A SIMPLIFIED IDEA ALGORITHM

NICK HOFFMAN

Abstract. In this paper, a simplified version of the International Data Encryption Algorithm (IDEA) is described. This simplified version, like simplified versions of DES [8] [12] and AES [6] [7] that have appeared in print, is intended to help students understand the algorithm by providing a version that permits examples to be worked by hand. IDEA is useful teaching tool to help students bridge the gap between DES and AES.

1. Introduction

The International Data Encryption Algorithm (IDEA) is a symmetric-key, block cipher. It was published in 1991 by Lai, Massey, and Murphy [3]. IDEA is a modification of the Proposed Encryption Standard (PES) that was published in 1990 by Lai and Massy [1]; PES was designed as a replacement for the Data Encryption Standard (DES). The algorithm was modified and published in 1991 after Biham and Shamir described the technique of differential cryptanalysis. The new algorithm was called the Improved Proposed Encryption Standard (IPES); its name changed to IDEA in 1992. IDEA is a candidate block cipher to the NESSIE Project. NESSIE is a project within the Information Societies Technology (IST) Program of the European Commission [3].

In the Second Edition (1996) of *Applied Cryptography* Bruce Schneier [9] describes IDEA as "... the best and most secure block algorithm available to the public at this time;" however, in 1999 [10] he began to recommend newer algorithms because IDEA "...isn't very fast ... [and] IDEA is patented."

Although IDEA did not replace DES, it was incorporated into Pretty Good Privacy (PGP).

The algorithm is patented and licensed by MediaCrypt. MediaCrypt now offers a successor algorithm IDEA NXT.

2. Description of the Encryption Algorithm

IDEA encrypts a 64-bit block of plaintext to 64-bit block of ciphertext. It uses a 128-bit key. The algorithm consists of eight identical rounds and a "half" round final transformation.

Today, because of 128-bit cryptosystems like AES, IDEA is obsolete, but its algorithm can be a useful teaching tool to help students bridge the gap between DES, which uses XOR but no algebraic operations, and AES, which requires understanding of algebraic operations on finite fields. IDEA uses algebraic operations, but it is only necessary to understand modular addition and modular multiplication to understand the IDEA algorithm.

Key words and phrases. IDEA, symmetric-key ciphers, block ciphers.

The algebraic idea behind IDEA is the mixing of three incompatible algebraic operations on 16-bit blocks: bitwise XOR, addition modulo 2^{16}, and multiplication modulo $2^{16}+1$.

There are 2^{16} possible 16-bit blocks: 0000000000000000, ..., 1111111111111111, which represent the integers 0, ..., $2^{16}-1$. Each operation with the set of possible 16-bit blocks is an algebraic group. Bitwise XOR is bitwise addition modulo 2, and addition modulo 2^{16} is the usual group operation. Some spin must be put on the elements – the 16-bit blocks – to make sense of multiplication modulo $2^{16}+1$, however. 0 (i.e., 0000000000000000) is not an element of the multiplicative group because it has no inverse, but by thinking of the elements of the group instead as 0000000000000001, ..., 1111111111111111, 0000000000000000, which now represent the integers 1, ..., $2^{16}-1$, 2^{16}, everything works for multiplication. $2^{16} \equiv -1 \mod 2^{16}+1$, and 0000000000000000 is its own inverse under multiplication modulo $2^{16}+1$.

For a description of IDEA, we follow Schneier [9], who breaks the encryption algorithm into fourteen steps. (Another source for the algorithm is [5].) For each of the eight complete rounds, the 64-bit plaintext block is split into four 16-bit sub-blocks: X_1, X_2, X_3, X_4. The 64-bit input block is the concatenation of the sub-blocks: $X_1 \parallel X_2 \parallel X_3 \parallel X_4$, where $\parallel$ denotes concatenation. Each complete round requires six subkeys. The 128-bit key is split into eight 16-bit blocks, which become eight subkeys. The first six subkeys are used in round one, and the remaining two subkeys are used in round two. We will discuss the generation of the remaining keys in the next section.

Each round uses each of the three algebraic operations: bitwise XOR, addition modulo 2^{16}, and multiplication modulo $2^{16}+1$.

Here are the fourteen steps of a complete round (multiply means multiplication modulo $2^{16}+1$, and add means addition modulo 2^{16}):

1. Multiply X_1 and the first subkey Z_1.
2. Add X_2 and the second subkey Z_2.
3. Add X_3 and the third subkey Z_3.
4. Multiply X_4 and the fourth subkey Z_4.
5. Bitwise XOR the results of steps 1 and 3.
6. Bitwise XOR the results of steps 2 and 4.
7. Multiply the result of step 5 and the fifth subkey Z_5.
8. Add the results of steps 6 and 7.
9. Multiply the result of step 8 and the sixth subkey Z_6.
10. Add the results of steps 7 and 9.
11. Bitwise XOR the results of steps 1 and 9.
12. Bitwise XOR the results of steps 3 and 9.
13. Bitwise XOR the results of steps 2 and 10.
14. Bitwise XOR the results of steps 4 and 10.

For every round except the final transformation, a swap occurs, and the input to the next round is: result of step 11 $\parallel$ result of step 13 $\parallel$ result of step 12 $\parallel$ result of step 14, which becomes $X_1 \parallel X_2 \parallel X_3 \parallel X_4$, the input for the next round.

After round 8, a ninth "half round" final transformation occurs:

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.

The concatenation of the blocks is the output.

3. Key Scheduling

Each of the eight complete rounds requires six subkeys, and the final transformation "half round" requires four subkeys; so, the entire process requires 52 subkeys.

The 128-bit key is split into eight 16-bit subkeys. Then the bits are shifted to the left 25 bits. The resulting 128-bit string is split into eight 16-bit blocks that become the next eight subkeys. The shifting and splitting process is repeated until 52 subkeys are generated.

The shifts of 25 bits ensure that repetition does not occur in the subkeys.

Six subkeys are used in each of the 8 rounds. The final 4 subkeys are used in the ninth "half round" final transformation.

4. The Simplified Encryption Algorithm

The simplified IDEA encrypts a 16-bit block of plaintext to a 16-bit block of ciphertext. It uses a 32-bit key. The simplified algorithm consists of four identical rounds and a "half round" final transformation.

The simplified algorithm mixes three algebraic operations on nibbles (4-bit blocks): bitwise XOR, addition modulo $2^4 (= 16)$, and multiplication modulo $2^4 + 1 (= 17)$. There are 16 possible nibbles: 0000, ..., 1111, which represent 0, ..., 15, for addition modulo 16. The 16 nibbles are thought of as 0001, ..., 1111, 0000, which represent 1, ..., 15, 16, for multiplication modulo 17. Notice that 0000, which is 16, is congruent to -1 modulo 17. 0000 is its own inverse under multiplication modulo 17

The 32-bit key, say 11011100011011110011111101011001 is split into eight nibbles 1101 1100 0110 1111 0011 1111 0101 1001. The first six nibbles are used as the subkeys for round 1. The remaining two nibbles are the first two subkeys for round 2. Then the bits are shifted cyclically 6 places to the left, and the new 32-bit string is split into eight nibbles that become the next eight subkeys. The first four of these nibbles are used to complete the subkeys needed for round 2, and the remaining four subkeys are used in round 3. The shifting and splitting process is repeated until all 28 subkeys are generated.

The 32-bit key is 1101 1100 0110 1111 0011 1111 0101 1001.

	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6
Round 1	1101	1100	0110	1111	0011	1111
Round 2	0101	1001$\star$	0001	1011	1100	1111
Round 3	1101	0110	0111	0111$\star$	1111	0011
Round 4	1111	0101	1001	1101	1100	0110$\star$
Round 5	1111	1101	0110	0111		

Encryption key schedule
$\star$ denotes a shift of bits

Six subkeys are used in each of the 4 rounds. The final 4 subkeys are used in the fifth "half round" final transformation.

As an example, we will encrypt the plaintext message 1001110010101100 using the key 11011100011011110011111111.

The ciphertext message is 1011101101001011.

5. Simplified Decryption Algorithm

IDEA decrypts using the same steps as encryption, but new keys must be generated for decryption.

K_j^i denotes the j-th decryption key of decryption round i. Z_j^i denotes the j-th encryption key of encryption round i. For the first decryption round: $K_1^1 = (Z_1^5)^{-1}$, where $(Z_1^5)^{-1}$ denotes the multiplicative inverse of the first encryption key of encryption round 5 – the "half round" final transformation – modulo 17; $K_2^1 = -Z_2^5$, where $-Z_2^5$ denotes the additive inverse of the second encryption key of encryption round 5 modulo 16; $K_3^1 = -Z_3^5$; $K_4^1 = (Z_4^5)^{-1}$; $K_5^1 = Z_5^4$; and $K_6^1 = Z_6^4$. The decryption keys are similarly generated in the remaining complete decryption rounds. The decryption keys for the final transformation "half round" are: $K_1^5 = (Z_1^1)^{-1}$, $K_2^5 = -Z_2^1$, $K_3^5 = -Z_3^1$, and $K_4^5 = (Z_4^1)^{-1}$.

Number in binary	Integer	Inverse in binary	Inverse in integer
0000	0	0000	0
0001	1	1111	15
0010	2	1110	14
0011	3	1101	13
0100	4	1100	12
0101	5	1011	11
1100	6	1010	10
0111	7	1001	9
1000	8	1000	8
1001	9	0111	7
1010	10	0110	6
1011	11	0101	5
1100	12	0100	4
1101	13	0011	3
1110	14	0010	2
1111	15	0001	1

Inverses of nibbles for addition modulo 16

Number in binary	Integer	Inverse in binary	Inverse in integer
0001	1	0001	1
0010	2	1001	9
0011	3	0110	6
0100	4	1101	13
0101	5	0111	7
0110	6	0011	3
0111	7	0101	5
1000	8	1111	15
1001	9	0010	2
1010	10	1100	12
1011	11	1110	14
1100	12	1010	10
1101	13	0100	4
1110	14	1011	11
1111	15	1000	8
0000	16 = -1	0000	16 = -1

Inverses of nibbles for multiplication modulo 17

For our example the decryption keys are:

	K_1	K_2	K_3	K_4	K_5	K_6
Round 1	1000	0011	1010	0101	1100	0110
Round 2	1000	1011	0111	0100	1111	0011
Round 3	0100	1010	1001	0101	1100	1111
Round 4	0111	0111	1111	1110	0011	1111
Round 5	0100	0100	1010	1000		

Decryption key schedule

Although it is difficult to "see through" the decryption process, a sense of what happens can be obtained by doing an example by hand. Decryption is an example of the "shoes and socks principle" – during decryption, the last encryption is the first removed.

It worked! The original plaintext message 1001110010101100 is returned.

6. Design Principles

Shannon's 1949 paper [11] set the standard for modern cryptosystems. It requires confusion (i.e., there should not be a simple relationship between the ciphertext and the key) and diffusion (i.e., ideally, every plaintext bit should influence every ciphertext bit and every key bit should influence every ciphertext bit).

The IDEA algorithm achieves confusion by mixing the three operations bitwise XOR, addition modulo 2^{16}, and multiplication modulo $2^{16}+1$ on 16-bit blocks. The operations are arranged so that the output of one operation is never the input to another operation of the same type. The operations are incompatible in the sense that no two of them satisfy a distributive law, for example, $a \oplus (b \odot c) \neq$

$(a \oplus b) \odot (a \oplus c)$, and no two of them satisfy an associative law, for example, $a \oplus (b \odot c) \neq (a \oplus b) \odot c$.

The IDEA algorithm achieves diffusion by the multiplication-addition structure that appears, for example, in steps 7, 8, 9, and 10 of each round.

IDEA exhibits a generalization of the pure Feistel structure of DES by mixing three algebraic operations. The three algebraic operations are relatively easy to implement in software and hardware. Similar ideas appeared later in AES. Unlike DES, IDEA avoids the need for "lookup tables."

7. Conclusion

IDEA is a well-known cipher that has been analyzed by many researchers for the past decade, and, yet, no attack against five or more of its 8.5 rounds has been found. Due to its strength against cryptanalytic attacks and due to its inclusion in several popular cryptographic packages, IDEA is widely used. [4]

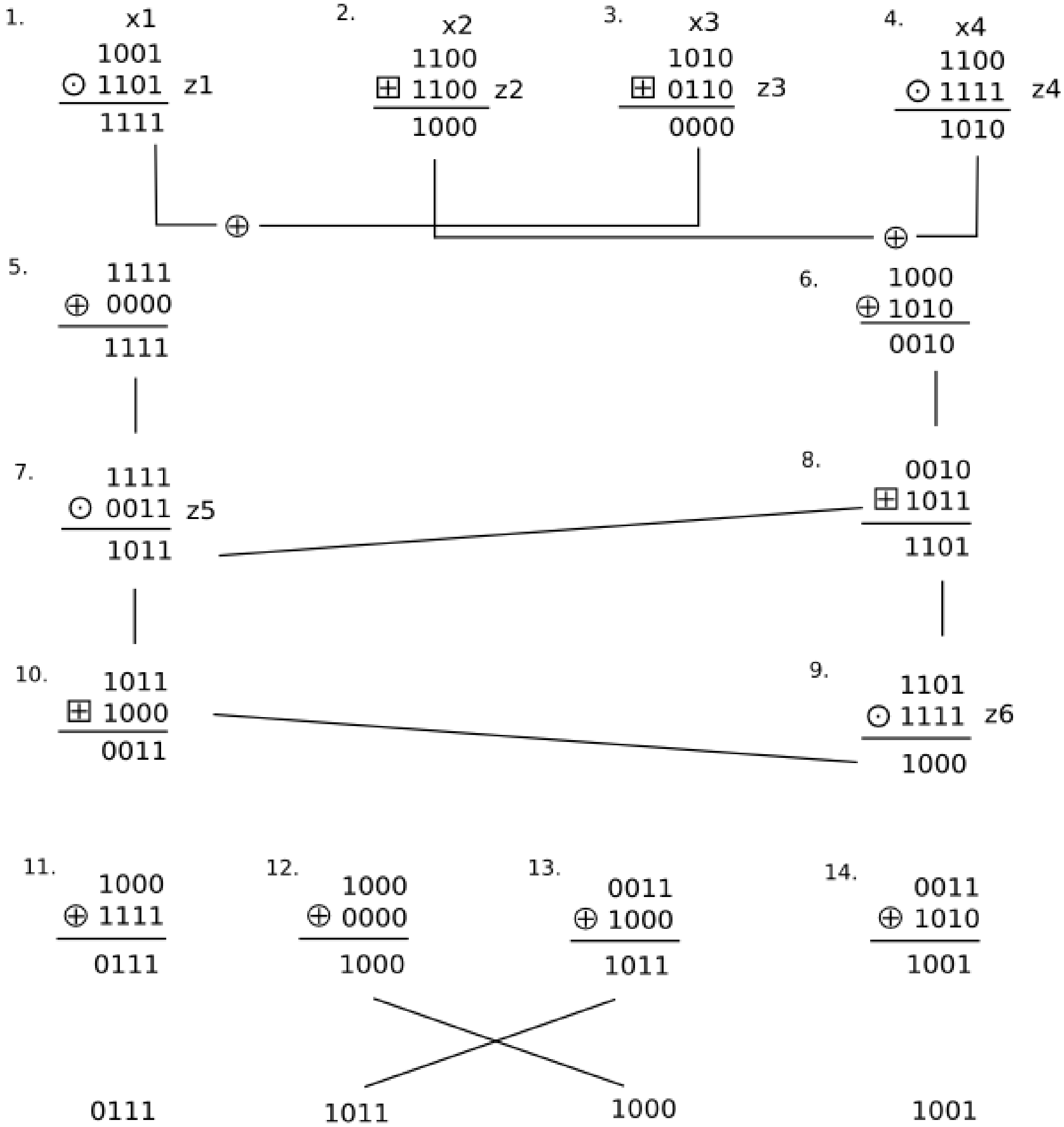
The Simplified IDEA algorithm is not intended to be compared for efficiency or security with simplified versions of DES or AES. The Simplified IDEA algorithm is intended to help students understand the IDEA algorithm by providing a version of IDEA that permits examples to be worked by hand and to provide a comparison of the method of IDEA with the methods of DES and AES.

References

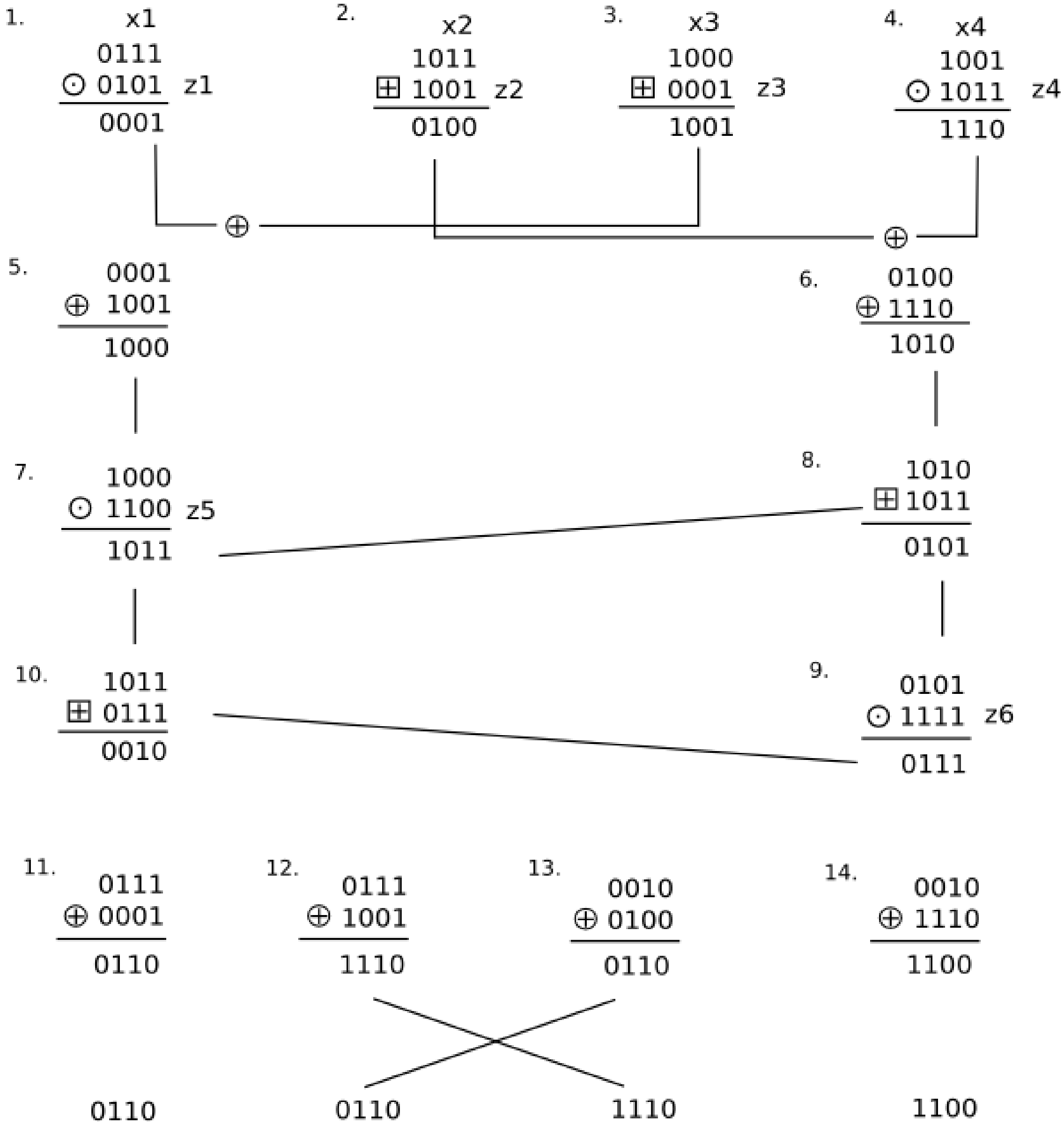
1. Lai, Xuejia, and Massey, James L., A Proposal for a New Block Encryption Standard, *Advances in Cryptology - EUROCRYPT '90, Lecture Notes in Computer Science*, Springer-Verlag, 1991: 389-404.
2. Lai, X., Massey, J., and Murphy, S., Markov Ciphers and Differential Cryptanalysis, *Advances in Cryptology – EUROCRYPT '91, Lecture Notes in Computer Science*, Springer-Verlag, 1991: 17-38.
3. Mediacrypt AG, The IDEA Block Cipher, submission to the NESSIE Project, http://cryptonessie.org
4. Meier, W., On the Security of the IDEA block cipher, *Advances in Cryptology*
5. Menezes, A., van Oorschot, P., and Vanstone, S. 1996. *Handbook of Applied Cryptography*. CRC Press. This book may downloaded from http://www.cacr.math.uwaterloo.ca/hac/
6. Musa, M., Shaefer, E., and Wedig S. 2003. A Simplified AES Algorithm and its Linear and Differential Cryptanalysis. *Cryptologia*. 17 (2): 148 - 177.
7. Phan, R. 2002. Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students. *Cryptologia*. 26 (4): 283 - 306.
8. Schaefer, E. 1996. A Simplified Data Encryption Standard Algorithm. *Cryptologia*. 20 (1): 77 - 84.
9. Schneier, B. 1996. *Applied Cryptography, Second Edition*. Wiley.
10. Schneier, B. 1999. http://slashdot.org/interviews/99/10/29/0832246.shtml.
11. Shannon, Claude, Communications theory of Secrecy Systems, *Bell systems Technical Journal*. 28 (4): 656 - 715.
12. Trappe, W. and Washington, L. 2006. *Introduction to Cryptography with Coding Theory, Second Edition*. Prentice Hall.

Department of Mathematics, Northern Kentucky University
E-mail address: hoffmannick@gmail.com

Round 1
Encryption



Round 2
Encryption



Round 3
Encryption

1. x1
$$\begin{array}{rl} & 0110 \\ \odot & 1101 \quad z1 \\ \hline & 1010 \end{array}$$

2. x2
$$\begin{array}{rl} & 0110 \\ \boxplus & 0110 \quad z2 \\ \hline & 1100 \end{array}$$

3. x3
$$\begin{array}{rl} & 1110 \\ \boxplus & 0111 \quad z3 \\ \hline & 0101 \end{array}$$

4. x4
$$\begin{array}{rl} & 1100 \\ \odot & 0111 \quad z4 \\ \hline & 0000 \end{array}$$

$\oplus$ $\oplus$

5.
$$\begin{array}{rl} & 1010 \\ \oplus & 0101 \\ \hline & 1111 \end{array}$$

6.
$$\begin{array}{rl} & 1100 \\ \oplus & 0000 \\ \hline & 1100 \end{array}$$

7.
$$\begin{array}{rl} & 1111 \\ \odot & 1111 \quad z5 \\ \hline & 0100 \end{array}$$

8.
$$\begin{array}{rl} & 1100 \\ \boxplus & 0100 \\ \hline & 0000 \end{array}$$

10.
$$\begin{array}{rl} & 0100 \\ \boxplus & 1110 \\ \hline & 0010 \end{array}$$

9.
$$\begin{array}{rl} & 0000 \\ \odot & 0011 \quad z6 \\ \hline & 1110 \end{array}$$

11.
$$\begin{array}{rl} & 1110 \\ \oplus & 1010 \\ \hline & 0100 \end{array}$$

12.
$$\begin{array}{rl} & 1110 \\ \oplus & 0101 \\ \hline & 1011 \end{array}$$

13.
$$\begin{array}{rl} & 0010 \\ \oplus & 1100 \\ \hline & 1110 \end{array}$$

14.
$$\begin{array}{rl} & 0010 \\ \oplus & 0000 \\ \hline & 0010 \end{array}$$

0100 1110 1011 0010

Round 4
Encryption

1. $x1$
$$\begin{array}{rl} & 0100 \\ \odot & 1111 \quad z1 \\ \hline & 1001 \end{array}$$

2. $x2$
$$\begin{array}{rl} & 1110 \\ \boxplus & 0101 \quad z2 \\ \hline & 0011 \end{array}$$

3. $x3$
$$\begin{array}{rl} & 1011 \\ \boxplus & 1001 \quad z3 \\ \hline & 0100 \end{array}$$

4. $x4$
$$\begin{array}{rl} & 0010 \\ \odot & 1101 \quad z4 \\ \hline & 1001 \end{array}$$

$\oplus$ $\oplus$

5.
$$\begin{array}{rl} & 1001 \\ \oplus & 0100 \\ \hline & 1101 \end{array}$$

6.
$$\begin{array}{rl} & 1001 \\ \oplus & 0011 \\ \hline & 1010 \end{array}$$

7.
$$\begin{array}{rl} & 1101 \\ \odot & 1100 \quad z5 \\ \hline & 0011 \end{array}$$

8.
$$\begin{array}{rl} & 1010 \\ \boxplus & 0011 \\ \hline & 1101 \end{array}$$

9.
$$\begin{array}{rl} & 1101 \\ \odot & 0110 \quad z6 \\ \hline & 1010 \end{array}$$

10.
$$\begin{array}{rl} & 0011 \\ \boxplus & 1010 \\ \hline & 1101 \end{array}$$

11.
$$\begin{array}{rl} & 1001 \\ \oplus & 1010 \\ \hline & 0011 \end{array}$$

12.
$$\begin{array}{rl} & 0100 \\ \oplus & 1010 \\ \hline & 1110 \end{array}$$

13.
$$\begin{array}{rl} & 0011 \\ \oplus & 1101 \\ \hline & 1110 \end{array}$$

14.
$$\begin{array}{rl} & 1001 \\ \oplus & 1101 \\ \hline & 0100 \end{array}$$

0011 1110 1110 0100

Round 5
Encryption
Final Transformation

1. x1

$$\begin{array}{r} 0011 \\ \odot\ 1111 \\ \hline 1011 \end{array} \quad z1$$

1011

2. x2

$$\begin{array}{r} 1110 \\ \boxplus\ 1101 \\ \hline 1011 \end{array} \quad z2$$

1011

3. x3

$$\begin{array}{r} 1110 \\ \boxplus\ 0110 \\ \hline 0100 \end{array} \quad z3$$

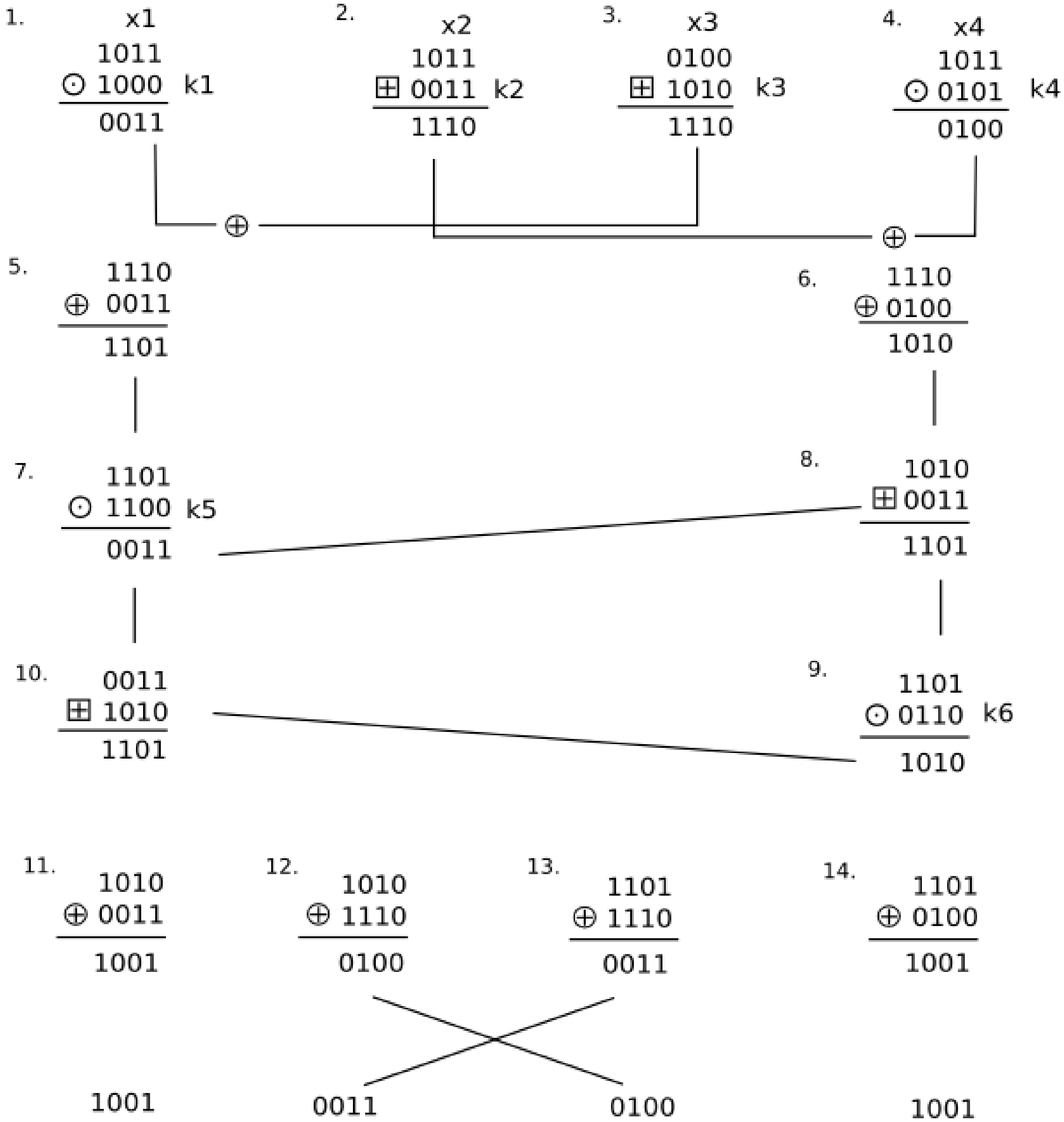
0100

4. x4

$$\begin{array}{r} 0100 \\ \odot\ 0111 \\ \hline 1011 \end{array} \quad z4$$

1011

Round 1
Decryption



Round 2
Decryption

1. x1

$$\begin{array}{r} 1001 \\ \odot\ 1000 \\ \hline 0100 \end{array} \quad k1$$

2. x2

$$\begin{array}{r} 0011 \\ \boxplus\ 1011 \\ \hline 1110 \end{array} \quad k2$$

3. x3

$$\begin{array}{r} 0100 \\ \boxplus\ 0111 \\ \hline 1011 \end{array} \quad k3$$

4. x4

$$\begin{array}{r} 1001 \\ \odot\ 0100 \\ \hline 0010 \end{array} \quad k4$$

5.

$$\begin{array}{r} 1011 \\ \oplus\ 0100 \\ \hline 1111 \end{array}$$

6.

$$\begin{array}{r} 1110 \\ \oplus\ 0010 \\ \hline 1100 \end{array}$$

7.

$$\begin{array}{r} 1111 \\ \odot\ 1111 \\ \hline 0100 \end{array} \quad k5$$

8.

$$\begin{array}{r} 1100 \\ \boxplus\ 0100 \\ \hline 0000 \end{array}$$

10.

$$\begin{array}{r} 0100 \\ \boxplus\ 1110 \\ \hline 0010 \end{array}$$

9.

$$\begin{array}{r} 0000 \\ \odot\ 0011 \\ \hline 1110 \end{array} \quad k6$$

11.

$$\begin{array}{r} 1110 \\ \oplus\ 0100 \\ \hline 1010 \end{array}$$

12.

$$\begin{array}{r} 1110 \\ \oplus\ 1011 \\ \hline 0101 \end{array}$$

13.

$$\begin{array}{r} 0010 \\ \oplus\ 1110 \\ \hline 1100 \end{array}$$

14.

$$\begin{array}{r} 0010 \\ \oplus\ 0010 \\ \hline 0000 \end{array}$$

1010 1100 0101 0000

Round 3
Decryption

1. x1
$$\begin{array}{r} 1010 \\ \odot\ 0100 \\ \hline 0110 \end{array}\quad k1$$

2. x2
$$\begin{array}{r} 1100 \\ \boxplus\ 1010 \\ \hline 0110 \end{array}\quad k2$$

3. x3
$$\begin{array}{r} 0101 \\ \boxplus\ 1001 \\ \hline 1110 \end{array}\quad k3$$

4. x4
$$\begin{array}{r} 0000 \\ \odot\ 0101 \\ \hline 1100 \end{array}\quad k4$$

$\oplus$ $\oplus$

5.
$$\begin{array}{r} 0110 \\ \oplus\ 1110 \\ \hline 1000 \end{array}$$

6.
$$\begin{array}{r} 1100 \\ \oplus\ 0110 \\ \hline 1010 \end{array}$$

7.
$$\begin{array}{r} 1000 \\ \odot\ 1100 \\ \hline 1011 \end{array}\quad k5$$

8.
$$\begin{array}{r} 1010 \\ \boxplus\ 1011 \\ \hline 0101 \end{array}$$

10.
$$\begin{array}{r} 1011 \\ \boxplus\ 0111 \\ \hline 0010 \end{array}$$

9.
$$\begin{array}{r} 0101 \\ \odot\ 1111 \\ \hline 0111 \end{array}\quad k6$$

11.
$$\begin{array}{r} 0111 \\ \oplus\ 0110 \\ \hline 0001 \end{array}$$

12.
$$\begin{array}{r} 0111 \\ \oplus\ 1110 \\ \hline 1001 \end{array}$$

13.
$$\begin{array}{r} 0010 \\ \oplus\ 0110 \\ \hline 0100 \end{array}$$

14.
$$\begin{array}{r} 0010 \\ \oplus\ 1100 \\ \hline 1110 \end{array}$$

0001 0100 1001 1110

Round 4
Decryption

1. x1
$$\begin{array}{rl} & 0001 \\ \odot & 0111 \quad k1 \\ \hline & 0111 \end{array}$$

2. x2
$$\begin{array}{rl} & 0100 \\ \boxplus & 0111 \quad k2 \\ \hline & 1011 \end{array}$$

3. x3
$$\begin{array}{rl} & 1001 \\ \boxplus & 1111 \quad k3 \\ \hline & 1000 \end{array}$$

4. x4
$$\begin{array}{rl} & 1110 \\ \odot & 1110 \quad k4 \\ \hline & 1001 \end{array}$$

5.
$$\begin{array}{rl} & 0111 \\ \oplus & 1000 \\ \hline & 1111 \end{array}$$

6.
$$\begin{array}{rl} & 1011 \\ \oplus & 1001 \\ \hline & 0010 \end{array}$$

7.
$$\begin{array}{rl} & 1111 \\ \odot & 0011 \quad k5 \\ \hline & 1011 \end{array}$$

8.
$$\begin{array}{rl} & 0010 \\ \boxplus & 1011 \\ \hline & 1101 \end{array}$$

9.
$$\begin{array}{rl} & 1101 \\ \odot & 1111 \quad k6 \\ \hline & 1000 \end{array}$$

10.
$$\begin{array}{rl} & 1011 \\ \boxplus & 1000 \\ \hline & 0011 \end{array}$$

11.
$$\begin{array}{rl} & 1000 \\ \oplus & 0111 \\ \hline & 1111 \end{array}$$

12.
$$\begin{array}{rl} & 1000 \\ \oplus & 1000 \\ \hline & 0000 \end{array}$$

13.
$$\begin{array}{rl} & 0011 \\ \oplus & 1101 \\ \hline & 1000 \end{array}$$

14.
$$\begin{array}{rl} & 0011 \\ \oplus & 1001 \\ \hline & 1010 \end{array}$$

1111 1000 0000 1010

Round 5
Decryption
Final Transformation

1. x1

$$\begin{array}{r} 1111 \\ \odot\ 0100 \\ \hline 1001 \end{array}\quad k1$$

2. x2

$$\begin{array}{r} 1000 \\ \boxplus\ 0100 \\ \hline 1100 \end{array}\quad k2$$

3. x3

$$\begin{array}{r} 0000 \\ \boxplus\ 1010 \\ \hline 1010 \end{array}\quad k3$$

4. x4

$$\begin{array}{r} 1010 \\ \odot\ 1000 \\ \hline 1100 \end{array}\quad k4$$

1001 1100 1010 1100