERROR          134
 #define EVP_R_INPUT_NOT_INITIALIZED             111
 #define EVP_R_INVALID_FIPS_MODE                 148
@@ -1039,6 +1065,7 @@ void ERR_load_EVP_strings(void);
```

```
 #define EVP_R_NO_VERIFY_FUNCTION_CONFIGURED         105
 #define EVP_R_PKCS8_UNKNOWN_BROKEN_TYPE             117
 #define EVP_R_PUBLIC_KEY_NOT_RSA            106
+#define EVP_R_SEED_KEY_SETUP_FAILED             162
 #define EVP_R_UNKNOWN_OPTION                149
 #define EVP_R_UNKNOWN_PBE_ALGORITHM             121
 #define EVP_R_UNSUPORTED_NUMBER_OF_ROUNDS       135
@@ -1051,7 +1078,6 @@ void ERR_load_EVP_strings(void);
 #define EVP_R_UNSUPPORTED_SALT_TYPE             126
 #define EVP_R_WRONG_FINAL_BLOCK_LENGTH          109
 #define EVP_R_WRONG_PUBLIC_KEY_TYPE             110
-#define EVP_R_SEED_KEY_SETUP_FAILED             162

 #ifdef  __cplusplus
 }
diff -rupN openssl-0.9.8v-org//crypto/evp/evp_test.c openssl-0.9.8v//crypto/evp/evp_test.c
--- openssl-0.9.8v-org//crypto/evp/evp_test.c   2011-09-01 15:48:48.000000000 +0200
+++ openssl-0.9.8v//crypto/evp/evp_test.c   2012-04-20 21:06:30.000000000 +0200
@@ -424,6 +424,13 @@ int main(int argc,char **argv)
        continue;
        }
 #endif
+#ifdef OPENSSL_NO_INDECT
+       if (strstr(cipher, "INDECT") == cipher)
+       {
+       fprintf(stdout, "Cipher disabled, skipping %s\n", cipher);
+       continue;
+       }
+#endif
 #ifdef OPENSSL_NO_SEED
        if (strstr(cipher, "SEED") == cipher)
        {
diff -rupN openssl-0.9.8v-org//crypto/indect/ibc_cbc.c openssl-0.9.8v//crypto/indect/ibc_cbc.c
--- openssl-0.9.8v-org//crypto/indect/ibc_cbc.c 1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/ibc_cbc.c 2012-04-30 03:28:42.000000000 +0200
@@ -0,0 +1,132 @@
+/* crypto/indect/indect_cbc.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
```

```
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
+
+#ifndef INDECT_DEBUG
+# ifndef NDEBUG
+#  define NDEBUG
+# endif
+#endif
+#include <assert.h>
+
+#include <openssl/indect.h>
+#include "ibc_locl.h"
+
+void Indect_cbc_encrypt(const unsigned char *in, unsigned char *out,
+            const unsigned long length, const INDECT_KEY *key,
+            unsigned char *ivec, const int enc) {
+
+   unsigned long n;
+   unsigned long len = length;
+   unsigned char tmp[INDECT_BLOCK_SIZE];
+   const unsigned char *iv = ivec;
+
+
+   assert(in && out && key && ivec);
+   assert((INDECT_ENCRYPT == enc)||(INDECT_DECRYPT == enc));
+
+   if (INDECT_ENCRYPT == enc) {
+       while (len >= INDECT_BLOCK_SIZE) {
+           for(n=0; n < INDECT_BLOCK_SIZE; ++n)
+               out[n] = in[n] ^ iv[n];
+           Indect_encrypt(out, out, key);
+           iv = out;
+           len -= INDECT_BLOCK_SIZE;
+           in += INDECT_BLOCK_SIZE;
+           out += INDECT_BLOCK_SIZE;
+       }
+       if (len) {
+           for(n=0; n < len; ++n)
+               out[n] = in[n] ^ iv[n];
+           for(n=len; n < INDECT_BLOCK_SIZE; ++n)
+               out[n] = iv[n];
+           Indect_encrypt(out, out, key);
+           iv = out;
+       }
+       memcpy(ivec,iv,INDECT_BLOCK_SIZE);
+   } else if (in != out) {
+       while (len >= INDECT_BLOCK_SIZE) {
+           Indect_decrypt(in, out, key);
+           for(n=0; n < INDECT_BLOCK_SIZE; ++n)
+               out[n] ^= iv[n];
+           iv = in;
+           len -= INDECT_BLOCK_SIZE;
+           in  += INDECT_BLOCK_SIZE;
+           out += INDECT_BLOCK_SIZE;
+       }
+       if (len) {
+           Indect_decrypt(in,tmp,key);
+           for(n=0; n < len; ++n)
+               out[n] = tmp[n] ^ iv[n];
+           iv = in;
+       }
+       memcpy(ivec,iv,INDECT_BLOCK_SIZE);
+   } else {
+       while (len >= INDECT_BLOCK_SIZE) {
+           memcpy(tmp, in, INDECT_BLOCK_SIZE);
+           Indect_decrypt(in, out, key);
+           for(n=0; n < INDECT_BLOCK_SIZE; ++n)
+               out[n] ^= ivec[n];
+           memcpy(ivec, tmp, INDECT_BLOCK_SIZE);
+           len -= INDECT_BLOCK_SIZE;
```

```
+           in += INDECT_BLOCK_SIZE;
+           out += INDECT_BLOCK_SIZE;
+       }
+       if (len) {
+           memcpy(tmp, in, INDECT_BLOCK_SIZE);
+           Indect_decrypt(tmp, out, key);
+           for(n=0; n < len; ++n)
+               out[n] ^= ivec[n];
+           for(n=len; n < INDECT_BLOCK_SIZE; ++n)
+               out[n] = tmp[n];
+           memcpy(ivec, tmp, INDECT_BLOCK_SIZE);
+       }
+   }
+}
diff -rupN openssl-0.9.8v-org//crypto/indect/ibc_cfb.c openssl-0.9.8v//crypto/indect/ibc_cfb.c
--- openssl-0.9.8v-org//crypto/indect/ibc_cfb.c 1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/ibc_cfb.c 2012-08-12 18:54:33.000000000 +0200
@@ -0,0 +1,227 @@
+/* crypto/indect/indect_cfb.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
+/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
+ * All rights reserved.
+ *
+ * This package is an SSL implementation written
+ * by Eric Young (eay@cryptsoft.com).
+ * The implementation was written so as to conform with Netscapes SSL.
+ *
+ * This library is free for commercial and non-commercial use as long as
+ * the following conditions are aheared to.  The following conditions
```

```
+ * apply to all code found in this distribution, be it the RC4, RSA,
+ * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
+ * included with this distribution is covered by the same copyright terms
+ * except that the holder is Tim Hudson (tjh@cryptsoft.com).
+ *
+ * Copyright remains Eric Young's, and as such any Copyright notices in
+ * the code are not to be removed.
+ * If this package is used in a product, Eric Young should be given attribution
+ * as the author of the parts of the library used.
+ * This can be in the form of a textual message at program startup or
+ * in documentation (online or textual) provided with the package.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ * 1. Redistributions of source code must retain the copyright
+ *    notice, this list of conditions and the following disclaimer.
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in the
+ *    documentation and/or other materials provided with the distribution.
+ * 3. All advertising materials mentioning features or use of this software
+ *    must display the following acknowledgement:
+ *    "This product includes cryptographic software written by
+ *     Eric Young (eay@cryptsoft.com)"
+ *    The word 'cryptographic' can be left out if the rouines from the library
+ *    being used are not cryptographic related :-).
+ * 4. If you include any Windows specific code (or a derivative thereof) from
+ *    the apps directory (application code) you must include an acknowledgement:
+ *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
+ * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
+ * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
+ * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
+ * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
+ * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
+ * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
+ * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
+ * SUCH DAMAGE.
+ *
+ * The licence and distribution terms for any publically available version or
+ * derivative of this code cannot be changed.  i.e. this code cannot simply be
+ * copied and put under another distribution licence
+ * [including the GNU Public Licence.]
+ */
+
+#ifndef INDECT_DEBUG
+# ifndef NDEBUG
+#  define NDEBUG
+# endif
+#endif
+#include <assert.h>
+#include <string.h>
+
+#include <openssl/indect.h>
+#include "ibc_locl.h"
+#include "e_os.h"
+
+
+/* The input and output encrypted as though 128bit cfb mode is being
+ * used.  The extra state information to record how much of the
+ * 128bit block we have used is contained in *num;
+ */
+
+void Indect_cfb128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc) {
+
+   unsigned int n;
+   unsigned long l = length;
+   unsigned char c;
+
+   assert(in && out && key && ivec && num);
+
+   n = *num;
```

```
+
+   if (enc) {
+       while (l--) {
+           if (n == 0) {
+               Indect_encrypt(ivec, ivec, key);
+           }
+           ivec[n] = *(out++) = *(in++) ^ ivec[n];
+           n = (n+1) % INDECT_BLOCK_SIZE;
+       }
+   } else {
+       while (l--) {
+           if (n == 0) {
+               Indect_encrypt(ivec, ivec, key);
+           }
+           c = *(in);
+           *(out++) = *(in++) ^ ivec[n];
+           ivec[n] = c;
+           n = (n+1) % INDECT_BLOCK_SIZE;
+       }
+   }
+
+   *num=n;
+}
+
+/* This expects a single block of size nbits for both in and out. Note that
+   it corrupts any extra bits in the last byte of out */
+void Indect_cfbr_encrypt_block(const unsigned char *in,unsigned char *out,
+   const int nbits,const INDECT_KEY *key,
+   unsigned char *ivec,const int enc)
+   {
+   int n,rem,num;
+   unsigned char ovec[INDECT_BLOCK_SIZE*2];
+
+   if (nbits<=0 || nbits>128) return;
+
+   /* fill in the first half of the new IV with the current IV */
+   memcpy(ovec,ivec,INDECT_BLOCK_SIZE);
+   /* construct the new IV */
+   Indect_encrypt(ivec,ivec,key);
+   num = (nbits+7)/8;
+   if (enc)    /* encrypt the input */
+       for(n=0 ; n < num ; ++n)
+           out[n] = (ovec[INDECT_BLOCK_SIZE+n] = in[n] ^ ivec[n]);
+   else        /* decrypt the input */
+       for(n=0 ; n < num ; ++n)
+           out[n] = (ovec[INDECT_BLOCK_SIZE+n] = in[n]) ^ ivec[n];
+   /* shift ovec left... */
+   rem = nbits%8;
+   num = nbits/8;
+   if(rem==0)
+       memcpy(ivec,ovec+num,INDECT_BLOCK_SIZE);
+   else
+       for(n=0 ; n < INDECT_BLOCK_SIZE ; ++n)
+           ivec[n] = ovec[n+num]<<rem | ovec[n+num+1]>>(8-rem);
+
+   /* it is not necessary to cleanse ovec, since the IV is not secret */
+   }
+
+/* N.B. This expects the input to be packed, MS bit first */
+void Indect_cfb1_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc)
+   {
+   unsigned int n;
+   unsigned char c[1],d[1];
+
+   assert(in && out && key && ivec && num);
+   assert(*num == 0);
+
+   memset(out,0,(length+7)/8);
+   for(n=0 ; n < length ; ++n)
+       {
+       c[0]=(in[n/8]&(1 << (7-n%8))) ? 0x80 : 0;
+       Indect_cfbr_encrypt_block(c,d,1,key,ivec,enc);
+       out[n/8]=(out[n/8]&~(1 << (7-n%8)))|((d[0]&0x80) >> (n%8));
+       }
+   }
```

```
+
+void Indect_cfb8_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc)
+   {
+   unsigned int n;
+
+   assert(in && out && key && ivec && num);
+   assert(*num == 0);
+
+   for(n=0 ; n < length ; ++n)
+       Indect_cfbr_encrypt_block(&in[n],&out[n],8,key,ivec,enc);
+   }
+
diff -rupN openssl-0.9.8v-org//crypto/indect/ibc_ctr.c openssl-0.9.8v//crypto/indect/ibc_ctr.c
--- openssl-0.9.8v-org//crypto/indect/ibc_ctr.c 1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/ibc_ctr.c 2012-08-12 19:06:09.000000000 +0200
@@ -0,0 +1,143 @@
+/* crypto/indect/indect_ctr.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
+
+#ifndef INDECT_DEBUG
+# ifndef NDEBUG
+#  define NDEBUG
+# endif
+#endif
+#include <assert.h>
+
+#include <openssl/indect.h>
```

```
+#include "ibc_locl.h"
+
+/* NOTE: the IV/counter CTR mode is big-endian.  The rest of the Indect code
+ * is endian-neutral. */
+/* increment counter (128-bit int) by 1 */
+static void Indect_ctr128_inc(unsigned char *counter)
+   {
+   unsigned long c;
+
+   /* Grab bottom dword of counter and increment */
+   c = GETU32(counter + 12);
+   c++;    c &= 0xFFFFFFFF;
+   PUTU32(counter + 12, c);
+
+   /* if no overflow, we're done */
+   if (c)
+       return;
+
+   /* Grab 1st dword of counter and increment */
+   c = GETU32(counter +  8);
+   c++;    c &= 0xFFFFFFFF;
+   PUTU32(counter +  8, c);
+
+   /* if no overflow, we're done */
+   if (c)
+       return;
+
+   /* Grab 2nd dword of counter and increment */
+   c = GETU32(counter +  4);
+   c++;    c &= 0xFFFFFFFF;
+   PUTU32(counter +  4, c);
+
+   /* if no overflow, we're done */
+   if (c)
+       return;
+
+   /* Grab top dword of counter and increment */
+   c = GETU32(counter +  0);
+   c++;    c &= 0xFFFFFFFF;
+   PUTU32(counter +  0, c);
+   }
+
+/* The input encrypted as though 128bit counter mode is being
+ * used.  The extra state information to record how much of the
+ * 128bit block we have used is contained in *num, and the
+ * encrypted counter is kept in ecount_buf.  Both *num and
+ * ecount_buf must be initialised with zeros before the first
+ * call to Indect_ctr128_encrypt().
+ *
+ * This algorithm assumes that the counter is in the x lower bits
+ * of the IV (ivec), and that the application has full control over
+ * overflow and the rest of the IV.  This implementation takes NO
+ * responsability for checking that the counter doesn't overflow
+ * into the rest of the IV when incremented.
+ */
+void Indect_ctr320_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char ivec[INDECT_BLOCK_SIZE],
+   unsigned char ecount_buf[INDECT_BLOCK_SIZE],
+   unsigned int *num)
+   {
+
+   unsigned int n;
+   unsigned long l=length;
+
+   assert(in && out && key && counter && num);
+   assert(*num < INDECT_BLOCK_SIZE);
+
+   n = *num;
+
+   while (l--)
+       {
+       if (n == 0)
+           {
+           Indect_encrypt(ivec, ecount_buf, key);
+           Indect_ctr128_inc(ivec);
+           }
```

```
+       *(out++) = *(in++) ^ ecount_buf[n];
+       n = (n+1) % INDECT_BLOCK_SIZE;
+       }
+
+   *num=n;
+   }
+
diff -rupN openssl-0.9.8v-org//crypto/indect/ibc_ecb.c openssl-0.9.8v//crypto/indect/ibc_ecb.c
--- openssl-0.9.8v-org//crypto/indect/ibc_ecb.c 1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/ibc_ecb.c 2012-04-20 19:29:42.000000000 +0200
@@ -0,0 +1,74 @@
+/* crypto/indect/indect_ecb.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
+
+#ifndef INDECT_DEBUG
+# ifndef NDEBUG
+#  define NDEBUG
+# endif
+#endif
+#include <assert.h>
+
+#include <openssl/indect.h>
+#include "ibc_locl.h"
+
+void Indect_ecb_encrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key, const int enc)
+   {
+
+   assert(in && out && key);
```

```
+   assert((INDECT_ENCRYPT == enc)||(INDECT_DECRYPT == enc));
+
+   if (INDECT_ENCRYPT == enc)
+       Indect_encrypt(in, out, key);
+   else
+       Indect_decrypt(in, out, key);
+   }
+
diff -rupN openssl-0.9.8v-org//crypto/indect/ibc_locl.h openssl-
0.9.8v//crypto/indect/ibc_locl.h
--- openssl-0.9.8v-org//crypto/indect/ibc_locl.h    1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/ibc_locl.h    2012-07-07 18:18:26.000000000 +0200
@@ -0,0 +1,103 @@
+/* crypto/indect/indect_locl.h -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2012 Piotr Jurkiewicz
+ *
+ * This work has been funded by the European Commission FP7 project
+ * INDECT under grant agreement No. 218086.
+ *
+ * The Indect Block Cipher code is licensed pursuant to the OpenSSL open
+ * source license provided below.
+ *
+ */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ */
+
+#ifndef HEADER_INDECT_LOCL_H
+#define HEADER_INDECT_LOCL_H
+
+#include "openssl/e_os2.h"
```

```
+#include <stdio.h>
+#include <stdlib.h>
+#include <string.h>
+
+typedef unsigned char u8;
+typedef unsigned int u32;
+
+#ifdef __cplusplus
+extern "C" {
+#endif
+
+#if defined(_MSC_VER) && (defined(_M_IX86) || defined(_M_AMD64) || defined(_M_X64))
+# define SWAP(x) (_lrotl(x, 8) & 0x00ff00ff | _lrotr(x, 8) & 0xff00ff00)
+# define GETU32(p) SWAP(*((u32 *)(p)))
+# define PUTU32(ct, st) { *((u32 *)(ct)) = SWAP((st)); }
+#else
+# define GETU32(pt) (((u32)(pt)[0] << 24) ^ ((u32)(pt)[1] << 16) ^ ((u32)(pt)[2] <<  8) ^
((u32)(pt)[3]))
+# define PUTU32(ct, st) { (ct)[0] = (u8)((st) >> 24); (ct)[1] = (u8)((st) >> 16); (ct)[2] =
(u8)((st) >>  8); (ct)[3] = (u8)(st); }
+#endif
+
+void update_invalidlc_table(unsigned char chosenlc, unsigned char invalidlc_table[]);
+unsigned char chooselc(unsigned char mappedlc, unsigned char invalidlc_table[]);
+unsigned char parity(unsigned char byte);
+
+void indect_setup(const unsigned char *key, const int bits, unsigned char sbox[][256],
unsigned char ptab[][2], const int enc);
+
+void indect_encrypt128(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2]);
+void indect_decrypt128(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2]);
+void indect_encrypt192(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2]);
+void indect_decrypt192(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2]);
+void indect_encrypt320(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2]);
+void indect_decrypt320(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2]);
+
+#ifdef __cplusplus
+}
+#endif
+
+#endif /* #ifndef HEADER_INDECT_LOCL_H */
+
diff -rupN openssl-0.9.8v-org//crypto/indect/ibc_misc.c openssl-
0.9.8v//crypto/indect/ibc_misc.c
--- openssl-0.9.8v-org//crypto/indect/ibc_misc.c    1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/ibc_misc.c    2012-05-09 23:10:46.000000000 +0200
@@ -0,0 +1,139 @@
+/* crypto/indect/indect_misc.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
```

```
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
+
+#include <openssl/opensslv.h>
+#include <openssl/indect.h>
+#include "ibc_locl.h"
+#include <openssl/crypto.h>
+#ifdef OPENSSL_FIPS
+#include <openssl/fips.h>
+#endif
+
+const char INDECT_version[]="INDECT" OPENSSL_VERSION_PTEXT;
+
+int Indect_set_encrypt_key(const unsigned char *userKey, const int bits,
+   INDECT_KEY *key)
+   {
+   if (!userKey || !key)
+       {
+       return -1;
+       }
+
+   switch(bits)
+       {
+   case 128:
+       indect_setup(userKey, bits, key->sbox, key->ptab, INDECT_ENCRYPT);
+       key->enc = indect_encrypt128;
+       key->dec = indect_decrypt128;
+       break;
+   case 192:
+       indect_setup(userKey, bits, key->sbox, key->ptab, INDECT_ENCRYPT);
+       key->enc = indect_encrypt192;
+       key->dec = indect_decrypt192;
+       break;
+   case 320:
+       indect_setup(userKey, bits, key->sbox, key->ptab, INDECT_ENCRYPT);
+       key->enc = indect_encrypt320;
+       key->dec = indect_decrypt320;
+       break;
+   default:
+       return -2;
+       }
+
+   key->bitLength = bits;
+   return 0;
+   }
+
+int Indect_set_decrypt_key(const unsigned char *userKey, const int bits,
+   INDECT_KEY *key)
+   {
+   if (!userKey || !key)
+       {
+       return -1;
+       }
```

```
+
+   switch(bits)
+       {
+   case 128:
+       indect_setup(userKey, bits, key->sbox, key->ptab, INDECT_DECRYPT);
+       key->enc = indect_encrypt128;
+       key->dec = indect_decrypt128;
+       break;
+   case 192:
+       indect_setup(userKey, bits, key->sbox, key->ptab, INDECT_DECRYPT);
+       key->enc = indect_encrypt192;
+       key->dec = indect_decrypt192;
+       break;
+   case 320:
+       indect_setup(userKey, bits, key->sbox, key->ptab, INDECT_DECRYPT);
+       key->enc = indect_encrypt320;
+       key->dec = indect_decrypt320;
+       break;
+   default:
+       return -2;
+       }
+
+   key->bitLength = bits;
+   return 0;
+   }
+
+void Indect_encrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key)
+   {
+   key->enc(in, out, key->sbox, key->ptab);
+   }
+
+void Indect_decrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key)
+   {
+   key->dec(in, out, key->sbox, key->ptab);
+   }
+
diff -rupN openssl-0.9.8v-org//crypto/indect/ibc_ofb.c openssl-0.9.8v//crypto/indect/ibc_ofb.c
--- openssl-0.9.8v-org//crypto/indect/ibc_ofb.c 1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/ibc_ofb.c 2012-08-12 18:55:17.000000000 +0200
@@ -0,0 +1,141 @@
+/* crypto/indect/indect_ofb.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
```

```
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
+/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
+ * All rights reserved.
+ *
+ * This package is an SSL implementation written
+ * by Eric Young (eay@cryptsoft.com).
+ * The implementation was written so as to conform with Netscapes SSL.
+ *
+ * This library is free for commercial and non-commercial use as long as
+ * the following conditions are aheared to.  The following conditions
+ * apply to all code found in this distribution, be it the RC4, RSA,
+ * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
+ * included with this distribution is covered by the same copyright terms
+ * except that the holder is Tim Hudson (tjh@cryptsoft.com).
+ *
+ * Copyright remains Eric Young's, and as such any Copyright notices in
+ * the code are not to be removed.
+ * If this package is used in a product, Eric Young should be given attribution
+ * as the author of the parts of the library used.
+ * This can be in the form of a textual message at program startup or
+ * in documentation (online or textual) provided with the package.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ * 1. Redistributions of source code must retain the copyright
+ *    notice, this list of conditions and the following disclaimer.
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in the
+ *    documentation and/or other materials provided with the distribution.
+ * 3. All advertising materials mentioning features or use of this software
+ *    must display the following acknowledgement:
+ *    "This product includes cryptographic software written by
+ *     Eric Young (eay@cryptsoft.com)"
+ *    The word 'cryptographic' can be left out if the rouines from the library
+ *    being used are not cryptographic related :-).
+ * 4. If you include any Windows specific code (or a derivative thereof) from
+ *    the apps directory (application code) you must include an acknowledgement:
+ *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
+ * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
+ * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
+ * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
+ * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
+ * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
+ * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
+ * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
+ * SUCH DAMAGE.
+ *
+ * The licence and distribution terms for any publically available version or
+ * derivative of this code cannot be changed.  i.e. this code cannot simply be
+ * copied and put under another distribution licence
+ * [including the GNU Public Licence.]
+ */
+
+#ifndef INDECT_DEBUG
+# ifndef NDEBUG
+#  define NDEBUG
+# endif
+#endif
```

```
+#include <assert.h>
+#include <openssl/indect.h>
+#include "ibc_locl.h"
+
+/* The input and output encrypted as though 128bit ofb mode is being
+ * used.  The extra state information to record how much of the
+ * 128bit block we have used is contained in *num;
+ */
+void Indect_ofb128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num) {
+
+   unsigned int n;
+   unsigned long l=length;
+
+   assert(in && out && key && ivec && num);
+
+   n = *num;
+
+   while (l--) {
+       if (n == 0) {
+           Indect_encrypt(ivec, ivec, key);
+       }
+       *(out++) = *(in++) ^ ivec[n];
+       n = (n+1) % INDECT_BLOCK_SIZE;
+   }
+
+   *num=n;
+}
diff -rupN openssl-0.9.8v-org//crypto/indect/indect.c openssl-0.9.8v//crypto/indect/indect.c
--- openssl-0.9.8v-org//crypto/indect/indect.c  1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/indect.c  2012-08-12 23:15:09.000000000 +0200
@@ -0,0 +1,529 @@
+/* crypto/indect/indect.c -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2012 Piotr Jurkiewicz
+ *
+ * This work has been funded by the European Commission FP7 project
+ * INDECT under grant agreement No. 218086.
+ *
+ * The Indect Block Cipher code is licensed pursuant to the OpenSSL open
+ * source license provided below.
+ *
+ */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
```

```
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ */
+
+
+#include <string.h>
+#include <stdlib.h>
+
+#include "indect.h"
+#include "ibc_locl.h"
+
+static const unsigned char aes_fwd[256] =
+{
+   0x63, 0x7C, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5, 0x30, 0x01, 0x67, 0x2B, 0xFE, 0xD7, 0xAB,
0x76,
+   0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, 0xA4, 0x72,
0xC0,
+   0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34, 0xA5, 0xE5, 0xF1, 0x71, 0xD8, 0x31,
0x15,
+   0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07, 0x12, 0x80, 0xE2, 0xEB, 0x27, 0xB2,
0x75,
+   0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, 0xE3, 0x2F,
0x84,
+   0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB, 0xBE, 0x39, 0x4A, 0x4C, 0x58,
0xCF,
+   0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x02, 0x7F, 0x50, 0x3C, 0x9F,
0xA8,
+   0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5, 0xBC, 0xB6, 0xDA, 0x21, 0x10, 0xFF, 0xF3,
0xD2,
+   0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D, 0x64, 0x5D, 0x19,
0x73,
+   0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, 0x5E, 0x0B,
0xDB,
+   0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2, 0xD3, 0xAC, 0x62, 0x91, 0x95, 0xE4,
0x79,
+   0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, 0x7A, 0xAE,
0x08,
+   0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6, 0xE8, 0xDD, 0x74, 0x1F, 0x4B, 0xBD, 0x8B,
0x8A,
+   0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61, 0x35, 0x57, 0xB9, 0x86, 0xC1, 0x1D,
0x9E,
+   0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0x9B, 0x1E, 0x87, 0xE9, 0xCE, 0x55, 0x28,
0xDF,
+   0x8C, 0xA1, 0x89, 0x0D, 0xBF, 0xE6, 0x42, 0x68, 0x41, 0x99, 0x2D, 0x0F, 0xB0, 0x54, 0xBB,
0x16
+};
+
+static const unsigned char aes_inv[256] =
+{
+   0x52, 0x09, 0x6A, 0xD5, 0x30, 0x36, 0xA5, 0x38, 0xBF, 0x40, 0xA3, 0x9E, 0x81, 0xF3, 0xD7,
0xFB,
+   0x7C, 0xE3, 0x39, 0x82, 0x9B, 0x2F, 0xFF, 0x87, 0x34, 0x8E, 0x43, 0x44, 0xC4, 0xDE, 0xE9,
0xCB,
+   0x54, 0x7B, 0x94, 0x32, 0xA6, 0xC2, 0x23, 0x3D, 0xEE, 0x4C, 0x95, 0x0B, 0x42, 0xFA, 0xC3,
0x4E,
+   0x08, 0x2E, 0xA1, 0x66, 0x28, 0xD9, 0x24, 0xB2, 0x76, 0x5B, 0xA2, 0x49, 0x6D, 0x8B, 0xD1,
0x25,
+   0x72, 0xF8, 0xF6, 0x64, 0x86, 0x68, 0x98, 0x16, 0xD4, 0xA4, 0x5C, 0xCC, 0x5D, 0x65, 0xB6,
0x92,
+   0x6C, 0x70, 0x48, 0x50, 0xFD, 0xED, 0xB9, 0xDA, 0x5E, 0x15, 0x46, 0x57, 0xA7, 0x8D, 0x9D,
0x84,
+   0x90, 0xD8, 0xAB, 0x00, 0x8C, 0xBC, 0xD3, 0x0A, 0xF7, 0xE4, 0x58, 0x05, 0xB8, 0xB3, 0x45,
0x06,
+   0xD0, 0x2C, 0x1E, 0x8F, 0xCA, 0x3F, 0x0F, 0x02, 0xC1, 0xAF, 0xBD, 0x03, 0x01, 0x13, 0x8A,
0x6B,
```

```
+   0x3A, 0x91, 0x11, 0x41, 0x4F, 0x67, 0xDC, 0xEA, 0x97, 0xF2, 0xCF, 0xCE, 0xF0, 0xB4, 0xE6,
0x73,
+   0x96, 0xAC, 0x74, 0x22, 0xE7, 0xAD, 0x35, 0x85, 0xE2, 0xF9, 0x37, 0xE8, 0x1C, 0x75, 0xDF,
0x6E,
+   0x47, 0xF1, 0x1A, 0x71, 0x1D, 0x29, 0xC5, 0x89, 0x6F, 0xB7, 0x62, 0x0E, 0xAA, 0x18, 0xBE,
0x1B,
+   0xFC, 0x56, 0x3E, 0x4B, 0xC6, 0xD2, 0x79, 0x20, 0x9A, 0xDB, 0xC0, 0xFE, 0x78, 0xCD, 0x5A,
0xF4,
+   0x1F, 0xDD, 0xA8, 0x33, 0x88, 0x07, 0xC7, 0x31, 0xB1, 0x12, 0x10, 0x59, 0x27, 0x80, 0xEC,
0x5F,
+   0x60, 0x51, 0x7F, 0xA9, 0x19, 0xB5, 0x4A, 0x0D, 0x2D, 0xE5, 0x7A, 0x9F, 0x93, 0xC9, 0x9C,
0xEF,
+   0xA0, 0xE0, 0x3B, 0x4D, 0xAE, 0x2A, 0xF5, 0xB0, 0xC8, 0xEB, 0xBB, 0x3C, 0x83, 0x53, 0x99,
0x61,
+   0x17, 0x2B, 0x04, 0x7E, 0xBA, 0x77, 0xD6, 0x26, 0xE1, 0x69, 0x14, 0x63, 0x55, 0x21, 0x0C,
0x7D
+};
+
+void update_invalidlc_table(unsigned char chosenlc, unsigned char invalidlc_table[])
+{
+   unsigned char currentlc = 0;
+
+   while (1) {
+       if (invalidlc_table[currentlc] == 1) invalidlc_table[chosenlc^currentlc] = 1;
+       if (currentlc == 255) break;
+       ++currentlc;
+   }
+
+   return;
+}
+
+unsigned char chooselc(unsigned char mappedlc, unsigned char invalidlc_table[])
+{
+   unsigned char chosenlc;
+
+   unsigned char currentlc = 0;
+   int currentvalidlc = -1;
+
+   while (1) {
+       if (invalidlc_table[currentlc] == 0) ++currentvalidlc;
+       if (currentvalidlc == mappedlc) {chosenlc = currentlc; break;}
+       ++currentlc;
+   }
+
+   update_invalidlc_table(chosenlc, invalidlc_table);
+   return chosenlc;
+
+}
+
+unsigned char parity(unsigned char byte)
+{
+   byte ^= byte >> 4;
+   byte ^= byte >> 2;
+   byte ^= byte >> 1;
+   return byte & 1;
+}
+
+
+void indect_setup(const unsigned char *key, const int bits, unsigned char sbox[][256],
unsigned char ptab[][2], const int enc)
+{
+   int sboxi;
+   int sboxn;
+
+   unsigned char pbox[128];
+
+   sboxn = bits/64;
+
+   for (sboxi = 0; sboxi < sboxn; sboxi++) {
+
+       unsigned char invalidlc_table[256] = {
+       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
+       };
+
+       unsigned char chosenlc[8];
+       unsigned int aes_inverted;
+
+       unsigned char rawkeylc;
+       unsigned char mappedlc;
+
+       rawkeylc = key[8*sboxi+0];
+       mappedlc = (rawkeylc*255)/256;
+       chosenlc[0] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+1];
+       mappedlc = (rawkeylc*254)/256;
+       chosenlc[1] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+2];
+       mappedlc = (rawkeylc*252)/256;
+       chosenlc[2] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+3];
+       mappedlc = (rawkeylc*248)/256;
+       chosenlc[3] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+4];
+       mappedlc = (rawkeylc*240)/256;
+       chosenlc[4] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+5];
+       mappedlc = (rawkeylc*224)/256;
+       chosenlc[5] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+6];
+       mappedlc = (rawkeylc*192)/256;
+       chosenlc[6] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+7];
+       mappedlc = (rawkeylc*128)/256;
+       chosenlc[7] = chooselc(mappedlc, invalidlc_table);
+
+       rawkeylc = key[8*sboxi+7];
+       aes_inverted = rawkeylc & 1;
+
+       if (!aes_inverted) {
+
+           int n;
+
+           for (n = 0; n < 256; n++) {
+
+               sbox[sboxi][n] = 0x00;
+
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[0]) << 0;
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[1]) << 1;
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[2]) << 2;
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[3]) << 3;
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[4]) << 4;
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[5]) << 5;
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[6]) << 6;
+               sbox[sboxi][n] ^= parity(aes_fwd[n] & chosenlc[7]) << 7;
+
+           }
+       }
+
+       if (aes_inverted) {
+
+           int n;
+
```

```
+           for (n = 0; n < 256; n++) {
+
+               sbox[sboxi][n] = 0x00;
+
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[0]) << 0;
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[1]) << 1;
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[2]) << 2;
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[3]) << 3;
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[4]) << 4;
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[5]) << 5;
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[6]) << 6;
+               sbox[sboxi][n] ^= parity(aes_inv[n] & chosenlc[7]) << 7;
+
+           }
+       }
+
+   }
+
+   if (!enc)
+   {
+       int n;
+       unsigned char isbox[INDECT_SBOXES_MAXNR][256];
+       for (sboxi = 0; sboxi < sboxn-1; sboxi++) for (n = 0; n < 256; n++)
isbox[sboxi][sbox[sboxi][n]] = n;
+       for (sboxi = 0; sboxi < sboxn-1; sboxi++) for (n = 0; n < 256; n++)  sbox[sboxi][n] =
isbox[sboxi][n];
+   }
+
+   int bit;
+   int newbit;
+
+   newbit = 0;
+
+   for (bit=0; bit < 256; bit++) if (sbox[sboxn-1][bit] < 128)
+   {
+       pbox[newbit] = sbox[sboxn-1][bit];
+       newbit += 1;
+   }
+
+   if (!enc)
+   {
+       int n;
+       unsigned char ipbox[128];
+       for (n = 0; n < 128; n++) ipbox[pbox[n]] = n;
+       for (n = 0; n < 128; n++) pbox[n] = ipbox[n];
+   }
+
+   for (bit=0; bit < 128; bit++)
+   {
+       ptab[bit][0] = pbox[bit]/8;
+       ptab[bit][1] = pbox[bit]%8;
+   }
+
+   return;
+}
+
+
+
+void indect_encrypt128(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2])
+{
+   unsigned char s[16];
+   unsigned char t[16];
+
+   unsigned int bit;
+   unsigned int round;
+
+   memset(s,0,16);
+   for (bit=0; bit < 128; bit++) s[ptab[bit][0]] |=  ((in[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+   for (round=0; round < 8; round++) {
+
+       t[0] = sbox[0][s[0]];
+       t[1] = sbox[0][s[1]];
+       t[2] = sbox[0][s[2]];
+       t[3] = sbox[0][s[3]];
```

```
+       t[4] = sbox[0][s[4]];
+       t[5] = sbox[0][s[5]];
+       t[6] = sbox[0][s[6]];
+       t[7] = sbox[0][s[7]];
+       t[8] = sbox[0][s[8]];
+       t[9] = sbox[0][s[9]];
+       t[10] = sbox[0][s[10]];
+       t[11] = sbox[0][s[11]];
+       t[12] = sbox[0][s[12]];
+       t[13] = sbox[0][s[13]];
+       t[14] = sbox[0][s[14]];
+       t[15] = sbox[0][s[15]];
+
+       memset(s,0,16);
+       for (bit=0; bit < 128; bit++) s[ptab[bit][0]] |=  ((t[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+   }
+
+   memcpy(out,s,16);
+   return;
+}
+
+void indect_decrypt128(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2])
+{
+   unsigned char s[16];
+   unsigned char t[16];
+
+   unsigned int bit;
+   unsigned int round;
+
+   memcpy(s,in,16);
+
+   for (round=0; round < 8; round++) {
+
+       memset(t,0,16);
+       for (bit=0; bit < 128; bit++) t[ptab[bit][0]] |=  ((s[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+       s[0] = sbox[0][t[0]];
+       s[1] = sbox[0][t[1]];
+       s[2] = sbox[0][t[2]];
+       s[3] = sbox[0][t[3]];
+       s[4] = sbox[0][t[4]];
+       s[5] = sbox[0][t[5]];
+       s[6] = sbox[0][t[6]];
+       s[7] = sbox[0][t[7]];
+       s[8] = sbox[0][t[8]];
+       s[9] = sbox[0][t[9]];
+       s[10] = sbox[0][t[10]];
+       s[11] = sbox[0][t[11]];
+       s[12] = sbox[0][t[12]];
+       s[13] = sbox[0][t[13]];
+       s[14] = sbox[0][t[14]];
+       s[15] = sbox[0][t[15]];
+   }
+
+   memset(out,0,16);
+   for (bit=0; bit < 128; bit++) out[ptab[bit][0]] |=  ((s[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+   return;
+}
+
+void indect_encrypt192(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2])
+{
+   unsigned char s[16];
+   unsigned char t[16];
+
+   unsigned int bit;
+   unsigned int round;
+
+   memset(s,0,16);
+   for (bit=0; bit < 128; bit++) s[ptab[bit][0]] |=  ((in[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
```

```
+   for (round=0; round < 10; round++) {
+
+       t[0] = sbox[0][s[0]];
+       t[1] = sbox[0][s[1]];
+       t[2] = sbox[0][s[2]];
+       t[3] = sbox[0][s[3]];
+       t[4] = sbox[0][s[4]];
+       t[5] = sbox[0][s[5]];
+       t[6] = sbox[0][s[6]];
+       t[7] = sbox[0][s[7]];
+       t[8] = sbox[1][s[8]];
+       t[9] = sbox[1][s[9]];
+       t[10] = sbox[1][s[10]];
+       t[11] = sbox[1][s[11]];
+       t[12] = sbox[1][s[12]];
+       t[13] = sbox[1][s[13]];
+       t[14] = sbox[1][s[14]];
+       t[15] = sbox[1][s[15]];
+
+       memset(s,0,16);
+       for (bit=0; bit < 128; bit++) s[ptab[bit][0]] |=  ((t[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+   }
+
+   memcpy(out,s,16);
+   return;
+}
+
+void indect_decrypt192(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2])
+{
+   unsigned char s[16];
+   unsigned char t[16];
+
+   unsigned int bit;
+   unsigned int round;
+
+   memcpy(s,in,16);
+
+   for (round=0; round < 10; round++) {
+
+       memset(t,0,16);
+       for (bit=0; bit < 128; bit++) t[ptab[bit][0]] |=  ((s[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+       s[0] = sbox[0][t[0]];
+       s[1] = sbox[0][t[1]];
+       s[2] = sbox[0][t[2]];
+       s[3] = sbox[0][t[3]];
+       s[4] = sbox[0][t[4]];
+       s[5] = sbox[0][t[5]];
+       s[6] = sbox[0][t[6]];
+       s[7] = sbox[0][t[7]];
+       s[8] = sbox[1][t[8]];
+       s[9] = sbox[1][t[9]];
+       s[10] = sbox[1][t[10]];
+       s[11] = sbox[1][t[11]];
+       s[12] = sbox[1][t[12]];
+       s[13] = sbox[1][t[13]];
+       s[14] = sbox[1][t[14]];
+       s[15] = sbox[1][t[15]];
+   }
+
+   memset(out,0,16);
+   for (bit=0; bit < 128; bit++) out[ptab[bit][0]] |=  ((s[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+   return;
+}
+
+void indect_encrypt320(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2])
+{
+   unsigned char s[16];
+   unsigned char t[16];
+
+   unsigned int bit;
```

```
+   unsigned int round;
+
+   memset(s,0,16);
+   for (bit=0; bit < 128; bit++) s[ptab[bit][0]] |=  ((in[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+   for (round=0; round < 12; round++) {
+
+       t[0] = sbox[0][s[0]];
+       t[1] = sbox[0][s[1]];
+       t[2] = sbox[0][s[2]];
+       t[3] = sbox[0][s[3]];
+       t[4] = sbox[1][s[4]];
+       t[5] = sbox[1][s[5]];
+       t[6] = sbox[1][s[6]];
+       t[7] = sbox[1][s[7]];
+       t[8] = sbox[2][s[8]];
+       t[9] = sbox[2][s[9]];
+       t[10] = sbox[2][s[10]];
+       t[11] = sbox[2][s[11]];
+       t[12] = sbox[3][s[12]];
+       t[13] = sbox[3][s[13]];
+       t[14] = sbox[3][s[14]];
+       t[15] = sbox[3][s[15]];
+
+       memset(s,0,16);
+       for (bit=0; bit < 128; bit++) s[ptab[bit][0]] |=  ((t[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+   }
+
+   memcpy(out,s,16);
+   return;
+}
+
+void indect_decrypt320(const unsigned char *in, unsigned char *out, const unsigned char
sbox[][256], const unsigned char ptab[][2])
+{
+   unsigned char s[16];
+   unsigned char t[16];
+
+   unsigned int bit;
+   unsigned int round;
+
+   memcpy(s,in,16);
+
+   for (round=0; round < 12; round++) {
+
+       memset(t,0,16);
+       for (bit=0; bit < 128; bit++) t[ptab[bit][0]] |=  ((s[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+       s[0] = sbox[0][t[0]];
+       s[1] = sbox[0][t[1]];
+       s[2] = sbox[0][t[2]];
+       s[3] = sbox[0][t[3]];
+       s[4] = sbox[1][t[4]];
+       s[5] = sbox[1][t[5]];
+       s[6] = sbox[1][t[6]];
+       s[7] = sbox[1][t[7]];
+       s[8] = sbox[2][t[8]];
+       s[9] = sbox[2][t[9]];
+       s[10] = sbox[2][t[10]];
+       s[11] = sbox[2][t[11]];
+       s[12] = sbox[3][t[12]];
+       s[13] = sbox[3][t[13]];
+       s[14] = sbox[3][t[14]];
+       s[15] = sbox[3][t[15]];
+   }
+
+   memset(out,0,16);
+   for (bit=0; bit < 128; bit++) out[ptab[bit][0]] |=  ((s[bit/8] << (bit%8)) & 128) >>
ptab[bit][1];
+
+   return;
+}
+
diff -rupN openssl-0.9.8v-org//crypto/indect/indect.h openssl-0.9.8v//crypto/indect/indect.h
```

```
--- openssl-0.9.8v-org//crypto/indect/indect.h  1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//crypto/indect/indect.h  2012-08-12 18:56:25.000000000 +0200
@@ -0,0 +1,132 @@
+/* crypto/indect/indect.h -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
+
+#ifndef HEADER_INDECT_H
+#define HEADER_INDECT_H
+
+#include <openssl/opensslconf.h>
+
+#ifdef OPENSSL_NO_INDECT
+#error INDECT is disabled.
+#endif
+
+#define INDECT_ENCRYPT 1
+#define INDECT_DECRYPT 0
+
+/* Because array size can't be a const in C, the following two are macros.
+   Both sizes are in bytes. */
+#define INDECT_BLOCK_SIZE 16
+#define INDECT_MAX_KEY_SIZE 576 // in bits
+#define INDECT_SBOXES_MAXNR (INDECT_MAX_KEY_SIZE / 64)
+
+
+
+
+#ifdef  __cplusplus
+extern "C" {
```

```
+#endif
+
+/* This should be a hidden type, but EVP requires that the size be known */
+
+
+
+struct indect_key_st
+   {
+   unsigned char sbox[INDECT_SBOXES_MAXNR][256];
+   unsigned char ptab[128][2];
+   int bitLength;
+   void (*enc)(const unsigned char *in, unsigned char *out, const unsigned char sbox[][256],
const unsigned char ptab[][2]);
+   void (*dec)(const unsigned char *in, unsigned char *out, const unsigned char sbox[][256],
const unsigned char ptab[][2]);
+   };
+typedef struct indect_key_st INDECT_KEY;
+
+int Indect_set_encrypt_key(const unsigned char *userKey, const int bits,
+   INDECT_KEY *key);
+int Indect_set_decrypt_key(const unsigned char *userKey, const int bits,
+   INDECT_KEY *key);
+
+void Indect_encrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key);
+void Indect_decrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key);
+
+void Indect_ecb_encrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key, const int enc);
+void Indect_cbc_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, const int enc);
+void Indect_cfb128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc);
+void Indect_cfb1_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc);
+void Indect_cfb8_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc);
+void Indect_cfbr_encrypt_block(const unsigned char *in,unsigned char *out,
+   const int nbits,const INDECT_KEY *key,
+   unsigned char *ivec,const int enc);
+void Indect_ofb128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num);
+void Indect_ctr128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char ivec[INDECT_BLOCK_SIZE],
+   unsigned char ecount_buf[INDECT_BLOCK_SIZE],
+   unsigned int *num);
+
+#ifdef  __cplusplus
+}
+#endif
+
+#endif /* !HEADER_Indect_H */
+
diff -rupN openssl-0.9.8v-org//crypto/install.com openssl-0.9.8v//crypto/install.com
--- openssl-0.9.8v-org//crypto/install.com  2009-08-25 09:28:18.000000000 +0200
+++ openssl-0.9.8v//crypto/install.com  2012-04-20 23:36:04.000000000 +0200
@@ -43,7 +43,7 @@ $
 $  SDIRS := ,-
         OBJECTS,-
         MD2,MD4,MD5,SHA,MDC2,HMAC,RIPEMD,-
-        DES,AES,RC2,RC4,RC5,IDEA,BF,CAST,CAMELLIA,SEED,-
+        DES,AES,RC2,RC4,RC5,IDEA,BF,CAST,CAMELLIA,INDECT,SEED,-
         BN,EC,RSA,DSA,ECDSA,DH,ECDH,DSO,ENGINE,-
         BUFFER,BIO,STACK,LHASH,RAND,ERR,-
         EVP,ASN1,PEM,X509,X509V3,CONF,TXT_DB,PKCS7,PKCS12,COMP,OCSP,-
@@ -68,6 +68,7 @@ $ EXHEADER_IDEA := idea.h
 $  EXHEADER_BF := blowfish.h
 $  EXHEADER_CAST := cast.h
 $  EXHEADER_CAMELLIA := camellia.h
+$  EXHEADER_INDECT := indect.h
```

```
 $  EXHEADER_SEED := seed.h
 $  EXHEADER_BN := bn.h
 $  EXHEADER_EC := ec.h
diff -rupN openssl-0.9.8v-org//crypto/objects/obj_dat.h openssl-
0.9.8v//crypto/objects/obj_dat.h
--- openssl-0.9.8v-org//crypto/objects/obj_dat.h    2010-01-25 17:08:51.000000000 +0100
+++ openssl-0.9.8v//crypto/objects/obj_dat.h    2012-08-12 18:52:09.000000000 +0200
@@ -62,12 +62,12 @@
  * [including the GNU Public Licence.]
  */

-#define NUM_NID 893
-#define NUM_SN 886
-#define NUM_LN 886
-#define NUM_OBJ 840
+#define NUM_NID 911
+#define NUM_SN 904
+#define NUM_LN 904
+#define NUM_OBJ 852

-static unsigned char lvalues[5824]={
+static unsigned char lvalues[5956]={
 0x00,                                          /* [  0] OBJ_undef */
 0x2A,0x86,0x48,0x86,0xF7,0x0D,                 /* [  1] OBJ_rsadsi */
 0x2A,0x86,0x48,0x86,0xF7,0x0D,0x01,            /* [  7] OBJ_pkcs */
@@ -908,6 +908,18 @@ static unsigned char lvalues[5824]={
 0x55,0x04,0x34,                                /* [5814] OBJ_supportedAlgorithms */
 0x55,0x04,0x35,                                /* [5817] OBJ_deltaRevocationList */
 0x55,0x04,0x36,                                /* [5820] OBJ_dmdName */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x01,/* [5823] OBJ_indect_128_ecb */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x02,/* [5834] OBJ_indect_128_cbc */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x03,/* [5845] OBJ_indect_128_ofb128 */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x04,/* [5856] OBJ_indect_128_cfb128 */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x15,/* [5867] OBJ_indect_192_ecb */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x16,/* [5878] OBJ_indect_192_cbc */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x17,/* [5889] OBJ_indect_192_ofb128 */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x18,/* [5900] OBJ_indect_192_cfb128 */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x29,/* [5911] OBJ_indect_320_ecb */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x2A,/* [5922] OBJ_indect_320_cbc */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x2B,/* [5933] OBJ_indect_320_ofb128 */
+0x2B,0x06,0x01,0x04,0x01,0x82,0xB0,0x44,0x85,0x1A,0x2C,/* [5944] OBJ_indect_320_cfb128 */
 };

 static ASN1_OBJECT nid_objs[NUM_NID]={
@@ -2351,6 +2363,36 @@ static ASN1_OBJECT nid_objs[NUM_NID]={
 {"deltaRevocationList","deltaRevocationList",NID_deltaRevocationList,
    3,&(lvalues[5817]),0},
 {"dmdName","dmdName",NID_dmdName,3,&(lvalues[5820]),0},
+{"INDECT-128-ECB","indect-128-ecb",NID_indect_128_ecb,11,
+   &(lvalues[5823]),0},
+{"INDECT-128-CBC","indect-128-cbc",NID_indect_128_cbc,11,
+   &(lvalues[5834]),0},
+{"INDECT-128-OFB","indect-128-ofb",NID_indect_128_ofb128,11,
+   &(lvalues[5845]),0},
+{"INDECT-128-CFB","indect-128-cfb",NID_indect_128_cfb128,11,
+   &(lvalues[5856]),0},
+{"INDECT-192-ECB","indect-192-ecb",NID_indect_192_ecb,11,
+   &(lvalues[5867]),0},
+{"INDECT-192-CBC","indect-192-cbc",NID_indect_192_cbc,11,
+   &(lvalues[5878]),0},
+{"INDECT-192-OFB","indect-192-ofb",NID_indect_192_ofb128,11,
+   &(lvalues[5889]),0},
+{"INDECT-192-CFB","indect-192-cfb",NID_indect_192_cfb128,11,
+   &(lvalues[5900]),0},
+{"INDECT-320-ECB","indect-320-ecb",NID_indect_320_ecb,11,
+   &(lvalues[5911]),0},
+{"INDECT-320-CBC","indect-320-cbc",NID_indect_320_cbc,11,
+   &(lvalues[5922]),0},
+{"INDECT-320-OFB","indect-320-ofb",NID_indect_320_ofb128,11,
+   &(lvalues[5933]),0},
+{"INDECT-320-CFB","indect-320-cfb",NID_indect_320_cfb128,11,
+   &(lvalues[5944]),0},
+{"INDECT-128-CFB1","indect-128-cfb1",NID_indect_128_cfb1,0,NULL,0},
+{"INDECT-192-CFB1","indect-192-cfb1",NID_indect_192_cfb1,0,NULL,0},
+{"INDECT-320-CFB1","indect-320-cfb1",NID_indect_320_cfb1,0,NULL,0},
+{"INDECT-128-CFB8","indect-128-cfb8",NID_indect_128_cfb8,0,NULL,0},
+{"INDECT-192-CFB8","indect-192-cfb8",NID_indect_192_cfb8,0,NULL,0},
```

```
+{"INDECT-320-CFB8","indect-320-cfb8",NID_indect_320_cfb8,0,NULL,0},
 };

 static ASN1_OBJECT *sn_objs[NUM_SN]={
@@ -2439,6 +2481,24 @@ static ASN1_OBJECT *sn_objs[NUM_SN]={
 &(nid_objs[35]),/* "IDEA-CFB" */
 &(nid_objs[36]),/* "IDEA-ECB" */
 &(nid_objs[46]),/* "IDEA-OFB" */
+&(nid_objs[894]),/* "INDECT-128-CBC" */
+&(nid_objs[896]),/* "INDECT-128-CFB" */
+&(nid_objs[905]),/* "INDECT-128-CFB1" */
+&(nid_objs[908]),/* "INDECT-128-CFB8" */
+&(nid_objs[893]),/* "INDECT-128-ECB" */
+&(nid_objs[895]),/* "INDECT-128-OFB" */
+&(nid_objs[898]),/* "INDECT-192-CBC" */
+&(nid_objs[900]),/* "INDECT-192-CFB" */
+&(nid_objs[906]),/* "INDECT-192-CFB1" */
+&(nid_objs[909]),/* "INDECT-192-CFB8" */
+&(nid_objs[897]),/* "INDECT-192-ECB" */
+&(nid_objs[899]),/* "INDECT-192-OFB" */
+&(nid_objs[902]),/* "INDECT-320-CBC" */
+&(nid_objs[904]),/* "INDECT-320-CFB" */
+&(nid_objs[907]),/* "INDECT-320-CFB1" */
+&(nid_objs[910]),/* "INDECT-320-CFB8" */
+&(nid_objs[901]),/* "INDECT-320-ECB" */
+&(nid_objs[903]),/* "INDECT-320-OFB" */
 &(nid_objs[181]),/* "ISO" */
 &(nid_objs[183]),/* "ISO-US" */
 &(nid_objs[645]),/* "ITU-T" */
@@ -3773,6 +3833,24 @@ static ASN1_OBJECT *ln_objs[NUM_LN]={
 &(nid_objs[36]),/* "idea-ecb" */
 &(nid_objs[46]),/* "idea-ofb" */
 &(nid_objs[676]),/* "identified-organization" */
+&(nid_objs[894]),/* "indect-128-cbc" */
+&(nid_objs[896]),/* "indect-128-cfb" */
+&(nid_objs[905]),/* "indect-128-cfb1" */
+&(nid_objs[908]),/* "indect-128-cfb8" */
+&(nid_objs[893]),/* "indect-128-ecb" */
+&(nid_objs[895]),/* "indect-128-ofb" */
+&(nid_objs[898]),/* "indect-192-cbc" */
+&(nid_objs[900]),/* "indect-192-cfb" */
+&(nid_objs[906]),/* "indect-192-cfb1" */
+&(nid_objs[909]),/* "indect-192-cfb8" */
+&(nid_objs[897]),/* "indect-192-ecb" */
+&(nid_objs[899]),/* "indect-192-ofb" */
+&(nid_objs[902]),/* "indect-320-cbc" */
+&(nid_objs[904]),/* "indect-320-cfb" */
+&(nid_objs[907]),/* "indect-320-cfb1" */
+&(nid_objs[910]),/* "indect-320-cfb8" */
+&(nid_objs[901]),/* "indect-320-ecb" */
+&(nid_objs[903]),/* "indect-320-ofb" */
 &(nid_objs[461]),/* "info" */
 &(nid_objs[101]),/* "initials" */
 &(nid_objs[869]),/* "internationaliSDNNumber" */
@@ -4972,5 +5050,17 @@ static ASN1_OBJECT *obj_objs[NUM_OBJ]={
 &(nid_objs[154]),/* OBJ_secretBag                     1 2 840 113549 1 12 10 1 5 */
 &(nid_objs[155]),/* OBJ_safeContentsBag               1 2 840 113549 1 12 10 1 6 */
 &(nid_objs[34]),/* OBJ_idea_cbc                      1 3 6 1 4 1 188 7 1 1 2 */
+&(nid_objs[893]),/* OBJ_indect_128_ecb                1 3 6 1 4 1 38980 666 1 */
+&(nid_objs[894]),/* OBJ_indect_128_cbc                1 3 6 1 4 1 38980 666 2 */
+&(nid_objs[895]),/* OBJ_indect_128_ofb128             1 3 6 1 4 1 38980 666 3 */
+&(nid_objs[896]),/* OBJ_indect_128_cfb128             1 3 6 1 4 1 38980 666 4 */
+&(nid_objs[897]),/* OBJ_indect_192_ecb                1 3 6 1 4 1 38980 666 21 */
+&(nid_objs[898]),/* OBJ_indect_192_cbc                1 3 6 1 4 1 38980 666 22 */
+&(nid_objs[899]),/* OBJ_indect_192_ofb128             1 3 6 1 4 1 38980 666 23 */
+&(nid_objs[900]),/* OBJ_indect_192_cfb128             1 3 6 1 4 1 38980 666 24 */
+&(nid_objs[901]),/* OBJ_indect_320_ecb                1 3 6 1 4 1 38980 666 41 */
+&(nid_objs[902]),/* OBJ_indect_320_cbc                1 3 6 1 4 1 38980 666 42 */
+&(nid_objs[903]),/* OBJ_indect_320_ofb128             1 3 6 1 4 1 38980 666 43 */
+&(nid_objs[904]),/* OBJ_indect_320_cfb128             1 3 6 1 4 1 38980 666 44 */
 };

diff -rupN openssl-0.9.8v-org//crypto/objects/objects.txt openssl-
0.9.8v//crypto/objects/objects.txt
--- openssl-0.9.8v-org//crypto/objects/objects.txt  2010-01-25 17:08:01.000000000 +0100
+++ openssl-0.9.8v//crypto/objects/objects.txt  2012-08-12 18:35:37.000000000 +0200
@@ -1243,6 +1243,40 @@ camellia 44      : CAMELLIA-256-CFB      : camel
```

```
            : CAMELLIA-128-CFB8     : camellia-128-cfb8
            : CAMELLIA-192-CFB8     : camellia-192-cfb8
            : CAMELLIA-256-CFB8     : camellia-256-cfb8
+
+# Definitions for Indect cipher - ECB, CBC, CFB, OFB MODE
+
+!Alias agh-indect 1 3 6 1 4 1 38980 666
+
+agh-indect 1       : INDECT-128-ECB        : indect-128-ecb
+agh-indect 2       : INDECT-128-CBC        : indect-128-cbc
+!Cname indect-128-ofb128
+agh-indect 3       : INDECT-128-OFB        : indect-128-ofb
+!Cname indect-128-cfb128
+agh-indect 4       : INDECT-128-CFB        : indect-128-cfb
+
+agh-indect 21      : INDECT-192-ECB        : indect-192-ecb
+agh-indect 22      : INDECT-192-CBC        : indect-192-cbc
+!Cname indect-192-ofb128
+agh-indect 23      : INDECT-192-OFB        : indect-192-ofb
+!Cname indect-192-cfb128
+agh-indect 24      : INDECT-192-CFB        : indect-192-cfb
+
+agh-indect 41      : INDECT-320-ECB        : indect-320-ecb
+agh-indect 42      : INDECT-320-CBC        : indect-320-cbc
+!Cname indect-320-ofb128
+agh-indect 43      : INDECT-320-OFB        : indect-320-ofb
+!Cname indect-320-cfb128
+agh-indect 44      : INDECT-320-CFB        : indect-320-cfb
+
+# There are no OIDs for these Indect modes...
+
+           : INDECT-128-CFB1       : indect-128-cfb1
+           : INDECT-192-CFB1       : indect-192-cfb1
+           : INDECT-320-CFB1       : indect-320-cfb1
+           : INDECT-128-CFB8       : indect-128-cfb8
+           : INDECT-192-CFB8       : indect-192-cfb8
+           : INDECT-320-CFB8       : indect-320-cfb8

 # Definitions for SEED cipher - ECB, CBC, OFB mode
```

```
diff -rupN openssl-0.9.8v-org//crypto/objects/obj_mac.h openssl-
0.9.8v//crypto/objects/obj_mac.h
--- openssl-0.9.8v-org//crypto/objects/obj_mac.h    2010-01-25 17:08:52.000000000 +0100
+++ openssl-0.9.8v//crypto/objects/obj_mac.h    2012-08-12 18:52:09.000000000 +0200
@@ -3883,6 +3883,92 @@
 #define LN_camellia_256_cfb8       "camellia-256-cfb8"
 #define NID_camellia_256_cfb8      765

+#define OBJ_agh_indect     1L,3L,6L,1L,4L,1L,38980L,666L
+
+#define SN_indect_128_ecb      "INDECT-128-ECB"
+#define LN_indect_128_ecb      "indect-128-ecb"
+#define NID_indect_128_ecb     893
+#define OBJ_indect_128_ecb     OBJ_agh_indect,1L
+
+#define SN_indect_128_cbc      "INDECT-128-CBC"
+#define LN_indect_128_cbc      "indect-128-cbc"
+#define NID_indect_128_cbc     894
+#define OBJ_indect_128_cbc     OBJ_agh_indect,2L
+
+#define SN_indect_128_ofb128       "INDECT-128-OFB"
+#define LN_indect_128_ofb128       "indect-128-ofb"
+#define NID_indect_128_ofb128      895
+#define OBJ_indect_128_ofb128      OBJ_agh_indect,3L
+
+#define SN_indect_128_cfb128       "INDECT-128-CFB"
+#define LN_indect_128_cfb128       "indect-128-cfb"
+#define NID_indect_128_cfb128      896
+#define OBJ_indect_128_cfb128      OBJ_agh_indect,4L
+
+#define SN_indect_192_ecb      "INDECT-192-ECB"
+#define LN_indect_192_ecb      "indect-192-ecb"
+#define NID_indect_192_ecb     897
+#define OBJ_indect_192_ecb     OBJ_agh_indect,21L
+
+#define SN_indect_192_cbc      "INDECT-192-CBC"
+#define LN_indect_192_cbc      "indect-192-cbc"
```

```
+#define NID_indect_192_cbc     898
+#define OBJ_indect_192_cbc     OBJ_agh_indect,22L
+
+#define SN_indect_192_ofb128       "INDECT-192-OFB"
+#define LN_indect_192_ofb128       "indect-192-ofb"
+#define NID_indect_192_ofb128      899
+#define OBJ_indect_192_ofb128      OBJ_agh_indect,23L
+
+#define SN_indect_192_cfb128       "INDECT-192-CFB"
+#define LN_indect_192_cfb128       "indect-192-cfb"
+#define NID_indect_192_cfb128      900
+#define OBJ_indect_192_cfb128      OBJ_agh_indect,24L
+
+#define SN_indect_320_ecb      "INDECT-320-ECB"
+#define LN_indect_320_ecb      "indect-320-ecb"
+#define NID_indect_320_ecb     901
+#define OBJ_indect_320_ecb     OBJ_agh_indect,41L
+
+#define SN_indect_320_cbc      "INDECT-320-CBC"
+#define LN_indect_320_cbc      "indect-320-cbc"
+#define NID_indect_320_cbc     902
+#define OBJ_indect_320_cbc     OBJ_agh_indect,42L
+
+#define SN_indect_320_ofb128       "INDECT-320-OFB"
+#define LN_indect_320_ofb128       "indect-320-ofb"
+#define NID_indect_320_ofb128      903
+#define OBJ_indect_320_ofb128      OBJ_agh_indect,43L
+
+#define SN_indect_320_cfb128       "INDECT-320-CFB"
+#define LN_indect_320_cfb128       "indect-320-cfb"
+#define NID_indect_320_cfb128      904
+#define OBJ_indect_320_cfb128      OBJ_agh_indect,44L
+
+#define SN_indect_128_cfb1     "INDECT-128-CFB1"
+#define LN_indect_128_cfb1     "indect-128-cfb1"
+#define NID_indect_128_cfb1        905
+
+#define SN_indect_192_cfb1     "INDECT-192-CFB1"
+#define LN_indect_192_cfb1     "indect-192-cfb1"
+#define NID_indect_192_cfb1        906
+
+#define SN_indect_320_cfb1     "INDECT-320-CFB1"
+#define LN_indect_320_cfb1     "indect-320-cfb1"
+#define NID_indect_320_cfb1        907
+
+#define SN_indect_128_cfb8     "INDECT-128-CFB8"
+#define LN_indect_128_cfb8     "indect-128-cfb8"
+#define NID_indect_128_cfb8        908
+
+#define SN_indect_192_cfb8     "INDECT-192-CFB8"
+#define LN_indect_192_cfb8     "indect-192-cfb8"
+#define NID_indect_192_cfb8        909
+
+#define SN_indect_320_cfb8     "INDECT-320-CFB8"
+#define LN_indect_320_cfb8     "indect-320-cfb8"
+#define NID_indect_320_cfb8        910
+
 #define SN_kisa         "KISA"
 #define LN_kisa         "kisa"
 #define NID_kisa        773
diff -rupN openssl-0.9.8v-org//crypto/objects/obj_mac.num openssl-
0.9.8v//crypto/objects/obj_mac.num
--- openssl-0.9.8v-org//crypto/objects/obj_mac.num  2009-03-25 20:01:03.000000000 +0100
+++ openssl-0.9.8v//crypto/objects/obj_mac.num  2012-08-12 18:52:09.000000000 +0200
@@ -890,3 +890,21 @@ houseIdentifier         889
 supportedAlgorithms        890
 deltaRevocationList        891
 dmdName        892
+indect_128_ecb     893
+indect_128_cbc     894
+indect_128_ofb128      895
+indect_128_cfb128      896
+indect_192_ecb     897
+indect_192_cbc     898
+indect_192_ofb128      899
+indect_192_cfb128      900
+indect_320_ecb     901
```

```
+indect_320_cbc     902
+indect_320_ofb128      903
+indect_320_cfb128      904
+indect_128_cfb1        905
+indect_192_cfb1        906
+indect_320_cfb1        907
+indect_128_cfb8        908
+indect_192_cfb8        909
+indect_320_cfb8        910
diff -rupN openssl-0.9.8v-org//crypto/opensslv.h openssl-0.9.8v//crypto/opensslv.h
--- openssl-0.9.8v-org//crypto/opensslv.h   2012-04-19 13:39:03.000000000 +0200
+++ openssl-0.9.8v//crypto/opensslv.h   2012-08-12 18:36:27.000000000 +0200
@@ -27,9 +27,9 @@
  */
 #define OPENSSL_VERSION_NUMBER 0x0090816fL
 #ifdef OPENSSL_FIPS
-#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v-fips 19 Apr 2012"
+#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v-fips 19 Apr 2012 + Indect 1.2 12 Aug 2012"
 #else
-#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v 19 Apr 2012"
+#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v 19 Apr 2012 + Indect 1.2 12 Aug 2012"
 #endif
 #define OPENSSL_VERSION_PTEXT  " part of " OPENSSL_VERSION_TEXT

diff -rupN openssl-0.9.8v-org//include/openssl/evp.h openssl-0.9.8v//include/openssl/evp.h
--- openssl-0.9.8v-org//include/openssl/evp.h   2008-09-17 19:11:00.000000000 +0200
+++ openssl-0.9.8v//include/openssl/evp.h   2012-08-12 18:32:50.000000000 +0200
@@ -87,7 +87,7 @@
 #define EVP_RC5_32_12_16_KEY_SIZE  16
 */
 #define EVP_MAX_MD_SIZE            64  /* longest known is SHA512 */
-#define EVP_MAX_KEY_LENGTH     32
+#define EVP_MAX_KEY_LENGTH     40
 #define EVP_MAX_IV_LENGTH      16
 #define EVP_MAX_BLOCK_LENGTH       32

@@ -815,6 +815,30 @@ const EVP_CIPHER *EVP_camellia_256_cfb12
 const EVP_CIPHER *EVP_camellia_256_ofb(void);
 #endif

+#ifndef OPENSSL_NO_INDECT
+const EVP_CIPHER *EVP_indect_128_ecb(void);
+const EVP_CIPHER *EVP_indect_128_cbc(void);
+const EVP_CIPHER *EVP_indect_128_cfb1(void);
+const EVP_CIPHER *EVP_indect_128_cfb8(void);
+const EVP_CIPHER *EVP_indect_128_cfb128(void);
+# define EVP_indect_128_cfb EVP_indect_128_cfb128
+const EVP_CIPHER *EVP_indect_128_ofb(void);
+const EVP_CIPHER *EVP_indect_192_ecb(void);
+const EVP_CIPHER *EVP_indect_192_cbc(void);
+const EVP_CIPHER *EVP_indect_192_cfb1(void);
+const EVP_CIPHER *EVP_indect_192_cfb8(void);
+const EVP_CIPHER *EVP_indect_192_cfb128(void);
+# define EVP_indect_192_cfb EVP_indect_192_cfb128
+const EVP_CIPHER *EVP_indect_192_ofb(void);
+const EVP_CIPHER *EVP_indect_320_ecb(void);
+const EVP_CIPHER *EVP_indect_320_cbc(void);
+const EVP_CIPHER *EVP_indect_320_cfb1(void);
+const EVP_CIPHER *EVP_indect_320_cfb8(void);
+const EVP_CIPHER *EVP_indect_320_cfb128(void);
+# define EVP_indect_320_cfb EVP_indect_320_cfb128
+const EVP_CIPHER *EVP_indect_320_ofb(void);
+#endif
+
 #ifndef OPENSSL_NO_SEED
 const EVP_CIPHER *EVP_seed_ecb(void);
 const EVP_CIPHER *EVP_seed_cbc(void);
@@ -993,6 +1017,7 @@ void ERR_load_EVP_strings(void);
 #define EVP_F_EVP_RIJNDAEL              126
 #define EVP_F_EVP_SIGNFINAL             107
 #define EVP_F_EVP_VERIFYFINAL           108
+#define EVP_F_INDECT_INIT_KEY           143
 #define EVP_F_PKCS5_PBE_KEYIVGEN        117
 #define EVP_F_PKCS5_V2_PBE_KEYIVGEN         118
 #define EVP_F_PKCS8_SET_BROKEN          112
@@ -1025,6 +1050,7 @@ void ERR_load_EVP_strings(void);
 #define EVP_R_EXPECTING_A_ECDSA_KEY         141
```

```
 #define EVP_R_EXPECTING_A_EC_KEY                142
 #define EVP_R_FIPS_MODE_NOT_SUPPORTED           147
+#define EVP_R_INDECT_KEY_SETUP_FAILED           150
 #define EVP_R_INITIALIZATION_ERROR              134
 #define EVP_R_INPUT_NOT_INITIALIZED             111
 #define EVP_R_INVALID_FIPS_MODE                 148
@@ -1039,6 +1065,7 @@ void ERR_load_EVP_strings(void);
 #define EVP_R_NO_VERIFY_FUNCTION_CONFIGURED         105
 #define EVP_R_PKCS8_UNKNOWN_BROKEN_TYPE             117
 #define EVP_R_PUBLIC_KEY_NOT_RSA                106
+#define EVP_R_SEED_KEY_SETUP_FAILED             162
 #define EVP_R_UNKNOWN_OPTION                    149
 #define EVP_R_UNKNOWN_PBE_ALGORITHM             121
 #define EVP_R_UNSUPORTED_NUMBER_OF_ROUNDS       135
@@ -1051,7 +1078,6 @@ void ERR_load_EVP_strings(void);
 #define EVP_R_UNSUPPORTED_SALT_TYPE             126
 #define EVP_R_WRONG_FINAL_BLOCK_LENGTH          109
 #define EVP_R_WRONG_PUBLIC_KEY_TYPE             110
-#define EVP_R_SEED_KEY_SETUP_FAILED             162

 #ifdef  __cplusplus
 }
diff -rupN openssl-0.9.8v-org//include/openssl/indect.h openssl-
0.9.8v//include/openssl/indect.h
--- openssl-0.9.8v-org//include/openssl/indect.h    1970-01-01 01:00:00.000000000 +0100
+++ openssl-0.9.8v//include/openssl/indect.h    2012-08-12 18:56:25.000000000 +0200
@@ -0,0 +1,132 @@
+/* crypto/indect/indect.h -*- mode:C; c-file-style: "eay" -*- */
+/* ====================================================================
+ * Copyright (c) 2006 The OpenSSL Project.  All rights reserved.
+ *
+ * Redistribution and use in source and binary forms, with or without
+ * modification, are permitted provided that the following conditions
+ * are met:
+ *
+ * 1. Redistributions of source code must retain the above copyright
+ *    notice, this list of conditions and the following disclaimer.
+ *
+ * 2. Redistributions in binary form must reproduce the above copyright
+ *    notice, this list of conditions and the following disclaimer in
+ *    the documentation and/or other materials provided with the
+ *    distribution.
+ *
+ * 3. All advertising materials mentioning features or use of this
+ *    software must display the following acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
+ *
+ * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
+ *    endorse or promote products derived from this software without
+ *    prior written permission. For written permission, please contact
+ *    openssl-core@openssl.org.
+ *
+ * 5. Products derived from this software may not be called "OpenSSL"
+ *    nor may "OpenSSL" appear in their names without prior written
+ *    permission of the OpenSSL Project.
+ *
+ * 6. Redistributions of any form whatsoever must retain the following
+ *    acknowledgment:
+ *    "This product includes software developed by the OpenSSL Project
+ *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
+ *
+ * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
+ * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
+ * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
+ * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
+ * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
+ * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
+ * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
+ * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
+ * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
+ * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
+ * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
+ * OF THE POSSIBILITY OF SUCH DAMAGE.
+ * ====================================================================
+ *
+ */
```

```
+
+#ifndef HEADER_INDECT_H
+#define HEADER_INDECT_H
+
+#include <openssl/opensslconf.h>
+
+#ifdef OPENSSL_NO_INDECT
+#error INDECT is disabled.
+#endif
+
+#define INDECT_ENCRYPT 1
+#define INDECT_DECRYPT 0
+
+/* Because array size can't be a const in C, the following two are macros.
+   Both sizes are in bytes. */
+#define INDECT_BLOCK_SIZE 16
+#define INDECT_MAX_KEY_SIZE 576 // in bits
+#define INDECT_SBOXES_MAXNR (INDECT_MAX_KEY_SIZE / 64)
+
+
+
+
+#ifdef  __cplusplus
+extern "C" {
+#endif
+
+/* This should be a hidden type, but EVP requires that the size be known */
+
+
+
+struct indect_key_st
+   {
+   unsigned char sbox[INDECT_SBOXES_MAXNR][256];
+   unsigned char ptab[128][2];
+   int bitLength;
+   void (*enc)(const unsigned char *in, unsigned char *out, const unsigned char sbox[][256],
const unsigned char ptab[][2]);
+   void (*dec)(const unsigned char *in, unsigned char *out, const unsigned char sbox[][256],
const unsigned char ptab[][2]);
+   };
+typedef struct indect_key_st INDECT_KEY;
+
+int Indect_set_encrypt_key(const unsigned char *userKey, const int bits,
+   INDECT_KEY *key);
+int Indect_set_decrypt_key(const unsigned char *userKey, const int bits,
+   INDECT_KEY *key);
+
+void Indect_encrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key);
+void Indect_decrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key);
+
+void Indect_ecb_encrypt(const unsigned char *in, unsigned char *out,
+   const INDECT_KEY *key, const int enc);
+void Indect_cbc_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, const int enc);
+void Indect_cfb128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc);
+void Indect_cfb1_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc);
+void Indect_cfb8_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num, const int enc);
+void Indect_cfbr_encrypt_block(const unsigned char *in,unsigned char *out,
+   const int nbits,const INDECT_KEY *key,
+   unsigned char *ivec,const int enc);
+void Indect_ofb128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char *ivec, int *num);
+void Indect_ctr128_encrypt(const unsigned char *in, unsigned char *out,
+   const unsigned long length, const INDECT_KEY *key,
+   unsigned char ivec[INDECT_BLOCK_SIZE],
+   unsigned char ecount_buf[INDECT_BLOCK_SIZE],
+   unsigned int *num);
```

```
+
+#ifdef  __cplusplus
+}
+#endif
+
+#endif /* !HEADER_Indect_H */
+
diff -rupN openssl-0.9.8v-org//include/openssl/obj_mac.h openssl-
0.9.8v//include/openssl/obj_mac.h
--- openssl-0.9.8v-org//include/openssl/obj_mac.h   2010-01-25 17:08:52.000000000 +0100
+++ openssl-0.9.8v//include/openssl/obj_mac.h   2012-08-12 18:52:09.000000000 +0200
@@ -3883,6 +3883,92 @@
 #define LN_camellia_256_cfb8       "camellia-256-cfb8"
 #define NID_camellia_256_cfb8      765

+#define OBJ_agh_indect     1L,3L,6L,1L,4L,1L,38980L,666L
+
+#define SN_indect_128_ecb      "INDECT-128-ECB"
+#define LN_indect_128_ecb      "indect-128-ecb"
+#define NID_indect_128_ecb     893
+#define OBJ_indect_128_ecb     OBJ_agh_indect,1L
+
+#define SN_indect_128_cbc      "INDECT-128-CBC"
+#define LN_indect_128_cbc      "indect-128-cbc"
+#define NID_indect_128_cbc     894
+#define OBJ_indect_128_cbc     OBJ_agh_indect,2L
+
+#define SN_indect_128_ofb128       "INDECT-128-OFB"
+#define LN_indect_128_ofb128       "indect-128-ofb"
+#define NID_indect_128_ofb128      895
+#define OBJ_indect_128_ofb128      OBJ_agh_indect,3L
+
+#define SN_indect_128_cfb128       "INDECT-128-CFB"
+#define LN_indect_128_cfb128       "indect-128-cfb"
+#define NID_indect_128_cfb128      896
+#define OBJ_indect_128_cfb128      OBJ_agh_indect,4L
+
+#define SN_indect_192_ecb      "INDECT-192-ECB"
+#define LN_indect_192_ecb      "indect-192-ecb"
+#define NID_indect_192_ecb     897
+#define OBJ_indect_192_ecb     OBJ_agh_indect,21L
+
+#define SN_indect_192_cbc      "INDECT-192-CBC"
+#define LN_indect_192_cbc      "indect-192-cbc"
+#define NID_indect_192_cbc     898
+#define OBJ_indect_192_cbc     OBJ_agh_indect,22L
+
+#define SN_indect_192_ofb128       "INDECT-192-OFB"
+#define LN_indect_192_ofb128       "indect-192-ofb"
+#define NID_indect_192_ofb128      899
+#define OBJ_indect_192_ofb128      OBJ_agh_indect,23L
+
+#define SN_indect_192_cfb128       "INDECT-192-CFB"
+#define LN_indect_192_cfb128       "indect-192-cfb"
+#define NID_indect_192_cfb128      900
+#define OBJ_indect_192_cfb128      OBJ_agh_indect,24L
+
+#define SN_indect_320_ecb      "INDECT-320-ECB"
+#define LN_indect_320_ecb      "indect-320-ecb"
+#define NID_indect_320_ecb     901
+#define OBJ_indect_320_ecb     OBJ_agh_indect,41L
+
+#define SN_indect_320_cbc      "INDECT-320-CBC"
+#define LN_indect_320_cbc      "indect-320-cbc"
+#define NID_indect_320_cbc     902
+#define OBJ_indect_320_cbc     OBJ_agh_indect,42L
+
+#define SN_indect_320_ofb128       "INDECT-320-OFB"
+#define LN_indect_320_ofb128       "indect-320-ofb"
+#define NID_indect_320_ofb128      903
+#define OBJ_indect_320_ofb128      OBJ_agh_indect,43L
+
+#define SN_indect_320_cfb128       "INDECT-320-CFB"
+#define LN_indect_320_cfb128       "indect-320-cfb"
+#define NID_indect_320_cfb128      904
+#define OBJ_indect_320_cfb128      OBJ_agh_indect,44L
+
```

```
+#define SN_indect_128_cfb1     "INDECT-128-CFB1"
+#define LN_indect_128_cfb1     "indect-128-cfb1"
+#define NID_indect_128_cfb1        905
+
+#define SN_indect_192_cfb1     "INDECT-192-CFB1"
+#define LN_indect_192_cfb1     "indect-192-cfb1"
+#define NID_indect_192_cfb1        906
+
+#define SN_indect_320_cfb1     "INDECT-320-CFB1"
+#define LN_indect_320_cfb1     "indect-320-cfb1"
+#define NID_indect_320_cfb1        907
+
+#define SN_indect_128_cfb8     "INDECT-128-CFB8"
+#define LN_indect_128_cfb8     "indect-128-cfb8"
+#define NID_indect_128_cfb8        908
+
+#define SN_indect_192_cfb8     "INDECT-192-CFB8"
+#define LN_indect_192_cfb8     "indect-192-cfb8"
+#define NID_indect_192_cfb8        909
+
+#define SN_indect_320_cfb8     "INDECT-320-CFB8"
+#define LN_indect_320_cfb8     "indect-320-cfb8"
+#define NID_indect_320_cfb8        910
+
 #define SN_kisa         "KISA"
 #define LN_kisa         "kisa"
 #define NID_kisa        773
diff -rupN openssl-0.9.8v-org//include/openssl/opensslv.h openssl-
0.9.8v//include/openssl/opensslv.h
--- openssl-0.9.8v-org//include/openssl/opensslv.h  2012-04-19 13:39:03.000000000 +0200
+++ openssl-0.9.8v//include/openssl/opensslv.h  2012-08-12 18:36:27.000000000 +0200
@@ -27,9 +27,9 @@
  */
 #define OPENSSL_VERSION_NUMBER 0x0090816fL
 #ifdef OPENSSL_FIPS
-#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v-fips 19 Apr 2012"
+#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v-fips 19 Apr 2012 + Indect 1.2 12 Aug 2012"
 #else
-#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v 19 Apr 2012"
+#define OPENSSL_VERSION_TEXT   "OpenSSL 0.9.8v 19 Apr 2012 + Indect 1.2 12 Aug 2012"
 #endif
 #define OPENSSL_VERSION_PTEXT  " part of " OPENSSL_VERSION_TEXT

diff -rupN openssl-0.9.8v-org//include/openssl/ssl.h openssl-0.9.8v//include/openssl/ssl.h
--- openssl-0.9.8v-org//include/openssl/ssl.h   2012-03-12 15:50:55.000000000 +0100
+++ openssl-0.9.8v//include/openssl/ssl.h   2012-08-12 18:37:32.000000000 +0200
@@ -286,6 +286,7 @@ extern "C" {
 #define SSL_TXT_SEED       "SEED"
 #define SSL_TXT_AES        "AES"
 #define SSL_TXT_CAMELLIA   "CAMELLIA"
+#define SSL_TXT_INDECT     "INDECT"
 #define SSL_TXT_MD5        "MD5"
 #define SSL_TXT_SHA1       "SHA1"
 #define SSL_TXT_SHA        "SHA"
diff -rupN openssl-0.9.8v-org//include/openssl/tls1.h openssl-0.9.8v//include/openssl/tls1.h
--- openssl-0.9.8v-org//include/openssl/tls1.h  2009-11-08 15:51:54.000000000 +0100
+++ openssl-0.9.8v//include/openssl/tls1.h  2012-05-09 22:58:53.000000000 +0200
@@ -230,6 +230,21 @@ SSL_CTX_callback_ctrl(ssl,SSL_CTRL_SET_T
 #define TLS1_CK_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA  0x03000088
 #define TLS1_CK_ADH_WITH_CAMELLIA_256_CBC_SHA      0x03000089

+/* Indect ciphersuites from XXX (private) */
+#define TLS1_CK_RSA_WITH_INDECT_128_CBC_SHA        0x0300FF41
+#define TLS1_CK_DH_DSS_WITH_INDECT_128_CBC_SHA     0x0300FF42
+#define TLS1_CK_DH_RSA_WITH_INDECT_128_CBC_SHA     0x0300FF43
+#define TLS1_CK_DHE_DSS_WITH_INDECT_128_CBC_SHA          0x0300FF44
+#define TLS1_CK_DHE_RSA_WITH_INDECT_128_CBC_SHA          0x0300FF45
+#define TLS1_CK_ADH_WITH_INDECT_128_CBC_SHA        0x0300FF46
+
+#define TLS1_CK_RSA_WITH_INDECT_320_CBC_SHA        0x0300FF84
+#define TLS1_CK_DH_DSS_WITH_INDECT_320_CBC_SHA     0x0300FF85
+#define TLS1_CK_DH_RSA_WITH_INDECT_320_CBC_SHA       0x0300FF86
+#define TLS1_CK_DHE_DSS_WITH_INDECT_320_CBC_SHA          0x0300FF87
+#define TLS1_CK_DHE_RSA_WITH_INDECT_320_CBC_SHA    0x0300FF88
+#define TLS1_CK_ADH_WITH_INDECT_320_CBC_SHA        0x0300FF89
+
 /* SEED ciphersuites from RFC4162 */
```

```
 #define TLS1_CK_RSA_WITH_SEED_SHA                       0x03000096
 #define TLS1_CK_DH_DSS_WITH_SEED_SHA                    0x03000097
@@ -345,6 +360,21 @@ SSL_CTX_callback_ctrl(ssl,SSL_CTRL_SET_T
 #define TLS1_TXT_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA "DHE-RSA-CAMELLIA256-SHA"
 #define TLS1_TXT_ADH_WITH_CAMELLIA_256_CBC_SHA     "ADH-CAMELLIA256-SHA"

+/* Indect ciphersuites from XXX (private) */
+#define TLS1_TXT_RSA_WITH_INDECT_128_CBC_SHA       "INDECT128-SHA"
+#define TLS1_TXT_DH_DSS_WITH_INDECT_128_CBC_SHA    "DH-DSS-INDECT128-SHA"
+#define TLS1_TXT_DH_RSA_WITH_INDECT_128_CBC_SHA          "DH-RSA-INDECT128-SHA"
+#define TLS1_TXT_DHE_DSS_WITH_INDECT_128_CBC_SHA   "DHE-DSS-INDECT128-SHA"
+#define TLS1_TXT_DHE_RSA_WITH_INDECT_128_CBC_SHA   "DHE-RSA-INDECT128-SHA"
+#define TLS1_TXT_ADH_WITH_INDECT_128_CBC_SHA       "ADH-INDECT128-SHA"
+
+#define TLS1_TXT_RSA_WITH_INDECT_320_CBC_SHA       "INDECT320-SHA"
+#define TLS1_TXT_DH_DSS_WITH_INDECT_320_CBC_SHA    "DH-DSS-INDECT320-SHA"
+#define TLS1_TXT_DH_RSA_WITH_INDECT_320_CBC_SHA          "DH-RSA-INDECT320-SHA"
+#define TLS1_TXT_DHE_DSS_WITH_INDECT_320_CBC_SHA   "DHE-DSS-INDECT320-SHA"
+#define TLS1_TXT_DHE_RSA_WITH_INDECT_320_CBC_SHA   "DHE-RSA-INDECT320-SHA"
+#define TLS1_TXT_ADH_WITH_INDECT_320_CBC_SHA       "ADH-INDECT320-SHA"
+
 /* SEED ciphersuites from RFC4162 */
 #define TLS1_TXT_RSA_WITH_SEED_SHA                      "SEED-SHA"
 #define TLS1_TXT_DH_DSS_WITH_SEED_SHA                   "DH-DSS-SEED-SHA"
diff -rupN openssl-0.9.8v-org//Makefile.org openssl-0.9.8v//Makefile.org
--- openssl-0.9.8v-org//Makefile.org    2010-01-27 17:06:36.000000000 +0100
+++ openssl-0.9.8v//Makefile.org    2012-04-20 23:44:00.000000000 +0200
@@ -139,7 +139,7 @@ SHLIBDIRS= crypto ssl fips
 SDIRS=  \
    objects \
    md2 md4 md5 sha mdc2 hmac ripemd \
-   des aes rc2 rc4 rc5 idea bf cast camellia seed \
+   des aes rc2 rc4 rc5 idea bf cast camellia indect seed \
    bn ec rsa dsa ecdsa dh ecdh dso engine \
    buffer bio stack lhash rand err \
    evp asn1 pem x509 x509v3 conf txt_db pkcs7 pkcs12 comp ocsp ui krb5 \
diff -rupN openssl-0.9.8v-org//makevms.com openssl-0.9.8v//makevms.com
--- openssl-0.9.8v-org//makevms.com 2010-03-25 17:25:42.000000000 +0100
+++ openssl-0.9.8v//makevms.com 2012-04-20 23:44:54.000000000 +0200
@@ -198,7 +198,7 @@ $ WRITE H_FILE "# define OPENSSL_SYS_VMS
 $ WRITE H_FILE "#endif"
 $ CONFIG_LOGICALS := NO_ASM,NO_RSA,NO_DSA,NO_DH,NO_MD2,NO_MD5,NO_RIPEMD,-
    NO_SHA,NO_SHA0,NO_SHA1,NO_DES/NO_MDC2;NO_MDC2,NO_RC2,NO_RC4,NO_RC5,-
-   NO_IDEA,NO_BF,NO_CAST,NO_CAMELLIA,NO_SEED,NO_HMAC,NO_SSL2
+   NO_IDEA,NO_BF,NO_CAST,NO_CAMELLIA,NO_INDECT,NO_SEED,NO_HMAC,NO_SSL2
 $ CONFIG_LOG_I = 0
 $ CONFIG_LOG_LOOP:
 $   CONFIG_LOG_E1 = F$ELEMENT(CONFIG_LOG_I,",",CONFIG_LOGICALS)
@@ -436,7 +436,7 @@ $ SDIRS := ,-
    _'ARCH',-
    OBJECTS,-
    MD2,MD4,MD5,SHA,MDC2,HMAC,RIPEMD,-
-   DES,AES,RC2,RC4,RC5,IDEA,BF,CAST,CAMELLIA,SEED,-
+   DES,AES,RC2,RC4,RC5,IDEA,BF,CAST,CAMELLIA,INDECT,SEED,-
    BN,EC,RSA,DSA,ECDSA,DH,ECDH,DSO,ENGINE,-
    BUFFER,BIO,STACK,LHASH,RAND,ERR,-
    EVP,ASN1,PEM,X509,X509V3,CONF,TXT_DB,PKCS7,PKCS12,COMP,OCSP,UI,KRB5,-
@@ -461,6 +461,7 @@ $ EXHEADER_IDEA := idea.h
 $ EXHEADER_BF := blowfish.h
 $ EXHEADER_CAST := cast.h
 $ EXHEADER_CAMELLIA := camellia.h
+$ EXHEADER_INDECT := indect.h
 $ EXHEADER_SEED := seed.h
 $ EXHEADER_BN := bn.h
 $ EXHEADER_EC := ec.h
diff -rupN openssl-0.9.8v-org//README openssl-0.9.8v//README
--- openssl-0.9.8v-org//README  2012-04-19 13:39:02.000000000 +0200
+++ openssl-0.9.8v//README  2012-04-29 23:32:21.000000000 +0200
@@ -1,5 +1,5 @@

- OpenSSL 0.9.8v 19 Apr 2012
+ OpenSSL 0.9.8v 19 Apr 2012 with Indect

   Copyright (c) 1998-2011 The OpenSSL Project
   Copyright (c) 1995-1998 Eric A. Young, Tim J. Hudson
diff -rupN openssl-0.9.8v-org//ssl/s3_lib.c openssl-0.9.8v//ssl/s3_lib.c
--- openssl-0.9.8v-org//ssl/s3_lib.c    2011-10-19 15:53:41.000000000 +0200
```

```
+++ openssl-0.9.8v//ssl/s3_lib.c    2012-05-09 23:51:34.000000000 +0200
@@ -1603,6 +1603,171 @@ OPENSSL_GLOBAL SSL_CIPHER ssl3_ciphers[]
             },
 #endif /* OPENSSL_NO_ECDH */

+#ifndef OPENSSL_NO_INDECT
+   /* Indect ciphersuites from XXX (private) (128-bit portion) */
+
+   /* Cipher FF41 */
+   {
+   1,
+   TLS1_TXT_RSA_WITH_INDECT_128_CBC_SHA,
+   TLS1_CK_RSA_WITH_INDECT_128_CBC_SHA,
+   SSL_kRSA|SSL_aRSA|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   128,
+   128,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF42 */
+   {
+   0, /* not implemented (non-ephemeral DH) */
+   TLS1_TXT_DH_DSS_WITH_INDECT_128_CBC_SHA,
+   TLS1_CK_DH_DSS_WITH_INDECT_128_CBC_SHA,
+   SSL_kDHd|SSL_aDH|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   128,
+   128,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF43 */
+   {
+   0, /* not implemented (non-ephemeral DH) */
+   TLS1_TXT_DH_RSA_WITH_INDECT_128_CBC_SHA,
+   TLS1_CK_DH_RSA_WITH_INDECT_128_CBC_SHA,
+   SSL_kDHr|SSL_aDH|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   128,
+   128,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF44 */
+   {
+   1,
+   TLS1_TXT_DHE_DSS_WITH_INDECT_128_CBC_SHA,
+   TLS1_CK_DHE_DSS_WITH_INDECT_128_CBC_SHA,
+   SSL_kEDH|SSL_aDSS|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   128,
+   128,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF45 */
+   {
+   1,
+   TLS1_TXT_DHE_RSA_WITH_INDECT_128_CBC_SHA,
+   TLS1_CK_DHE_RSA_WITH_INDECT_128_CBC_SHA,
+   SSL_kEDH|SSL_aRSA|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   128,
+   128,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF46 */
+   {
+   1,
+   TLS1_TXT_ADH_WITH_INDECT_128_CBC_SHA,
```

```
+   TLS1_CK_ADH_WITH_INDECT_128_CBC_SHA,
+   SSL_kEDH|SSL_aNULL|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   128,
+   128,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+#endif /* OPENSSL_NO_INDECT */
+
+#ifndef OPENSSL_NO_INDECT
+   /* Indect ciphersuites from X (private) (256-bit portion) */
+
+   /* Cipher FF84 */
+   {
+   1,
+   TLS1_TXT_RSA_WITH_INDECT_320_CBC_SHA,
+   TLS1_CK_RSA_WITH_INDECT_320_CBC_SHA,
+   SSL_kRSA|SSL_aRSA|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   320,
+   320,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF85 */
+   {
+   0, /* not implemented (non-ephemeral DH) */
+   TLS1_TXT_DH_DSS_WITH_INDECT_320_CBC_SHA,
+   TLS1_CK_DH_DSS_WITH_INDECT_320_CBC_SHA,
+   SSL_kDHd|SSL_aDH|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   320,
+   320,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF86 */
+   {
+   0, /* not implemented (non-ephemeral DH) */
+   TLS1_TXT_DH_RSA_WITH_INDECT_320_CBC_SHA,
+   TLS1_CK_DH_RSA_WITH_INDECT_320_CBC_SHA,
+   SSL_kDHr|SSL_aDH|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   320,
+   320,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF87 */
+   {
+   1,
+   TLS1_TXT_DHE_DSS_WITH_INDECT_320_CBC_SHA,
+   TLS1_CK_DHE_DSS_WITH_INDECT_320_CBC_SHA,
+   SSL_kEDH|SSL_aDSS|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   320,
+   320,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF88 */
+   {
+   1,
+   TLS1_TXT_DHE_RSA_WITH_INDECT_320_CBC_SHA,
+   TLS1_CK_DHE_RSA_WITH_INDECT_320_CBC_SHA,
+   SSL_kEDH|SSL_aRSA|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   320,
+   320,
+   SSL_ALL_CIPHERS,
```

```
+   SSL_ALL_STRENGTHS
+   },
+   /* Cipher FF89 */
+   {
+   1,
+   TLS1_TXT_ADH_WITH_INDECT_320_CBC_SHA,
+   TLS1_CK_ADH_WITH_INDECT_320_CBC_SHA,
+   SSL_kEDH|SSL_aNULL|SSL_INDECT|SSL_SHA|SSL_TLSV1,
+   SSL_NOT_EXP|SSL_HIGH,
+   0,
+   320,
+   320,
+   SSL_ALL_CIPHERS,
+   SSL_ALL_STRENGTHS
+   },
+#endif /* OPENSSL_NO_INDECT */

 /* end of list */
    };
diff -rupN openssl-0.9.8v-org//ssl/ssl_algs.c openssl-0.9.8v//ssl/ssl_algs.c
--- openssl-0.9.8v-org//ssl/ssl_algs.c  2010-04-07 15:19:48.000000000 +0200
+++ openssl-0.9.8v//ssl/ssl_algs.c  2012-05-09 22:57:42.000000000 +0200
@@ -88,6 +88,11 @@ int SSL_library_init(void)
    EVP_add_cipher(EVP_camellia_256_cbc());
 #endif

+#ifndef OPENSSL_NO_INDECT
+   EVP_add_cipher(EVP_indect_128_cbc());
+   EVP_add_cipher(EVP_indect_320_cbc());
+#endif
+
 #ifndef OPENSSL_NO_SEED
    EVP_add_cipher(EVP_seed_cbc());
 #endif
diff -rupN openssl-0.9.8v-org//ssl/ssl_ciph.c openssl-0.9.8v//ssl/ssl_ciph.c
--- openssl-0.9.8v-org//ssl/ssl_ciph.c  2011-12-02 13:50:44.000000000 +0100
+++ openssl-0.9.8v//ssl/ssl_ciph.c  2012-05-09 23:45:34.000000000 +0200
@@ -133,8 +133,9 @@
 #define SSL_ENC_CAMELLIA128_IDX    9
 #define SSL_ENC_CAMELLIA256_IDX    10
 #define SSL_ENC_SEED_IDX       11
-#define SSL_ENC_NUM_IDX        12
-
+#define SSL_ENC_INDECT128_IDX  12
+#define SSL_ENC_INDECT320_IDX  13
+#define SSL_ENC_NUM_IDX        14

 static const EVP_CIPHER *ssl_cipher_methods[SSL_ENC_NUM_IDX]={
    NULL,NULL,NULL,NULL,NULL,NULL,
@@ -203,6 +204,7 @@ static const SSL_CIPHER cipher_aliases[]
    {0,SSL_TXT_eFZA,0,SSL_eFZA,  0,0,0,0,SSL_ENC_MASK,0},
    {0,SSL_TXT_AES, 0,SSL_AES,   0,0,0,0,SSL_ENC_MASK,0},
    {0,SSL_TXT_CAMELLIA,0,SSL_CAMELLIA, 0,0,0,0,SSL_ENC_MASK,0},
+   {0,SSL_TXT_INDECT,0,SSL_INDECT, 0,0,0,0,SSL_ENC_MASK,0},

    {0,SSL_TXT_MD5, 0,SSL_MD5,   0,0,0,0,SSL_MAC_MASK,0},
    {0,SSL_TXT_SHA1,0,SSL_SHA1,  0,0,0,0,SSL_MAC_MASK,0},
@@ -252,6 +254,10 @@ void ssl_load_ciphers(void)
      EVP_get_cipherbyname(SN_camellia_128_cbc);
    ssl_cipher_methods[SSL_ENC_CAMELLIA256_IDX]=
      EVP_get_cipherbyname(SN_camellia_256_cbc);
+   ssl_cipher_methods[SSL_ENC_INDECT128_IDX]=
+     EVP_get_cipherbyname(SN_indect_128_cbc);
+   ssl_cipher_methods[SSL_ENC_INDECT320_IDX]=
+     EVP_get_cipherbyname(SN_indect_320_cbc);
    ssl_cipher_methods[SSL_ENC_SEED_IDX]=
      EVP_get_cipherbyname(SN_seed_cbc);

@@ -381,6 +387,14 @@ int ssl_cipher_get_evp(const SSL_SESSION
        default: i=-1; break;
            }
        break;
+   case SSL_INDECT:
+       switch(c->alg_bits)
+           {
+       case 128: i=SSL_ENC_INDECT128_IDX; break;
+       case 320: i=SSL_ENC_INDECT320_IDX; break;
```

```
+       default: i=-1; break;
+           }
+       break;
    case SSL_SEED:
        i=SSL_ENC_SEED_IDX;
        break;
@@ -490,8 +504,10 @@ static struct disabled_masks ssl_cipher_
    m256 = mask;
    mask |= (ssl_cipher_methods[SSL_ENC_AES128_IDX] == NULL) ? SSL_AES:0;
    mask |= (ssl_cipher_methods[SSL_ENC_CAMELLIA128_IDX] == NULL) ? SSL_CAMELLIA:0;
+   mask |= (ssl_cipher_methods[SSL_ENC_INDECT128_IDX] == NULL) ? SSL_INDECT:0;
    m256 |= (ssl_cipher_methods[SSL_ENC_AES256_IDX] == NULL) ? SSL_AES:0;
    m256 |= (ssl_cipher_methods[SSL_ENC_CAMELLIA256_IDX] == NULL) ? SSL_CAMELLIA:0;
+   m256 |= (ssl_cipher_methods[SSL_ENC_INDECT320_IDX] == NULL) ? SSL_INDECT:0;

    ret.mask = mask;
    ret.m256 = m256;
@@ -1219,6 +1235,14 @@ char *SSL_CIPHER_description(const SSL_C
        default: enc="Camellia(?""?""?)"; break;
            }
        break;
+   case SSL_INDECT:
+       switch(cipher->strength_bits)
+           {
+       case 128: enc="Indect(128)"; break;
+       case 320: enc="Indect(320)"; break;
+       default: enc="Indect(?""?""?)"; break;
+           }
+       break;
    case SSL_SEED:
        enc="SEED(128)";
        break;
diff -rupN openssl-0.9.8v-org//ssl/ssl.h openssl-0.9.8v//ssl/ssl.h
--- openssl-0.9.8v-org//ssl/ssl.h   2012-03-12 15:50:55.000000000 +0100
+++ openssl-0.9.8v//ssl/ssl.h   2012-08-12 18:37:32.000000000 +0200
@@ -286,6 +286,7 @@ extern "C" {
 #define SSL_TXT_SEED       "SEED"
 #define SSL_TXT_AES        "AES"
 #define SSL_TXT_CAMELLIA   "CAMELLIA"
+#define SSL_TXT_INDECT     "INDECT"
 #define SSL_TXT_MD5        "MD5"
 #define SSL_TXT_SHA1       "SHA1"
 #define SSL_TXT_SHA        "SHA"
diff -rupN openssl-0.9.8v-org//ssl/ssl_locl.h openssl-0.9.8v//ssl/ssl_locl.h
--- openssl-0.9.8v-org//ssl/ssl_locl.h  2012-03-09 16:51:56.000000000 +0100
+++ openssl-0.9.8v//ssl/ssl_locl.h  2012-04-20 23:59:10.000000000 +0200
@@ -280,7 +280,7 @@
 #define SSL_FZA            (SSL_aFZA|SSL_kFZA|SSL_eFZA)
 #define SSL_KRB5               (SSL_kKRB5|SSL_aKRB5)

-#define SSL_ENC_MASK       0x1C3F8000L
+#define SSL_ENC_MASK       0x3C3F8000L
 #define SSL_DES            0x00008000L
 #define SSL_3DES       0x00010000L
 #define SSL_RC4            0x00020000L
@@ -291,6 +291,7 @@
 #define SSL_AES            0x04000000L
 #define SSL_CAMELLIA       0x08000000L
 #define SSL_SEED           0x10000000L
+#define SSL_INDECT             0x20000000L

 #define SSL_MAC_MASK       0x00c00000L
 #define SSL_MD5            0x00400000L
@@ -302,7 +303,7 @@
 #define SSL_SSLV3      0x02000000L
 #define SSL_TLSV1      SSL_SSLV3   /* for now */

-/* we have used 1fffffff - 3 bits left to go. */
+/* we have used 3fffffff - 2 bits left to go. */

 /*
  * Export and cipher strength information. For each cipher we have to decide
diff -rupN openssl-0.9.8v-org//ssl/tls1.h openssl-0.9.8v//ssl/tls1.h
--- openssl-0.9.8v-org//ssl/tls1.h  2009-11-08 15:51:54.000000000 +0100
+++ openssl-0.9.8v//ssl/tls1.h  2012-05-09 22:58:53.000000000 +0200
@@ -230,6 +230,21 @@ SSL_CTX_callback_ctrl(ssl,SSL_CTRL_SET_T
 #define TLS1_CK_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA  0x03000088
```

```
 #define TLS1_CK_ADH_WITH_CAMELLIA_256_CBC_SHA      0x03000089

+/* Indect ciphersuites from XXX (private) */
+#define TLS1_CK_RSA_WITH_INDECT_128_CBC_SHA        0x0300FF41
+#define TLS1_CK_DH_DSS_WITH_INDECT_128_CBC_SHA     0x0300FF42
+#define TLS1_CK_DH_RSA_WITH_INDECT_128_CBC_SHA     0x0300FF43
+#define TLS1_CK_DHE_DSS_WITH_INDECT_128_CBC_SHA          0x0300FF44
+#define TLS1_CK_DHE_RSA_WITH_INDECT_128_CBC_SHA          0x0300FF45
+#define TLS1_CK_ADH_WITH_INDECT_128_CBC_SHA        0x0300FF46
+
+#define TLS1_CK_RSA_WITH_INDECT_320_CBC_SHA        0x0300FF84
+#define TLS1_CK_DH_DSS_WITH_INDECT_320_CBC_SHA     0x0300FF85
+#define TLS1_CK_DH_RSA_WITH_INDECT_320_CBC_SHA       0x0300FF86
+#define TLS1_CK_DHE_DSS_WITH_INDECT_320_CBC_SHA          0x0300FF87
+#define TLS1_CK_DHE_RSA_WITH_INDECT_320_CBC_SHA    0x0300FF88
+#define TLS1_CK_ADH_WITH_INDECT_320_CBC_SHA        0x0300FF89
+
 /* SEED ciphersuites from RFC4162 */
 #define TLS1_CK_RSA_WITH_SEED_SHA                       0x03000096
 #define TLS1_CK_DH_DSS_WITH_SEED_SHA                    0x03000097
@@ -345,6 +360,21 @@ SSL_CTX_callback_ctrl(ssl,SSL_CTRL_SET_T
 #define TLS1_TXT_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA "DHE-RSA-CAMELLIA256-SHA"
 #define TLS1_TXT_ADH_WITH_CAMELLIA_256_CBC_SHA     "ADH-CAMELLIA256-SHA"

+/* Indect ciphersuites from XXX (private) */
+#define TLS1_TXT_RSA_WITH_INDECT_128_CBC_SHA       "INDECT128-SHA"
+#define TLS1_TXT_DH_DSS_WITH_INDECT_128_CBC_SHA    "DH-DSS-INDECT128-SHA"
+#define TLS1_TXT_DH_RSA_WITH_INDECT_128_CBC_SHA          "DH-RSA-INDECT128-SHA"
+#define TLS1_TXT_DHE_DSS_WITH_INDECT_128_CBC_SHA   "DHE-DSS-INDECT128-SHA"
+#define TLS1_TXT_DHE_RSA_WITH_INDECT_128_CBC_SHA   "DHE-RSA-INDECT128-SHA"
+#define TLS1_TXT_ADH_WITH_INDECT_128_CBC_SHA       "ADH-INDECT128-SHA"
+
+#define TLS1_TXT_RSA_WITH_INDECT_320_CBC_SHA       "INDECT320-SHA"
+#define TLS1_TXT_DH_DSS_WITH_INDECT_320_CBC_SHA    "DH-DSS-INDECT320-SHA"
+#define TLS1_TXT_DH_RSA_WITH_INDECT_320_CBC_SHA          "DH-RSA-INDECT320-SHA"
+#define TLS1_TXT_DHE_DSS_WITH_INDECT_320_CBC_SHA   "DHE-DSS-INDECT320-SHA"
+#define TLS1_TXT_DHE_RSA_WITH_INDECT_320_CBC_SHA   "DHE-RSA-INDECT320-SHA"
+#define TLS1_TXT_ADH_WITH_INDECT_320_CBC_SHA       "ADH-INDECT320-SHA"
+
 /* SEED ciphersuites from RFC4162 */
 #define TLS1_TXT_RSA_WITH_SEED_SHA                      "SEED-SHA"
 #define TLS1_TXT_DH_DSS_WITH_SEED_SHA                   "DH-DSS-SEED-SHA"
diff -rupN openssl-0.9.8v-org//test/evp_test.c openssl-0.9.8v//test/evp_test.c
--- openssl-0.9.8v-org//test/evp_test.c 2011-09-01 15:48:48.000000000 +0200
+++ openssl-0.9.8v//test/evp_test.c 2012-04-20 21:06:30.000000000 +0200
@@ -424,6 +424,13 @@ int main(int argc,char **argv)
        continue;
        }
 #endif
+#ifdef OPENSSL_NO_INDECT
+       if (strstr(cipher, "INDECT") == cipher)
+       {
+       fprintf(stdout, "Cipher disabled, skipping %s\n", cipher);
+       continue;
+       }
+#endif
 #ifdef OPENSSL_NO_SEED
        if (strstr(cipher, "SEED") == cipher)
        {
diff -rupN openssl-0.9.8v-org//util/libeay.num openssl-0.9.8v//util/libeay.num
--- openssl-0.9.8v-org//util/libeay.num 2010-03-25 13:17:16.000000000 +0100
+++ openssl-0.9.8v//util/libeay.num 2012-08-12 23:16:15.000000000 +0200
@@ -3728,3 +3728,33 @@ JPAKE_STEP2_init
 pqueue_size                            4114   EXIST::FUNCTION:
 OPENSSL_uni2asc                        4115   EXIST:NETWARE:FUNCTION:
 OPENSSL_asc2uni                        4116   EXIST:NETWARE:FUNCTION:
+Indect_ecb_encrypt                     4117   EXIST::FUNCTION:INDECT
+EVP_indect_192_cfb1                    4118   EXIST::FUNCTION:INDECT
+EVP_indect_128_cfb1                    4119   EXIST::FUNCTION:INDECT
+EVP_indect_192_cbc                     4120   EXIST::FUNCTION:INDECT
+Indect_cfb8_encrypt                    4121   EXIST::FUNCTION:INDECT
+EVP_indect_320_cfb1                    4122   EXIST::FUNCTION:INDECT
+Indect_decrypt                         4123   EXIST::FUNCTION:INDECT
+EVP_indect_192_cfb128                  4124   EXIST::FUNCTION:INDECT
+Indect_encrypt                         4125   EXIST::FUNCTION:INDECT
+EVP_indect_320_cbc                     4126   EXIST::FUNCTION:INDECT
+Indect_set_decrypt_key                 4127   EXIST::FUNCTION:INDECT
```

```
+EVP_indect_192_ecb                     4128   EXIST::FUNCTION:INDECT
+EVP_indect_128_cfb8                    4129   EXIST::FUNCTION:INDECT
+EVP_indect_320_ecb                     4130   EXIST::FUNCTION:INDECT
+EVP_indect_128_cfb128                  4131   EXIST::FUNCTION:INDECT
+EVP_indect_320_ofb                     4132   EXIST::FUNCTION:INDECT
+Indect_cfbr_encrypt_block              4133   EXIST::FUNCTION:INDECT
+Indect_cfb128_encrypt                  4134   EXIST::FUNCTION:INDECT
+EVP_indect_192_ofb                     4135   EXIST::FUNCTION:INDECT
+EVP_indect_128_cbc                     4136   EXIST::FUNCTION:INDECT
+Indect_ofb128_encrypt                  4137   EXIST::FUNCTION:INDECT
+EVP_indect_128_ecb                     4138   EXIST::FUNCTION:INDECT
+EVP_indect_192_cfb8                    4139   EXIST::FUNCTION:INDECT
+Indect_ctr128_encrypt                  4140   EXIST::FUNCTION:INDECT
+EVP_indect_320_cfb8                    4141   EXIST::FUNCTION:INDECT
+Indect_set_encrypt_key                 4142   EXIST::FUNCTION:INDECT
+Indect_cbc_encrypt                     4143   EXIST::FUNCTION:INDECT
+EVP_indect_320_cfb128                  4144   EXIST::FUNCTION:INDECT
+EVP_indect_128_ofb                     4145   EXIST::FUNCTION:INDECT
+Indect_cfb1_encrypt                    4146   EXIST::FUNCTION:INDECT
diff -rupN openssl-0.9.8v-org//util/mk1mf.pl openssl-0.9.8v//util/mk1mf.pl
--- openssl-0.9.8v-org//util/mk1mf.pl   2009-09-20 14:46:42.000000000 +0200
+++ openssl-0.9.8v//util/mk1mf.pl   2012-04-21 00:46:06.000000000 +0200
@@ -76,7 +76,7 @@ and [options] can be one of
     no-md2 no-md4 no-md5 no-sha no-mdc2 - Skip this digest
     no-ripemd
     no-rc2 no-rc4 no-rc5 no-idea no-des     - Skip this symetric cipher
-    no-bf no-cast no-aes no-camellia no-seed
+    no-bf no-cast no-aes no-camellia no-indect no-seed
     no-rsa no-dsa no-dh         - Skip this public key cipher
     no-ssl2 no-ssl3             - Skip this version of SSL
     just-ssl                - remove all non-ssl keys/digest
@@ -211,6 +211,7 @@ $cflags= "$xcflags$cflags" if $xcflags n
 $cflags.=" -DOPENSSL_NO_IDEA" if $no_idea;
 $cflags.=" -DOPENSSL_NO_AES"  if $no_aes;
 $cflags.=" -DOPENSSL_NO_CAMELLIA"  if $no_camellia;
+$cflags.=" -DOPENSSL_NO_INDECT"  if $no_indect;
 $cflags.=" -DOPENSSL_NO_SEED" if $no_seed;
 $cflags.=" -DOPENSSL_NO_RC2"  if $no_rc2;
 $cflags.=" -DOPENSSL_NO_RC4"  if $no_rc4;
@@ -1012,6 +1013,7 @@ sub var_add
     return("") if $no_idea && $dir =~ /\/idea/;
     return("") if $no_aes  && $dir =~ /\/aes/;
     return("") if $no_camellia  && $dir =~ /\/camellia/;
+       return("") if $no_indect  && $dir =~ /\/indect/;
     return("") if $no_seed && $dir =~ /\/seed/;
     return("") if $no_rc2  && $dir =~ /\/rc2/;
     return("") if $no_rc4  && $dir =~ /\/rc4/;
@@ -1050,6 +1052,7 @@ sub var_add
     @a=grep(!/^e_.*_c$/,@a) if $no_cast;
     @a=grep(!/^e_rc4$/,@a) if $no_rc4;
     @a=grep(!/^e_camellia$/,@a) if $no_camellia;
+       @a=grep(!/^e_indect$/,@a) if $no_indect;
     @a=grep(!/^e_seed$/,@a) if $no_seed;

     @a=grep(!/(^s2_)|(^s23_)/,@a) if $no_ssl2;
@@ -1264,6 +1267,7 @@ sub read_options
         "no-idea" => \$no_idea,
         "no-aes" => \$no_aes,
         "no-camellia" => \$no_camellia,
+       "no-indect" => \$no_indect,
         "no-seed" => \$no_seed,
         "no-des" => \$no_des,
         "no-bf" => \$no_bf,
@@ -1305,7 +1309,7 @@ sub read_options
             [\$no_rc2, \$no_idea, \$no_des, \$no_bf, \$no_cast,
               \$no_md2, \$no_sha, \$no_mdc2, \$no_dsa, \$no_dh,
               \$no_ssl2, \$no_err, \$no_ripemd, \$no_rc5,
-              \$no_aes, \$no_camellia, \$no_seed],
+              \$no_aes, \$no_camellia, \$no_indect, \$no_seed],
         "rsaref" => 0,
         "gcc" => \$gcc,
         "debug" => \$debug,
diff -rupN openssl-0.9.8v-org//util/mkdef.pl openssl-0.9.8v//util/mkdef.pl
--- openssl-0.9.8v-org//util/mkdef.pl   2010-03-25 13:17:17.000000000 +0100
+++ openssl-0.9.8v//util/mkdef.pl   2012-04-21 00:49:04.000000000 +0200
@@ -84,7 +84,7 @@ my @known_ossl_platforms = ( "VMS", "WIN
 my @known_algorithms = ( "RC2", "RC4", "RC5", "IDEA", "DES", "BF",
```

```
             "CAST", "MD2", "MD4", "MD5", "SHA", "SHA0", "SHA1",
             "SHA256", "SHA512", "RIPEMD",
-            "MDC2", "RSA", "DSA", "DH", "EC", "ECDH", "ECDSA", "HMAC", "AES", "CAMELLIA",
"SEED",
+            "MDC2", "RSA", "DSA", "DH", "EC", "ECDH", "ECDSA", "HMAC", "AES", "CAMELLIA",
"INDECT", "SEED",
             # Envelope "algorithms"
             "EVP", "X509", "ASN1_TYPEDEFS",
             # Helper "algorithms"
@@ -121,7 +121,7 @@ my $no_rc2; my $no_rc4; my $no_rc5; my $
 my $no_cast;
 my $no_md2; my $no_md4; my $no_md5; my $no_sha; my $no_ripemd; my $no_mdc2;
 my $no_rsa; my $no_dsa; my $no_dh; my $no_hmac=0; my $no_aes; my $no_krb5;
-my $no_ec; my $no_ecdsa; my $no_ecdh; my $no_engine; my $no_hw; my $no_camellia;
+my $no_ec; my $no_ecdsa; my $no_ecdh; my $no_engine; my $no_hw; my $no_camellia; my
$no_indect;
 my $no_seed;
 my $no_fp_api; my $no_static_engine; my $no_gmp; my $no_deprecated;
 my $no_rfc3779; my $no_tlsext; my $no_cms; my $no_capieng; my $no_jpake;
@@ -195,6 +195,7 @@ foreach (@ARGV, split(/ /, $options))
     elsif (/^no-hmac$/) { $no_hmac=1; }
     elsif (/^no-aes$/)  { $no_aes=1; }
     elsif (/^no-camellia$/) { $no_camellia=1; }
+    elsif (/^no-indect$/)   { $no_indect=1; }
     elsif (/^no-seed$/)     { $no_seed=1; }
     elsif (/^no-evp$/)  { $no_evp=1; }
     elsif (/^no-lhash$/)    { $no_lhash=1; }
@@ -268,6 +269,7 @@ $crypto.=" crypto/sha/sha.h" ; # unless
 $crypto.=" crypto/ripemd/ripemd.h" ; # unless $no_ripemd;
 $crypto.=" crypto/aes/aes.h" ; # unless $no_aes;
 $crypto.=" crypto/camellia/camellia.h" ; # unless $no_camellia;
+$crypto.=" crypto/indect/indect.h" ; # unless $no_indect;
 $crypto.=" crypto/seed/seed.h"; # unless $no_seed;

 $crypto.=" crypto/bn/bn.h";
@@ -1143,6 +1145,7 @@ sub is_valid
             if ($keyword eq "HMAC" && $no_hmac) { return 0; }
             if ($keyword eq "AES" && $no_aes) { return 0; }
             if ($keyword eq "CAMELLIA" && $no_camellia) { return 0; }
+            if ($keyword eq "INDECT" && $no_indect) { return 0; }
             if ($keyword eq "SEED" && $no_seed) { return 0; }
             if ($keyword eq "EVP" && $no_evp) { return 0; }
             if ($keyword eq "LHASH" && $no_lhash) { return 0; }
diff -rupN openssl-0.9.8v-org//util/mkfiles.pl openssl-0.9.8v//util/mkfiles.pl
--- openssl-0.9.8v-org//util/mkfiles.pl 2008-10-27 13:30:33.000000000 +0100
+++ openssl-0.9.8v//util/mkfiles.pl 2012-04-21 00:49:56.000000000 +0200
@@ -25,6 +25,7 @@ my @dirs = (
 "crypto/cast",
 "crypto/aes",
 "crypto/camellia",
+"crypto/indect",
 "crypto/seed",
 "crypto/bn",
 "crypto/rsa",
```