Contemporary Engineering Sciences, Vol. 5, 2012, no. 6, 281 - 286

MARC – A New Block Cipher Algorithm

Jeffrey S.-W. Hsiao

Department of Electrical and Computer Engineering
University of Canterbury
Christchurch, New Zealand

T. Aaron Gulliver

Dept. of Electrical and Computer Engineering
University of Victoria
P.O. Box 3055, STN CSC
Victoria, B.C., Canada V8W 3P6
agullive@ece.uvic.ca

Abstract

A new block cipher based on the IDEA algorithm is presented. It incorporates key dependent rotations which provides additional security without a significant increase in computational complexity.

Keywords: Block cipher; Feistel network

1 Introduction

In the present world of data communications and computer technology, there is a growing necessity for protecting data against unauthorised use and illegal duplication. The increase of information transmitted electronically has lead to an increased reliance on cryptography. From a practical standpoint, it is necessary that encryption/decryption be fast even on a typical personal computer. This task is made even more difficult by the high-speed connections available in the home and elsewhere.

This paper introduces a new algorithm called MARC which employs simple modulo arithmetic operations along with XOR and variable rotations. It is based on the structure of IDEA, which was developed in 1991 by Lai and Massey [1].

There are two major classes of encryption, symmetric (private-key encryption) and asymmetric (public-key encryption). Public-key systems are bulky

and slow because of the hard problems they are based on. Private key encryption provides far better speed performance, and the algorithms tend to be much smaller. In these systems, a single key (symmetric) is used to encrypt and decrypt the data, Symmetric keys can be generated quite often. For example, browsers generate new symmetric keys each time they open a secure transaction. A requirement is that the sender and receiver keep the key a secret. Private-key encryption systems can be classified as block ciphers or stream ciphers. The difference between these is that block ciphers encrypt/decrypt blocks of data at a time, whereas stream ciphers can encrypt/decrypt continuously. This paper considers block ciphers.

A block cipher transforms a fixed length block of plaintext into a block of ciphertext by repeated use of rounds. Each round contains a finite number of operations such as addition, XOR, rotation, etc. The number of rounds is typically between 8 and 32.

2 The MARC Algorithm

Most block ciphers are Feistel networks [2]. A Feistel network is an algorithm that is invertible. One algorithm is used for both encryption and decryption except that the keys are applied in reverse order in decryption. A fast and secure algorithm can be achieved by combining simple operations which take few instructions on most general processors. The operations considered here are:

1. XOR
2. Rotation
3. Addition modulo N
4. Multiplication modulo M

The design philosophy is to create a mix of several different algebraic operations, and N and M should differ and have no common factor. In this paper, four operations are employed which are easily implemented in software:

1. ADDITION modulo 2^{16}
2. XOR
3. ROTATION (key dependent)
4. MULTIPLICATION mod $2^{16} + 1$

The MARC algorithm is designed using a structure similar to that of IDEA [2]. MARC has the following properties:

Block size: 64-bit
Key length: 128 bits
Numbers of round: 8
Key pre-computation required
No Table-lookup
Simple operations that are efficient on microprocessors

MARC is an iterated cipher consisting of 8 rounds followed by an output transformation. Each round includes eight 16-bit subkeys, four multiplies, four adds, six XORs and four rotations. The subkeys are determined from the 128 bit input key. A total of 68 16-bit subkeys are required (16 more than IDEA). A subkey generation algorithm similar to that for IDEA [2] can be employed. The use of independent keys would be more secure [2], but would require a total key length of $16 \times 68 = 1088$ bits.

The complete first round and the output transformation are shown in Fig. 1. The four different group operations are arranged so that the output of an operation of one type is never used as the input to an operation of the same type. To achieve good diffusion, each output bit of the first round should depend on every bit of the plaintext and on every bit of the subkeys used in that round. This diffusion is provided in the MARC cipher by the transformation called the multiplication-addition-rotation (MAR) structure shown in Fig. 2.

The MAR structure transforms two 16-bit subblocks into two new 16-bit subblocks, controlled by four 16-bit subkeys. Note that the inputs, U_1 and U_2, are also used to determine the first rotations to the right and left. For any choice of the key subblocks Z_5, Z_6, Z_7 and Z_8, MAR is an invertible transformation.

The algorithm was implemented in the C language for file encryption. The speed was compared with IDEA using a 180 MHz Pentium Pro computer and is summarised below.

Algorithm	Encryption Speed (Kilobytes/second)
IDEA	160
MARC	138

Eight rounds were used with both algorithms. The speed of MARC is not as fast as IDEA due to the addition of rotation operations. However, these extra operations increase the security of the algorithm. The speed of any algorithm is platform dependent, thus the implementation of MARC was not extensively optimised with respect to the hardware architecture. Optimisation for a particular application should improve the encryption speed [2, 3]. Another

way to improve the speed performance is to reduce the number of rounds of the algorithm. However, this degrades the security of the cipher. Thus, there is a trade-off between speed and protection which must be considered in the design process.

3 Conclusions

In this paper, MARC, a new block cipher algorithm was presented. This algorithm mixes four simple algebraic operations. The structure is similar to that of IDEA, but includes rotations. MARC has a standard 64-bit block size, a nominal 128-bit key and requires no table look-up operations.

References

[1] X. Lai, J.L. Massey and S. Murphy, Markov ciphers and Differential cryptanalysis, Advances in Cryptology, Proceedings of CRYPTO '91, Springer-Verlag Lecture Notes in Computer Science, 547 (1992), 17–38.

[2] B. Schneier, Applied Cryptography, Wiley, New York, 1996.

[3] B. Schneier and D. Whiting, Fast software encryption: Designing encryption algorithms for optimal software speed on the Intel Pentium processor, Proceedings of the International Workshop in Fast Software Encryption, Springer-Verlag Lecture Notes in Computer Science, 1267 (1997), 242–259.

Received: February, 2012

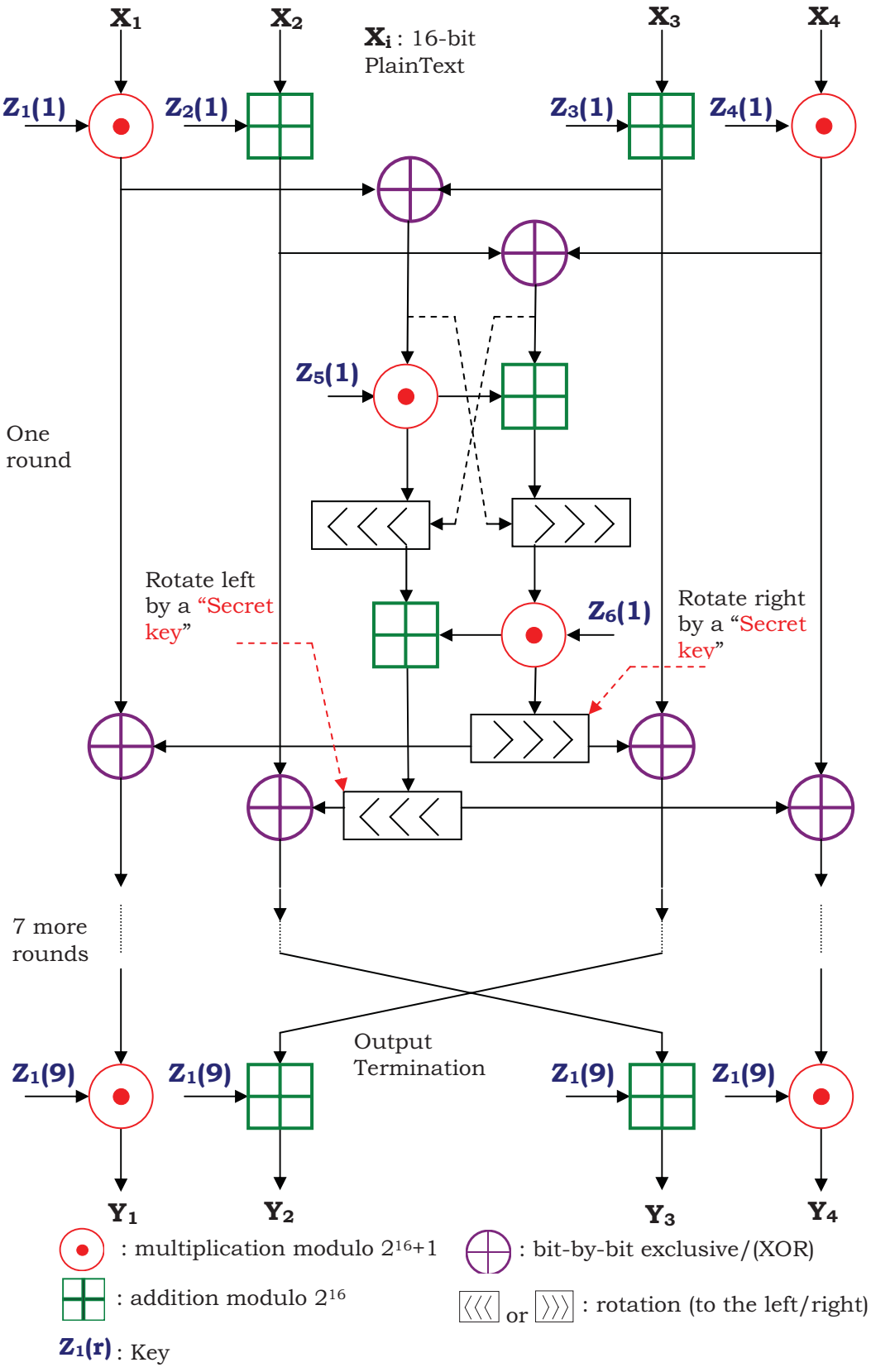

Figure 1: The MARC encryption algorithm.

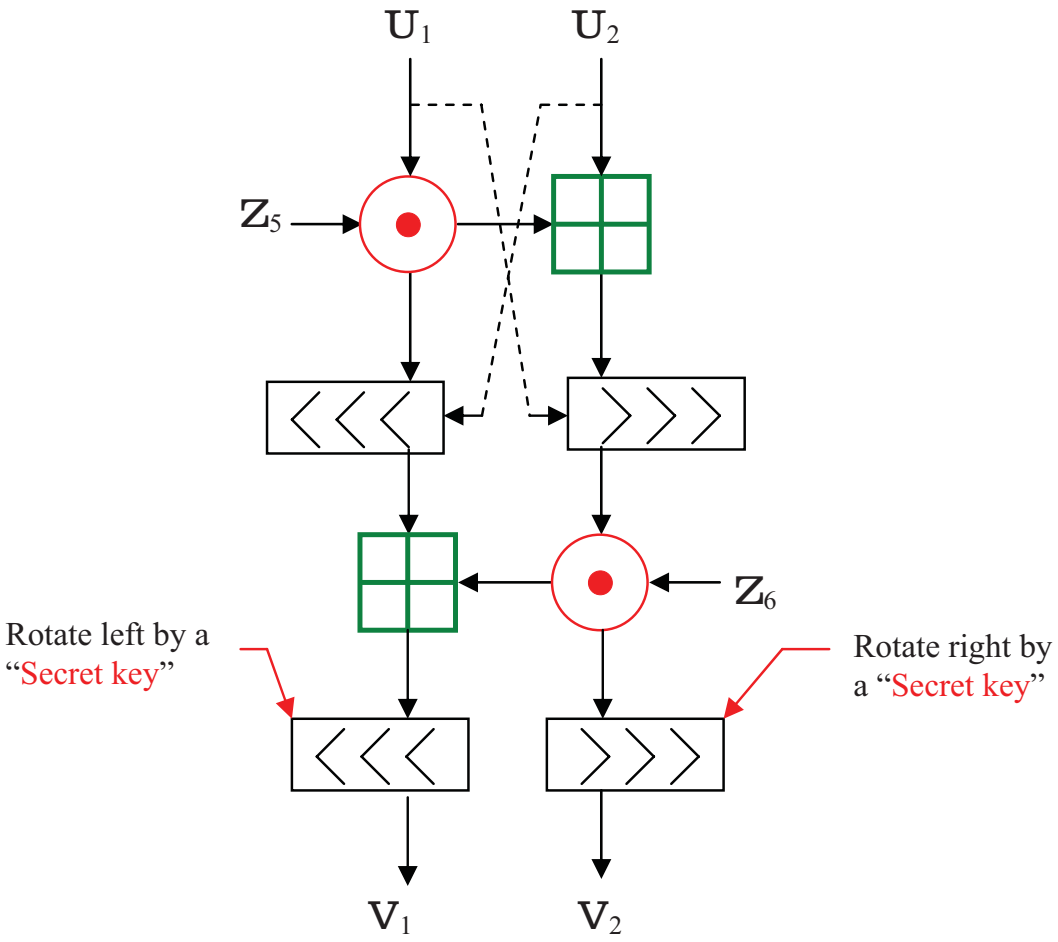

Figure 2: The MARC structure.