International Journal of the Physical Sciences Vol. 5(9), pp. 1338-1343, 18 August, 2010
Available online at http://www.academicjournals.org/ijps
ISSN 1992 - 1950 ©2010 Academic Journals

Full Length Research Paper

Software implementation and performance comparison of popular block ciphers on 8-bit low-cost microcontroller

Murat Çakiroğlu

Department of Computer Science, Faculty of Technical Education, University of Sakarya, Turkey.
E-mail: muratc@sakarya.edu.tr.

Accepted 06 August, 2010

8-bit microcontrollers can be used in a wide range of applications, such as the smart cards, telemedicine, cars and automation systems. In these application areas, security is one of the most important issues. The block ciphers, which can provide high-performance and reasonable security level, are meant to meet the security requirements of the 8-bit microcontrollers. In this paper, we realized the performance evaluation of the popular block cipher algorithms such as AES (Advanced Encryption Standard) (Daemen and Rijmen, 2002), Serpent (Anderson et al., 1998), Camellia (Aoki et al., 2001), CAST5 (Carlisle, 1997), and MARS (Burwick et al., 1998) on 8-bit Atmel microcontroller. The performance of the chosen block ciphers were evaluated in terms of the code/data memory requirement, execution time, and throughput criteria. According to the results obtained from target device, AES has better performance than other block ciphers in respect to throughput and code memory requirement. CAST5 and Camellia can be considered as a good alternative to AES. Serpent has a very bad performance in terms of throughput and Mars has a very bad performance in terms of memory requirement.

Key words: Block cipher, low-cost, 8-bit microcontroller, algorithm, implementation.

INTRODUCTION

Nowadays, 8-bit microcontrollers are widely used in various areas such as smart home appliances, factory automation systems, RFID cards, Point-of-Sales (POS) devices, the new generation phones, handheld computers, various network devices, and cars (Özcerit et al., 2005). Cryptography is an important requirement in such applications to download the program software into a microcontroller securely, perform secure messaging, and for authentication (Meiser et al., 2008). For example, in a POS device, personal information must be encrypted with cryptographic algorithms in such a way as to deny access to important information by malicious person.

Cryptographic algorithms can be divided into two categories; symmetric and asymmetric. Today, asymmetric algorithms are used in many applications such as digital signatures, etc. Although asymmetric algorithms can provide extremely high security level, they require rich resources such as high computing capability and memory capacity. For this reason, it is almost not possible to use an asymmetric encryption algorithm on 8-bit microcontrollers. Symmetric algorithms can be divided into two sub-categories such as block cipher and stream cipher (Stallings, 2005). Block ciphers can be contrasted with stream ciphers. The distinction between the two is not always precise. A stream cipher works on individual digits one at a time and the transformation varies during the encryption. A block cipher, when used in certain modes of operation, acts effectively as a stream cipher. Most block ciphers can work with different keys and data size. Therefore, their security features can be determined according to the application’s security need. These advantages of block ciphers make possible their use in many different environments / platforms.

In this study, the performance analysis of popular block ciphers such as AES, Serpent, Camellia, CAST5, and MARS algorithms were realized on an 8-bit Atmel microcontroller. Comparative performance evaluations of algorithms were presented in terms of memory requirement, execution time, and throughput criteria. Thus, the question of “which block ciphers are more suitable for 8-bit microcontrollers” is answered.

RELATED WORKS

There are various studies in the literature about the performance evaluation of block ciphers on low-cost microcontrollers. For example, Law et al. (2006) evaluated the performance of RC5, RC6, AES, MISTY1, KASUMI, and Camellia block ciphers for 16-bit RISC microcontroller MSP430F149. Rinne et al. (2007) measured the performance of DESL, Hight, SEA, and TEA / XTEA algorithms for 8-bit ATMEL microcontroller. Eisenbarth et al. (2007) measured the performance of AES, Hight, Clefia, mCrypton, DES, DESXL algorithms. In another study, Cakiroglu et al. (2010) implemented and evaluated the SEA, RC6, and AES block ciphers for 8-bit ATMEL microcontroller.

OVERVIEW OF BLOCK CIPHERS

In this section, the features and operating principles of the block ciphers, which are been implemented on 8-bit microcontroller, is briefly described.

The advanced encryption algorithm

AES (Daemen and Rijmen, 2002) was designed by J. Daemen and V. Rijmen, and introduced by National Institute of Standards and Technology in 2001. AES is a symmetric block encryption algorithm, also known as Rijndael, which is the successor of the Data Encryption Standard (DES) algorithm. AES can encrypt 128-bit data blocks by using 128, 192 or 256-bit keys and operates on 4x4 matrixes, which are called states, and each AES round is composed of four stages. AES performs encryption operations in 10, 12, or 14 rounds depending on predetermined key sizes (Daemen and Rijmen, 2002). AES, which can be implemented by hardware or software based methods effectively, is still accepted as the most secure block cipher. There are various attack for reduced version of AES, but there is no known practical attack against full version of AES (Biryukov and Khovratovich, 2009).

Serpent

Serpent (Anderson et al., 1998) is a symmetric key block cipher, which was a finalist in the Advanced Encryption Standard competition. Serpent was developed by Ross Anderson, Eli Biham and Lars Knudsen. It has a block size of 128 bits and can work with different combinations of key size such as 128, 192 or 256 bits. The cipher is a 32-round substitution-permutation network operating on a block of four 32-bit words. Each round applies one of eight 4-bit-to-4-bit s-boxes 32 times in parallel. Serpent was developed so that all operations can be executed in parallel, using 32 1-bit slices. The Serpent cipher has not been patented and can be freely used by anyone (Landau, 2000). In the literature, there are attacks on 10 and 11 round Serpent (Biham et al., 2002). However, 32 round Serpent still secure.

Camellia

Camellia (Aoki et al., 2001) is a block cipher that was analyzed by the European Union's NESSIE project and the Japanese CRYPTREC project. The cipher was designed by Mitsubishi and NTT in 2000. It has a block size of 128-bits and can use 128-bit, 192-bit or 256-bit keys like other AES submissions. Camellia is a Feistel cipher with either 18 rounds (when the key is 128 bits) or 24 rounds (when the key is 192 or 256 bits). It utilizes four 8 x 8-bit S-boxes with input and output affine transformations and logical operations (Matsui et al., 2004). In the literature, there are various attacks on 6, 7 and 11 round Camellia. However, 18 round Camellia is still secure (Law et al., 2006).

CAST5 (CAST-128)

CAST5 (Carlisle, 1997) is a symmetric block cipher with a block-size of 64-bit and a variable key-size of up to 128 bits. The algorithm was developed in 1996 by Carlisle Adams and Stafford Tavares . It is available worldwide on a royalty-free basis for commercial and non-commercial uses. CAST-128 is a 12 or 16-round Feistel network with a 64-bits block size and a key size of between 40 to 128 bits (but only in 8-bit increments). The full 16 rounds are employed when the key size is longer than 80 bits. Components include large 8 × 32-bits s-boxes based on bent functions, key-dependent rotations, modular addition/subtraction and XOR operations (Adams, 1997). There are some attacks on four and six round version of CAST5 (Wang et al., 2009). However, the full version of CAST5 is still secure.

MARS

MARS (Burwick et al., 1998) is a shared-key block cipher, with a block size of 128 bits and a variable key size, ranging from 128 to over 400 bits. It was developed to meet the requirements for a standard for shared-key encryption in the next few decades. MARS was chosen as an AES finalist in August 1999. The project was specifically designed to resist future advances in cryptography by adopting a layered, compartmentalized approach. MARS has a 128-bit block size and a

Table 1. Implementation detail of the block ciphers.

Cipher	AES	Serpent	Camellia	CAST5	MARS
Data block size	128	128	128	64	128
Key size	128	128	128	128	128
Round	10	32	18	16	32

variable key size of between 128 and 448 bits (in 32-bit increments) (Burwick et al., 1998). There have been various attacks on reduced version of MARS cipher. However, full version of MARS is still secure (Kelsey and Schneier B, 2000).

CIPHER IMPLEMENTATION DETAILS AND TOOLS

The software implementations of all the selected block cipher algorithms are realized using C programming language. Today's many embedded applications are developed by C programming language since there is no substantial difference between C and assembler languages in terms of performance and code size criteria. Therefore, we selected and used C programming language as a test language. Implementation details of the selected block ciphers are shown in Table 1.

All the selected algorithms has a block size of 128 bits, except for CAST5. To achieve a fair performance evaluation, key size of all algorithms were chosen equally. The number of rounds is determined by the algorithm according to the chosen key size. For example, AES uses 10 rounds for key size of 128-bits and uses 12 rounds for key size 192-bits.

The implementation and performance evaluation of the block ciphers were realized on 8-bit RISC architecture based Atmega128 (Atmega128, 2010) microcontroller. This microcontroller has a 128KB code and 4KB of data memory. It can be run with 16 MHz crystal frequency up to 16 MIPS. It has 32 general-purpose registers, four timer/counter units, seven 8-bit input /output ports, two UART ports and can execute most of the commands in a single clock cycle. Since it is low-cost and has low power, it is widely preferred in various embedded system applications.

We used the AVRStudio4 (Atmel Corporation, 2010) software for the implementation and performance evaluation of the block ciphers. AVRStudio4 is a development software, which has capability of compiling, debugging, and simulating of AVR microcontrollers. We also used the AVR GCC (AVR-GCC, 2010) compiler to C language support for AVRStudio4 software.

PERFORMANCE COMPARISON OF THE SELECTED BLOCK CIPHER

We have used three parameters namely code/data memory requirement, execution time, and throughput to compare the performance of the selected block ciphers. All of these parameters were obtained by means of AVR Studio (Atmel, 2010) and AVR GCC (AVR GCC, 2010) software.

Data and code memory requirements

Code and data memory requirements of a block cipher are important criteria for the microcontrollers, which has more limited hardware resource than the computers. The prices of microcontrollers vary according to the memory size. Therefore, it is expected from a block cipher that it require lower memory usage. Code memory requirement of the block ciphers is shown in Figure 1 and data memory requirement is shown in Figure 2.

Mars block cipher requires more code and data memory usage than the others. Approximately 31 KB code memory and 392 bytes data memory usage is so big for many microcontrollers. For example, the standard 8051 microcontroller has 4KB of code size and 256 bytes of data size. Under these conditions, it is not possible to use the Mars block cipher on an 8051 microcontroller. The code memory requirements of CAST5 and Camellia algorithms can also be considered high for medium-sized microcontrollers. However, AES and Serpent block ciphers are efficient in terms of code and data memory usage. The data memory usages of the block ciphers are reasonable sizes for most of microcontrollers except for Mars.

Execution time

One of the most important performance criteria of the encryption algorithms is execution time. Execution time is related to many parameters such as structure of the algorithm, the number of rounds, and the selected target device. In this study, we used Atmega128 (Atmega128, 2010) microcontroller with 16 Mhz crystal frequency as a target device. The execution time of the block ciphers is shown in Figure 3.

The execution time of the block ciphers are shown in Figure 3 as three separate parts namely initialization, encryption, and decryption. Key expansion process is performed in the initialization stage. In this stage, the round keys are created from the user key by means of the key scheduling algorithm. Different keys are used in

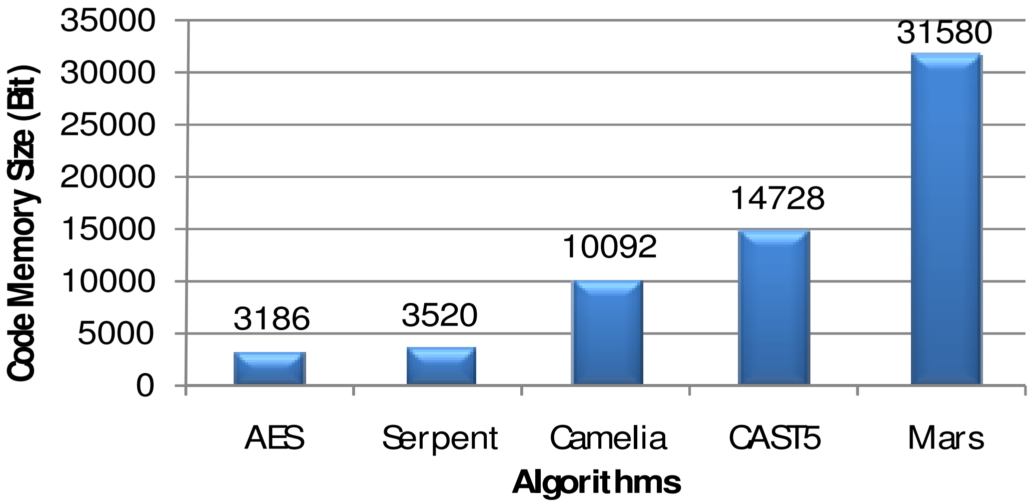

Figure 1. Code memory requirements of the block ciphers.

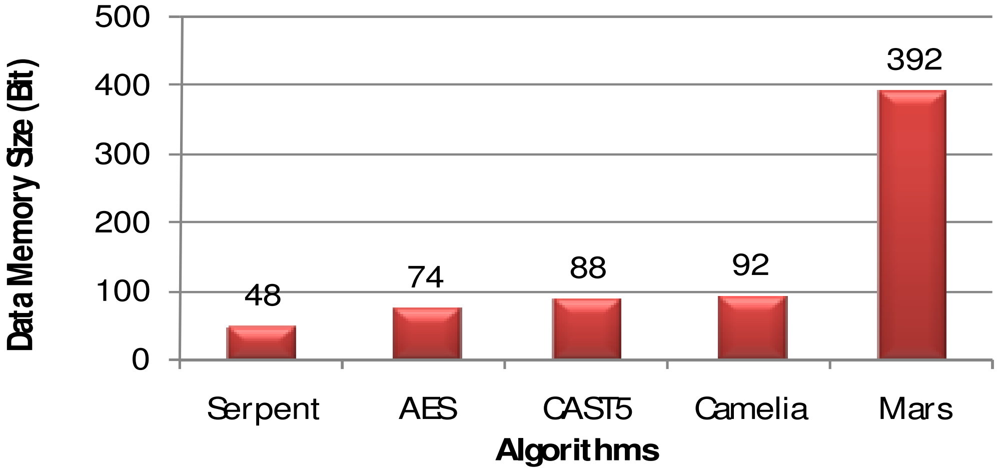

Figure 2. Data memory requirements of the block ciphers.

each round. For example, Serpent block cipher expands the user key up to 33 128-bits sub keys in the initialization stage. Among the selected block cipher algorithm, the AES's key expansion process is the shortest, and Serpent key expansion is the longest. The time spent in the initialization stage is related to the key scheduling method of the block cipher.

Within the encryption stage, the plaintext taken as blocks is encrypted with the round keys obtained in the initialization stage. After the last round, a 128-bits block of ciphertext is created for the 128-bits block of plaintext. Similarly, within the decryption stage a 128-bits block of plaintext is created for the 128-bits block of ciphertext. It can be seen that CAST5 block cipher can perform the encryption and decryption process within shorter time than the others. Actually, when considered CAST5 takes the plaintext of 64-bits and the other algorithms take the plaintext of 128-bits, AES outperforms the CAST5 in terms of encryption and decryption. CAST5 is followed by AES, Camellia, and MARS block ciphers respectively in terms of execution time. Serpent has a very bad performance in terms of the encryption and decryption performance.

Throughput

Throughput is a parameter, which indicates the encrypted and decrypted bit amounts per second, and it allows us to compare the block ciphers fairly. The throughput of five block ciphers is shown in Figure 4.

AES, as shown in Figure 4, is the fastest block cipher

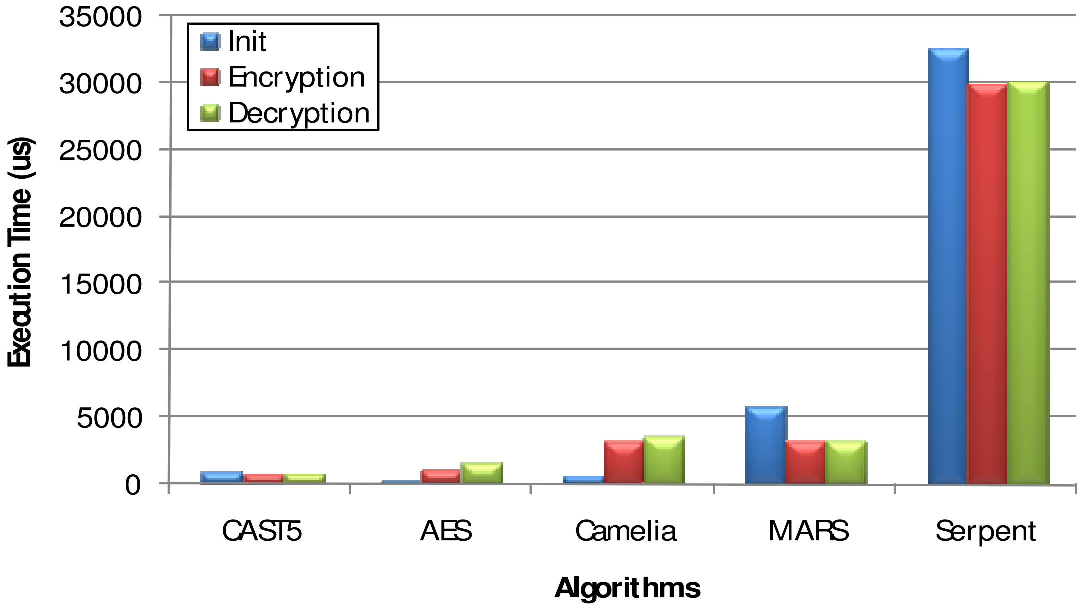

Figure 3. Execution time of the block ciphers.

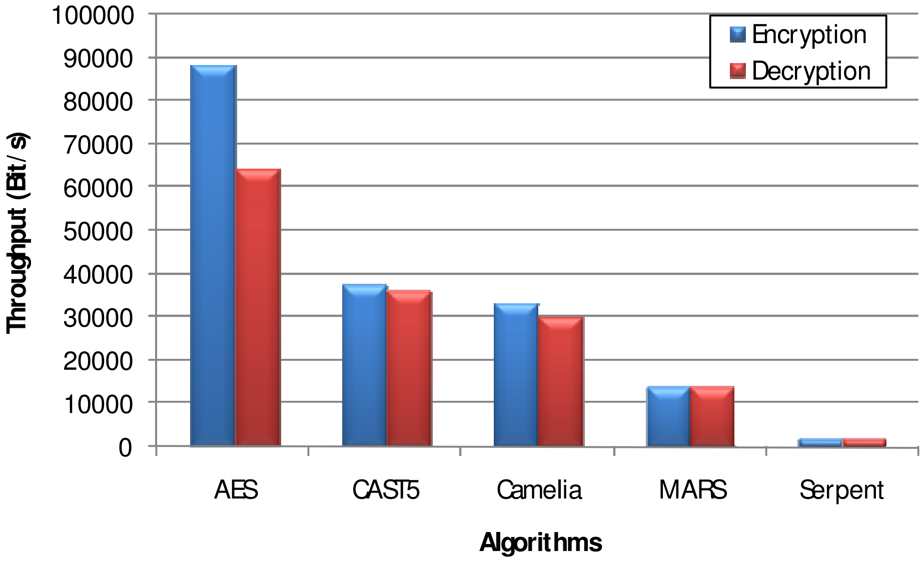

Figure 4. Throughput of the block ciphers.

among the selected five block ciphers. CAST5, Camellia, Mars and the Serpent algorithm follow AES cipher respectively, although the encryption and decryption time of the CAST5 are lower than AES as shown in Figure 3. The throughput of AES is higher than CAST5. The reason for this is that CAST5 is to take 64-bit block of plaintext and the other algorithms are to take 128-bit block of plaintext. In addition, the initialization stage of CAST5 is longer than the initialization stage of AES and Camellia. Therefore, the throughput of Camellia comes close to CAST5. Mars cipher provides about 4.5 – 6 times lower throughput compared to AES. Serpent has a very poor performance and it is slower than AES about 30 – 40 times.

CONCLUSIONS

In this study, we realized the C based software implementation of popular block ciphers such as AES, Serpent, Camellia, CAST5, and Mars and we have evaluated these algorithms on 8-bit microcontrollers. We compared the block ciphers with the memory requirements, execution time, and throughput criteria. According to the obtained results, it was observed that AES and Serpent are the most efficient algorithms and Mars is the most inefficient algorithm in terms of code and data memory usage. In term of execution time, CAST5 can perform the encryption/decryption procedures faster than the others. However, CAST5 takes 64-bit block of plaintext and AES take128-bit block of plaintext. AES outperforms CAST block cipher when this situation is considered. In term of throughput criteria, AES is the fastest algorithm among the chosen block ciphers. CAST5 and Camellia can be considered an alternative block cipher for AES. Although Serpent is the slowest algorithm, it is quite efficient in terms of memory usage. Therefore, it can be used on an application, which the speed is not important, but the memory size is limited. Mars has a very high memory requirement. Thus, it is not suitable for the microcontroller applications. As future works, these block cipher algorithms can be implemented with the help of assembly language to compare the implementation performance of C and assembly languages.

REFERENCES

Adams C (1997). The CAST-128 Encryption Algorithm, RFC 2144.

Anderson R, Biham E, Knudsen L (1998). Serpent: A Flexible Block Cipher with Maximum Assurance. In The First Advanced Encryption Standard Candidate Conference, Ventura, California, USA, 20–22, August.

Aoki K, Ichikawa T, Kanda M, Matsui M, Moriai S, Nakajima J, Tokita T (2001). Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms — Design and Analysis, Selected Areas in Cryptography, Volume 2012/2001.

ATMEGA128 (2010). 8-bit AVR microcontroller databook, Last Visited: July, Available: http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf.

Atmel Corporation (2010). AVR Studio 4, Last visited: July, Available: http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725.

AVR-GCC (2010). Free AVR C Compiler, Last visited: July, Available: http://sourceforge.net/projects/winavr/files/.

Biham E, Dunkelman O, Keller N (2002). Linear Cryptanalysis of Reduced Round Serpent. Fast Software Encryption, Volume 2355/2002, Lecture Notes in Computer Science, Springer Berlin / Heidelberg.

Biryukov A, Khovratovich D (2009). Related-key Cryptanalysis of the Full AES-192 and AES-256, in Advances in Cryptology – ASIACRYPT, 5912: 1-18.

Burwick C, Coppersmith D, D'Avignon E, Gennaro R, Halevi S, Jutla C, Matyas Jr. SM, O'Connor L, Peyravian M (1998). MARS -a candidate cipher for AES, AES submission.

Çakiroğlu M, Bayılmış C, Özcerit AT, Çetin Ö (2010). Performance Evaluation of Scalable Encryption Algorithm for Wireless Sensor Networks, Scientific Res. Essays, 5(9): 856-861, 4 May.

Carlisle MA (1997). Constructing Symmetric Ciphers Using the CAST Design Procedure Designs, Codes Cryptography, 12(3): 283-316, November, ISSN:0925-1022.

Daemen J, Rijmen V (2002). The Design of Rijndael: AES - The Advanced Encryption Standard, Springer-Verlag. 2002, ISBN 3-540-42580-2.

Eisenbarth T, Kumar S, Uhsadel L, Paar C, Poschmann A (2007). A Survey of Lightweight-Cryptography Implementations, IEEE Design & Test of Computers, Special Issue Secure ICs Secure Embedded Comput., 24(6): 522-533, November 2007.

Kelsey J, Schneier B (2000). MARS Attacks! Preliminary Cryptanalysis of Reduced-Round MARS Variants, the Third Advanced Encryption Standard Candidate Conference, 169-185, April 13-14, NY, USA.

Landau S (2000). Communications Security for the Twenty-first Century: The Advanced Encryption Standard, Notices of the AMS, 47(4).

Law Yee W, Doumen J, Hartel P (2006). Survey and benchmark of block ciphers for wireless sensor networks. ACM Trans. Sensor Networks, 2(1): 65-93.

Matsui M, Nakajima J, Moriai S (2004). A Description of the Camellia Encryption Algorithm, RFC 3713, April.

Meiser G, Eisenbarth T, Lemke-Rust K, Paar C (2008). Efficient Implementation of eSTREAM Ciphers on 8-bit AVR Microcontrollers, Industrial Embedded Systems, SIES 2008, p. 58-66.

Özcerit AT, Çakıroğlu M, Bayılmış C (2005). 8051 Mikrodenetleyici Uygulamaları, Papatya Yayınevi, Ekim, ISBN: 9756797649.

Rinne S, Eisenbarth T, Paar C (2007). Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers, SPEED.

Stallings W (2005). Cryptography and network security (4nd ed.) Principles and Practice, Prentice-Hall, ISBN-10: 0-13-187316-4.

Wang M, Wang X, Hu C (2009). New Linear Cryptanalytic Results of Reduced-Round of CAST-128 and CAST-256, Selected Areas in Cryptography, Volume 5381/2009, Lecture Notes in Computer Science, Springer Berlin / Heidelberg.