Politechnika Warszawska

Cryptography and Information Security

IDEA cipher

Software implementation of International Data Encryption Algorithm (IDEA) cipher with 4 ciphering modes.

Student:

David Miguel Lozano

Guided by:

Dr. Tomasz Adamski

Summer 2016

Contents

List of Figures

A Introduction

The purpose of this report is to introduce the International Data Encryption Algorithm (IDEA) and describe the implementation done.

"International Data Encryption Algorithm (IDEA), originally called Improved Proposed Encryption Standard (IPES), is a symmetric-key block cipher designed by James Massey of ETH Zurich and Xuejia Lai and was first described in 1991. The algorithm was intended as a replacement for the Data Encryption Standard (DES). IDEA is a minor revision of an earlier cipher, Proposed Encryption Standard (PES)." [2]

B Symmetric-key algorithm

A symmetric key algorithm is a cryptography algorithm that use the same key for encryption and decryption. This key is a shared secret between the different parties that want to keep some secret information. [4]

Definition "Consider an encryption scheme consisting of the sets of encryption and decryption transformations $\{E_e : e \in K\}$ and $\{D_d : d \in K\}$, respectively, where K is the key space. The encryption scheme is said to be *symmetric-key* if for each associated encryption/decryption key pair (e, d), it is computationally "easy" to determine d knowing only e, and to determine e from d.

Since $e = d$ in most practical symmetric-key encryption schemes, the term symmetric-key becomes appropriate. Other terms used in the literature are *single-key*, *one-key*, *private-key*, and *conventional encryption*." [5]

There are two different types of symmetric key algorithms: [4]

- **Stream ciphers:** encrypt the digits (typically bytes) of a message one at a time.
- **Block ciphers:** take a number of bits and encrypt them as a single unit, padding the plaintext so that it is a multiple of the block size.

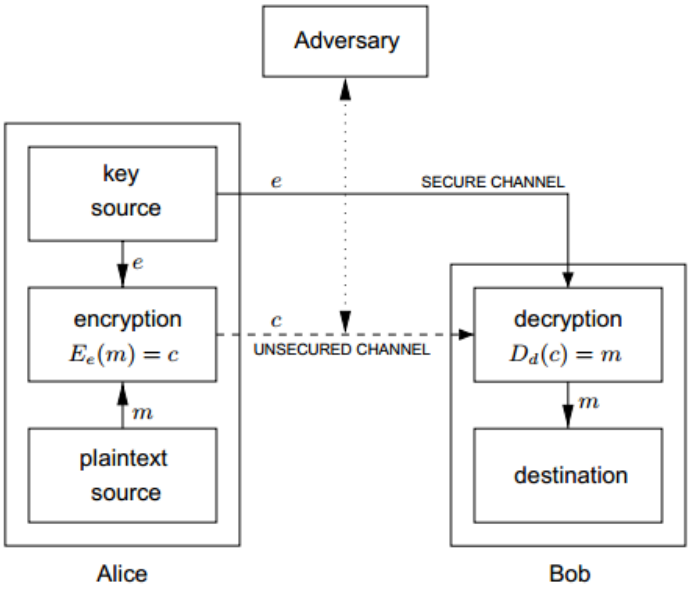

Figure 1: Two-party communication using symmetric encryption, with a secure channel for key exchange. [5]

C Block cipher

"A block cipher is an encryption scheme which breaks up the plaintext messages to be transmitted into strings (called blocks) of a fixed length t over an alphabet A, and encrypts one block at a time." [5]

Definition "A block cipher is specified by an encryption function which takes as input a key K of bit length k, called the key size, and a bit string P of length n, called the block size, and returns a string C of n bits. P is called the plaintext, and C is termed the ciphertext. For each K, the function $E_K(P)$ is required to be an invertible mapping on $\{0,1\}^n$.

$$E_K(P) := E(K,P) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$

The inverse for E is defined as a function

$$E_K^{-1}(C) := D_K(C) = D(K,C) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$

taking a key K and a ciphertext C to return a plaintext value P, such that

$$\forall K : D_K(E_K(P)) = P$$

For each key K, E_K is a permutation (a bijective mapping) over the set of input blocks. Each key selects one permutation from the possible set of $(2^n)!$." [1]

"A block cipher whose block size n is too small may be vulnerable to attacks based on statistical analysis. One such attack involves simple frequency analysis of ciphertext block. However, choosing too large a value for the blocksize n may create difficulties as the complexity of implementation of many ciphers grows rapidly with block size." [5]

C.1 Modes of operation

"A mode of operation is an algorithm that uses a block cipher to encrypt messages of arbitrary length in a way that provides confidentiality or authenticity. A block cipher by itself is only suitable for the secure cryptographic transformation (encryption or decryption) of one fixed-length group of bits called a block. A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely transform amounts of data larger than a block." [3]

The four most common modes are ECB, CBC, CFB, and OFB.

C.1.1 ECB mode

"The simplest of the encryption modes is the *Electronic Codebook* (ECB) mode. The message is divided into blocks, and each block is encrypted separately." [3]

The algorithm of the mode of operation ECB is the following:

Algorithm ECB mode of operation. [5]
INPUT: k-bit key K; n-bit plaintext blocks $x_1, ..., x_t$.
SUMMARY: produce ciphertext blocks $c_1, ..., c_t$; decrypt to recover plaintext.

1. Encryption: $for\ 1 \leq j \leq t, c_j \leftarrow E_K(x_j)$.
2. Decryption: $for\ 1 \leq j \leq t, x_j \leftarrow E_k^{-1}(c_j)$.

"The disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks; thus, it does not hide data patterns well." [3]

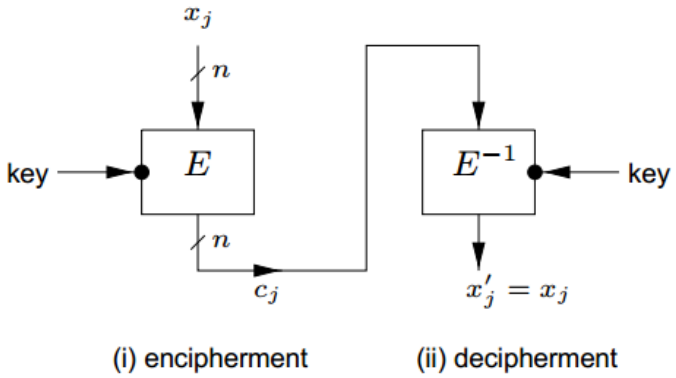

Figure 2: ECB mode of operation. [5]

C.1.2 CBC mode

"In *Cipher Block Chaining* (CBC) mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block depends on all plaintext blocks processed up to that point. To make each message unique, an initialization vector (IV) must be used in the first block." [3]

The algorithm of the mode of operation CBC is the following:

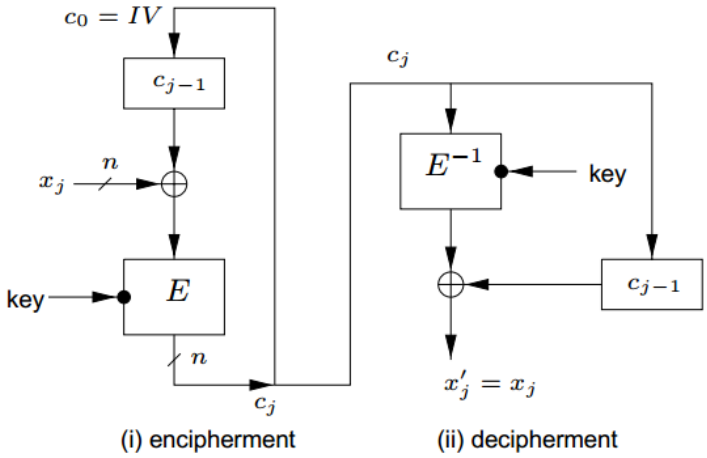

Figure 3: CBC mode of operation. [5]

Algorithm CBC mode of operation. [5]
INPUT: k-bit key K; n-bit IV ; n-bit plaintext blocks $x_1, ..., x_t$.
SUMMARY: produce ciphertext blocks $c_1, ..., c_t$; decrypt to recover plaintext.

1. Encryption: $c_0 \leftarrow IV$. $For\ 1 \leq j \leq t, c_j \leftarrow E_K(c_{j-1} \oplus x_j)$.
2. Decryption: $c_0 \leftarrow IV$. $For\ 1 \leq j \leq t, x_j \leftarrow c_{j-1} \oplus E_K^{-1}(c_j)$.

"Its main drawbacks are that encryption is sequential (i.e., it cannot be parallelized), and that the message must be padded to a multiple of the cipher block size." [3]

C.1.3 CFB mode

"The *Cipher Feedback* (CFB) mode, a close relative of CBC, makes a block cipher into a self-synchronizing stream cipher." [3]

"While the CBC mode processes plaintext n bits at a time (using an n-bit block cipher), some applications require that r-bit plaintext units be encrypted and transmitted without delay, for some fixed $r < n$ (often $r = 1$ or $r = 8$)." [5]

The algorithm of the mode of operation CFB is the following:

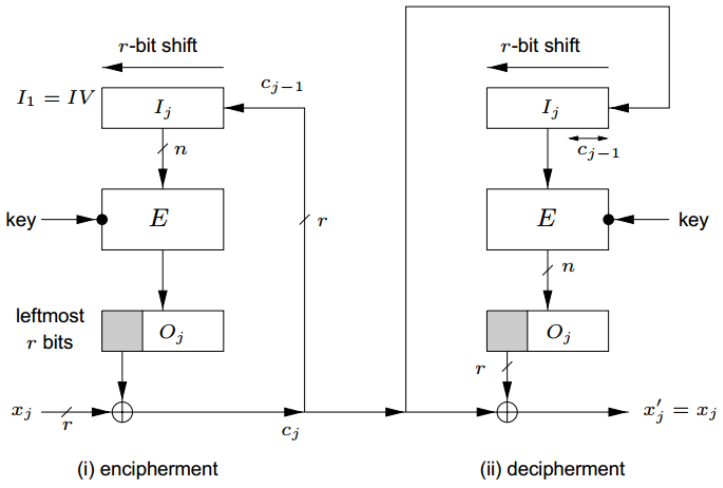

Figure 4: CFB mode of operation. [5]

Algorithm CFB-r mode of operation. [5]
INPUT: k-bit key K; n-bit IV ; r-bit plaintext blocks $x_1, ..., x_u$ $(1 \leq r \leq n)$.
SUMMARY: produce ciphertext blocks $c_1, ..., c_u$; decrypt to recover plaintext.

1. Encryption: $I_1 \leftarrow IV$. (I_j is the input value in a shift register). $For\ 1 \leq j \leq u$:
   (a) $O_j \leftarrow E_K(I_j)$. (Compute the block cipher output).
   (b) $t_j \leftarrow$ the r leftmost bits of O_j. (Assume the leftmost is identified as bit 1).
   (c) $c_j \leftarrow x_j \oplus t_j$. (Transmit the r-bit ciphertext block cj).
   (d) $I_{j+1} \leftarrow 2^r . I_j + c_j \ mod\ 2^n$. (Shift c_j into right end of shift register).
2. Decryption: $I_1 \leftarrow IV$. $For\ 1 \leq j \leq u$, upon receiving c_j:
   $x_j \leftarrow c_j \oplus t_j$, where t_j, O_j and I_j are computed as above.

"CFB shares two advantages over CBC mode with the stream cipher modes OFB and CTR: the block cipher is only ever used in the encrypting direction, and the message does not need to be padded to a multiple of the cipher block size (though ciphertext stealing can also be used to make padding unnecessary)." [3]

C.1.4 OFB mode

"The *Output Feedback* (OFB) mode makes a block cipher into a synchronous stream cipher. It generates keystream blocks, which are then XORed with the plaintext blocks to get the ciphertext. Just as with other stream ciphers, flipping a bit in the ciphertext produces a flipped bit in the plaintext at the same location. This property allows many error correcting codes to function normally even when applied before encryption." [3]

"Two versions of OFB using an n-bit block cipher are common. The ISO version requires an n-bit feedback, and is more secure. The earlier FIPS version allows $r < n$ bits of feedback." [5]

The algorithm of the mode of operation OFB is the following:

Algorithm OFB mode with full feedback (per ISO 10116). [5]
INPUT: k-bit key K; n-bit IV ; r-bit plaintext blocks $x_1, ..., x_u$ $(1 \leq r \leq n)$.
SUMMARY: produce ciphertext blocks $c_1, ..., c_u$; decrypt to recover plaintext.

1. Encryption: $I_1 \leftarrow IV$. $For\ 1 \leq j \leq u$, given plaintext block x_j:
   (a) $O_j \leftarrow E_K(I_j)$. (Compute the block cipher output).
   (b) $t_j \leftarrow$ the r leftmost bits of O_j. (Assume the leftmost is identified as bit 1).
   (c) $c_j \leftarrow x_j \oplus t_j$. (Transmit the r-bit ciphertext block cj).
   (d) $I_{j+1} \leftarrow O_j$. (Update the block cipher input for the next block).

2. Decryption: $I_1 \leftarrow IV$. *For* $1 \leq j \leq u$, upon receiving c_j:
   $x_j \leftarrow c_j \oplus t_j$, where t_j, O_j and I_j are computed as above.

Algorithm OFB mode with r-bit feedback (per FIPS 81). [5]
INPUT: k-bit key K; n-bit IV ; r-bit plaintext blocks $x_1, ..., x_u$ $(1 \leq r \leq n)$.
SUMMARY: produce ciphertext blocks $c_1, ..., c_u$; decrypt to recover plaintext.
As per Algorithm ISO 10116, but with "$I_{j+1} \leftarrow O_j$" replaced by:
$I_{j+1} \leftarrow 2^r \cdot I_j + t_j \ mod \ 2^n$. (Shift output t_j into right end of shift register).

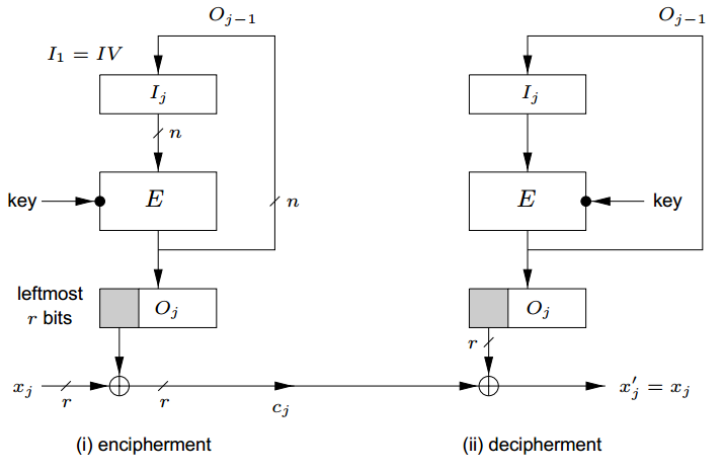

Figure 5: OFB mode of operation. [5]

D IDEA cipher

"The first incarnation of the IDEA cipher, by Xuejia Lai and James Massey, surfaced in 1990. It was called PES (Proposed Encryption Standard). The next year, after Biham and Shamir's demonstrated differential cryptanalysis, the authors strengthened their cipher against the attack and called the new algorithm IPES (Improved Proposed Encryption Standard). IPES changed its name to IDEA (International Data Encryption Algorithm) in 1992." [6]

"IDEA cipher encrypts 64-bit plaintext to 64-bit ciphertext blocks, using a 128-bit input key K. Based in part on a novel generalization of the Feistel structure, it consists of 8 computationally identical rounds followed by an output transformation. Round r uses six 16-bit subkeys $K_i^{(r)}$, $1 \leq i \leq 6$, to transform a 64-bit input X into an output of four 16-bit blocks, which are

input to the next round. The round 8 output enters the output transformation, employing four additional subkeys $K_i^{(9)}$, $1 \leq i \leq 4$ to produce the final ciphertext $Y = (Y_1, Y_2, Y_3, Y_4)$. "The same algorithm is used for both encryption and decryption." [6]

All subkeys are derived from K. A dominant design concept in IDEA is mixing operations from three different algebraic groups of 2^n elements. The corresponding group operations on sub-blocks a and b of bitlength $n = 16$ are:" [5]

- $a \oplus b$: bitwise XOR.
- $a \boxplus b$: addition $mod\ 2^n$: $(a + b)\ AND\ 0xFFFF$.
- $a \odot b$: (modified) multiplication $mod\ 2^n + 1$, with $0 \in Z_{2^n}$ associated with $2^n \in Z_{2^{n+1}}$.

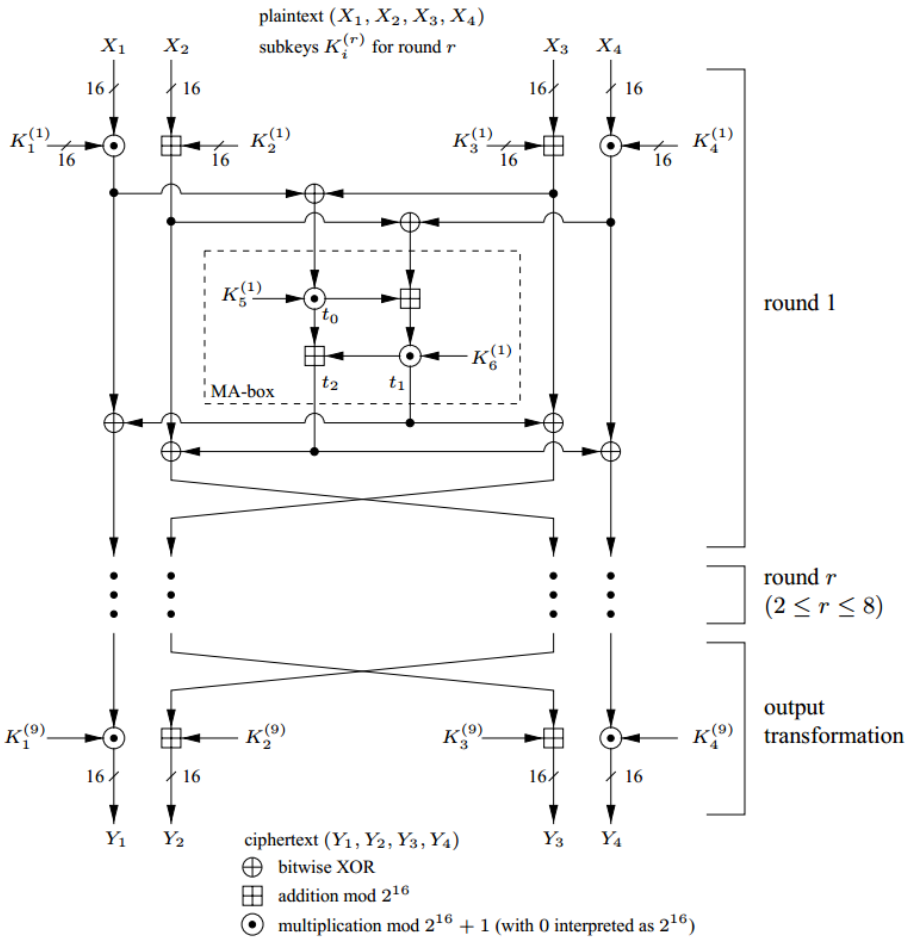

Figure 6: IDEA computation path. [5]

"The 64-bit data block is divided into four 16-bit sub-blocks: X_1, X_2, X_3 and X_4. These four sub-blocks become the input to the first round of the algorithm. There are eight rounds total. In each round the four sub-blocks are XORed, added, and multiplied with one another and with six 16-bit subkeys. Between rounds, the second and third sub-blocks are swapped.

Finally, the four sub-blocks are combined with four subkeys in an output transformation.

In each round, the sequence of events is as follows:

1. Multiply X_1 and the first subkey.
2. Add X_2 and the second subkey.
3. Add X_3 and the third subkey.
4. Multiply X_4 and the fourth subkey.
5. XOR the results of steps (1) and (3).
6. XOR the results of steps (2) and (4).
7. Multiply the results of step (5) with the fifth subkey.
8. Add the results of steps (6) and (7).
9. Multiply the results of step (8) with the sixth subkey.
10. Add the results of steps (7) and (9).
11. XOR the results of steps (1) and (9).
12. XOR the results of steps (3) and (9).
13. XOR the results of steps (2) and (10).
14. XOR the results of steps (4) and (10).

"The output of the round is the four sub-blocks that are the results of steps (11), (12), (13), and (14). Swap the two inner blocks (except for the last round) and that is the input to the next round.

After the eighth round, there is a final output transformation:

1. Multiply X_1 and the first subkey.
2. Add X_2 and the second subkey.
3. Add X_3 and the third subkey.
4. Multiply X_4 and the fourth subkey.

Finally, the four sub-blocks are reattached to produce the ciphertext.

Creating the subkeys is also easy. The algorithm uses 52 of them (six for each of the eight rounds and four more for the output transformation). First, the 128-bit key is divided into eight 16-bit subkeys. These are the first eight subkeys for the algorithm (the six for the first round, and the first two for the second round). Then, the key is rotated 25 bits to the left and again divided into eight subkeys. The first four are used in round 2; the last four

are used in round 3. The key is rotated another 25 bits to the left for the next eight subkeys, and so on until the end of the algorithm." [6]

"The very simple key schedule makes IDEA subject to a class of weak keys; some keys containing a large number of 0 bits produce weak encryption.[9] These are of little concern in practice, being sufficiently rare that they are unnecessary to avoid explicitly when generating keys randomly." [2]

"Decryption is exactly the same, except that the subkeys are reversed and slightly different. The decryption subkeys are either the additive or multiplicative inverses of the encryption subkeys. (For the purposes of IDEA, the all-zero sub-block is considered to represent $2^{16} = -1$ for multiplication modulo $2^{16} + 1$; thus the multiplicative inverse of 0 is 0). Calculating these takes some doing, but you only have to do it once for each decryption key." [6]

round r	$K'^{(r)}_1$	$K'^{(r)}_2$	$K'^{(r)}_3$	$K'^{(r)}_4$	$K'^{(r)}_5$	$K'^{(r)}_6$
$r = 1$	$(K^{(10-r)}_1)^{-1}$	$-K^{(10-r)}_2$	$-K^{(10-r)}_3$	$(K^{(10-r)}_4)^{-1}$	$K^{(9-r)}_5$	$K^{(9-r)}_6$
$2 \leq r \leq 8$	$(K^{(10-r)}_1)^{-1}$	$-K^{(10-r)}_3$	$-K^{(10-r)}_2$	$(K^{(10-r)}_4)^{-1}$	$K^{(9-r)}_5$	$K^{(9-r)}_6$
$r = 9$	$(K^{(10-r)}_1)^{-1}$	$-K^{(10-r)}_2$	$-K^{(10-r)}_3$	$(K^{(10-r)}_4)^{-1}$	—	—

Figure 7: IDEA decryption subkeys $K'^{(r)}_i$ derived from encryption subkeys $K^{(r)}_i$. [5]

The following figures provide an example of encryption and decryption of a 64-bit plain text message M using a 128-bit key K.

128-bit key $K = (1, 2, 3, 4, 5, 6, 7, 8)$							64-bit plaintext $M = (0, 1, 2, 3)$			
r	$K^{(r)}_1$	$K^{(r)}_2$	$K^{(r)}_3$	$K^{(r)}_4$	$K^{(r)}_5$	$K^{(r)}_6$	X_1	X_2	X_3	X_4
1	0001	0002	0003	0004	0005	0006	00f0	00f5	010a	0105
2	0007	0008	0400	0600	0800	0a00	222f	21b5	f45e	e959
3	0c00	0e00	1000	0200	0010	0014	0f86	39be	8ee8	1173
4	0018	001c	0020	0004	0008	000c	57df	ac58	c65b	ba4d
5	2800	3000	3800	4000	0800	1000	8e81	ba9c	f77f	3a4a
6	1800	2000	0070	0080	0010	0020	6942	9409	e21b	1c64
7	0030	0040	0050	0060	0000	2000	99d0	c7f6	5331	620e
8	4000	6000	8000	a000	c000	e001	0a24	0098	ec6b	4925
9	0080	00c0	0100	0140	—	—	11fb	ed2b	0198	6de5

Figure 8: IDEA encryption sample. [5]

$K = (1,2,3,4,5,6,7,8)$							$C =$ (11fb,ed2b,0198,6de5)			
r	$K'^{(r)}_1$	$K'^{(r)}_2$	$K'^{(r)}_3$	$K'^{(r)}_4$	$K'^{(r)}_5$	$K'^{(r)}_6$	X_1	X_2	X_3	X_4
1	fe01	ff40	ff00	659a	c000	e001	d98d	d331	27f6	82b8
2	fffd	8000	a000	cccc	0000	2000	bc4d	e26b	9449	a576
3	a556	ffb0	ffc0	52ab	0010	0020	0aa4	f7ef	da9c	24e3
4	554b	ff90	e000	fe01	0800	1000	ca46	fe5b	dc58	116d
5	332d	c800	d000	fffd	0008	000c	748f	8f08	39da	45cc
6	4aab	ffe0	ffe4	c001	0010	0014	3266	045e	2fb5	b02e
7	aa96	f000	f200	ff81	0800	0a00	0690	050a	00fd	1dfa
8	4925	fc00	fff8	552b	0005	0006	0000	0005	0003	000c
9	0001	fffe	fffd	c001	—	—	0000	0001	0002	0003

Figure 9: IDEA decryption sample. [5]

References

[1] Block cipher, May 2016. URL https://en.wikipedia.org/w/index.php?title=Block_cipher&oldid=718649413. Page Version ID: 718649413.

[2] International Data Encryption Algorithm, February 2016. URL https://en.wikipedia.org/w/index.php?title=International_Data_Encryption_Algorithm&oldid=704684534. Page Version ID: 704684534.

[3] Block cipher mode of operation, May 2016. URL https://en.wikipedia.org/w/index.php?title=Block_cipher_mode_of_operation&oldid=721473432. Page Version ID: 721473432.

[4] Symmetric-key algorithm, May 2016. URL https://en.wikipedia.org/w/index.php?title=Symmetric-key_algorithm&oldid=721229454. Page Version ID: 721229454.

[5] Alfred J. Menezes, A. J. Menezes, and Menezes. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, edición: new. edition, October 1996. ISBN 978-0-8493-8523-0.

[6] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley, New York, 2nd edition edition, October 1996. ISBN 978-0-471-11709-4.