Multi-symmetric Ciphering System Module Based on Microcontroller

Ghada Abdelhady and Hussam Elbehiery

***Abstract*—It is widely recognized that data security will play a central role in the design of future IT systems. Many of those IT applications will be realized as embedded systems, which rely heavily on security mechanisms. Examples include security for wireless phones, wireless computing, and ATM security system in Banks. All modern security protocols use symmetric-key and public-key algorithms. This contribution presents a ciphering module that surveys several important cryptographic concepts and their relevance to embedded system applications. This study presents a module that includes three different symmetric ciphering algorithms: DES, DES-EC and AES based on microcontroller. Depending on the application and the data type, each user can select the suitable algorithm from the presented module. This study provides a comparison between the regular DES and the presented new DES. A comparison has been conducted for those encryption algorithms at different settings such as the key size, block size, speed, security, and the execution time.**

***Index Terms*—AES, DES, DES-EC, Elliptic Curves.**

I. Introduction

Without any doubt the first modern symmetric encryption algorithm was that contained in the Data Encryption Standard (DES) [1]. DES was published by the United States' National Bureau of Standards in January 1977 as an algorithm to be used for unclassified data. The algorithm had been used in banks for funds transfer security [2].

In this paper, we will try to build an integrated symmetric ciphering system that includes three different symmetric algorithms that are commonly used in our life. These algorithms are DES (Data Encryption Standard), New DES based on Elliptic Curve and AES (Advanced Encryption Standard). In the past, the first and the last one were implemented individually and the AES is commonly used these days in many applications. Although DES was broken by EFF (Electronic Frontier Foundation) in USA, we will implement its algorithm to set an integrated system that collect three important systems used in different applications that need different level of security. i.e. if we don't need a high security for the data, we can use the simplest and the fastest one which is DES. But if the application needs high secured data, of course, we will use AES or DES-EC.

Manuscript received March 09, 2012; revised April 01, 2012. This work was supported in part by The Center for Special Studies and Programs, Bibliotheca Alexandrina.

Ghada Abdelhady is with German University in Cairo, New Cairo city, Egypt.Phone: 0122-5871028;e-mail:ghada.abdelhady@guc.edu.eg.

Hussam Elbehiery is with Faculty of Engineering, Benha University, Egypt. Phone: 0111-6566262; e-mail:hussam.elbehiery@gmail.com.

This paper will present a practical benchmarking for all the previous encryption systems as an integrated module containing a keypad to select the suitable encryption algorithm to encrypt or decrypt the coming data via serial, USB, or Ethernet. This module also has four modes of operations: Self-test, Connection test, Operational mode and Development mode. This paper will explain an integrated symmetric ciphering module including new symmetric ciphering algorithm. This module is considered as a prototype. It is not expensive with much less memory compared with symmetric algorithms currently used. Also the module is specified for different data in different applications like military and intelligence field for the high secure data, big stores, stock markets, banks, legacy, etc.

The coming sections are structured as follows. Section 2 explains the standard of cryptography: DES and AES. Section 3 shows the new symmetric system: DES-EC and its stages. Then, in section 4, we discuss the specifications of symmetric cryptography on embedded systems. In addition, this section explores the brute force attack as a security analysis that has been performed on the proposed new algorithm to demonstrate how much this algorithm is a satisfactory security. The comparison between the presented symmetric systems will be shown also in section 4. The target platform for the introduced design will also be described in Section 4. The experimental results are presented in Section 5. Finally, our conclusions are in Section 6.

II. Standards of Cryptography: DES & AES

A. Regular DES

DES is a block cipher, meaning it operates on plaintext blocks of a given size (64-bits) and returns cipher-text blocks of the same size. Each block of 64 bits is divided into two blocks of 32 bits each, a left half block L and a right half R. The operation of the DES can be described in the stages of DES encryption algorithm shown in fig. 1.

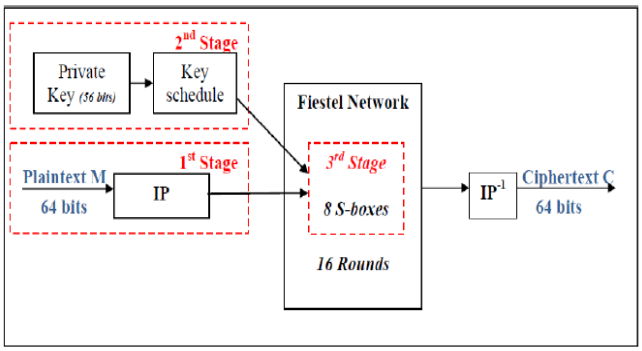

Fig 1. "DES" Encryption Stages

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

As shown in fig. 1, DES algorithm depends on three stages: Input Permutation (IP) stage, Key schedule stage and S-boxes stage. Since DES algorithm is symmetric, the encryption and the decryption algorithms share the structure presented in figure 4, but in opposite directions. The details of DES algorithm are explained in [5].

B. AES

AES is a symmetric algorithm and uses only one key of the three-encryption keys possible: 128 bits (16 bytes), 192 bits (24 bytes) or 256 bits (32 bytes).

AES algorithm converts a block of 128 bits to 128 bits of cipher text. The plaintext is converted in a 4x4 matrix, called "state". The initial encryption key is expanded into a table of 32-bit values. After that, the table is subdivided into groups of 32-bit values. The number of keys depends on the initial size of the key. A "round" is the base unit of transformation. The number of rounds depends on the key size: 10 for a key with a 128- bit length, 12 for a key with a 192-bit length and 14 for a key with a 256-bit length, respectively [7]. The details of AES algorithm are explained in [10].

Since our proposed algorithm is appeared recently in [3] and this paper will present its implementation on MCU, we need to take a strong symmetric system and commonly used nowadays like AES to be as a benchmarking for our new algorithm. Poettering in [11] has considered the target device class to be Atmel's AVR to implement AES.Atmel's AVR is a family of very fast and very powerful flash MCUs, operating at clock rates up to 16 MHz while executing one instruction per clock cycle (16 MIPS) and these specifications are similar to the used target device in this paper.

III. New Symmetric Ciphering System DES-EC

DES-EC also will depend on three stages: the plaintext masking stage, new key schedule stage and New S-boxes as shown in fig 2.

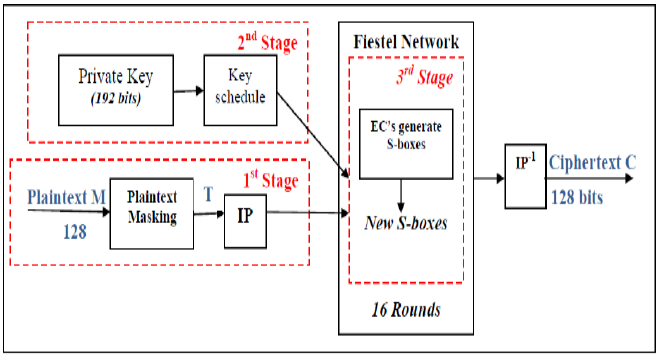

Fig 2. "DES-EC" Encryption Stages

We notice in fig 2 that each stage will be based on EC or groups of EC's. Since the new algorithm is symmetric like regular DES, the structure presented in fig. 2 is shared by the encryption and the decryption algorithms but in opposite directions. Meaning that the third stage represents the first stage after applying the inverse permutation and the first stage represents the last stage that includes the initial permutation and the inverse of the plaintext masking. The details of each stage are explained in [3].

IV. Symmetric Cryptography on Embedded Systems

A. Symmetric systems structure

The main components involved in cryptography are:

- Sender
- Receiver
- Plain text (the message before it is encrypted)
- Cipher text (the message that has been encrypted)
- Encryption and Decryption algorithms
- Key or keys.

In symmetric cryptography, the encryption and decryption algorithms are public but the key is secret. Only the key needs to be protected[12]. Both sender and receiver as shown in fig 3 share a common key.

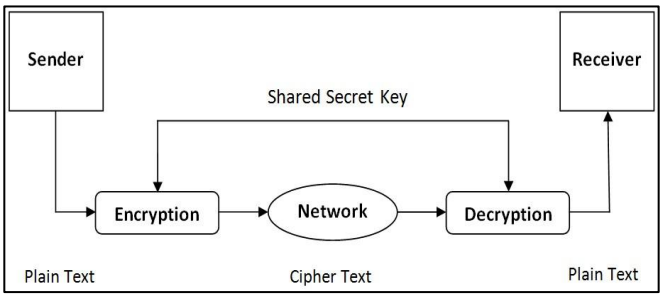

Fig 3. Symmetric-key Cryptography system

Particular specifications of a symmetric-key cryptography algorithm include the following:

Less time is needed to encrypt a message than when using a public key algorithm.

The key is usually smaller, so symmetric-key algorithms are used to encrypt and decrypt long messages [13].

B. Brute force attack of DES and AES vs. DES-EC

A brute-force attack involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. Table I shows how much time is involved for various key spaces [12].

Results are shown for three binary key sizes. The 56-bit key size is used with the DES (Data Encryption Standard) algorithm, the minimum key size specified for AES (Advanced Encryption Standard) is 128 bits and the 192- bit key size that is used in our suggested algorithm "NEW DES based on EC".

TABLE I. AVERAGE TIME REQUIRED FOR EXHAUSTIVE KEY SEARCH

Key size (bits)	No. of Alternative keys	Time required at 1 decryption/μs	Time required at 10^6 decryption/μs
56 (DES)	$2^{56}=7.2*10^{16}$	2^{55}μs=1142 years	10.01 hours
128 (AES)	$2^{128}=3.4*10^{38}$	2^{127}μs=$5.4*10^{24}$ years	$5.4*10^{18}$ years
192 (New DES)	$2^{192}=6.3*10^{57}$	2^{191}μs=10^{44} years	10^{38} years

For each key size, the results are shown assuming that it takes 1 *msec* to perform a single decryption, which is a reasonable order of magnitude for today's machines [12]. The security analysis of our new algorithm compared with AES is explained in details in [3].

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C. Embedded systems Specifications

Design and Efficiency

The central processing core in embedded systems is generally less complicated, making it easier to maintain. The limited function required of embedded systems allows them to be designed to most efficiently perform their functions [14].

Cost

The streamlined make-up of most embedded systems allows their parts to be smaller less expensive to produce.

Accessibility

Embedded systems are difficult to service because they are inside another machine, so a greater effort is made to carefully develop them. However, if something does go wrong with certain embedded systems they can be too inaccessible to repair. This concern is sometimes addressed in the design stage, such as by programming an embedded system so that it will not affect related systems negatively when malfunctioning.

Maintenance

Embedded systems are easier to maintain because the supplied power is embedded in the system and does not require remote maintenance.

D. Design Structure and Platform

The block diagram of our hardware module is presented in fig 4.

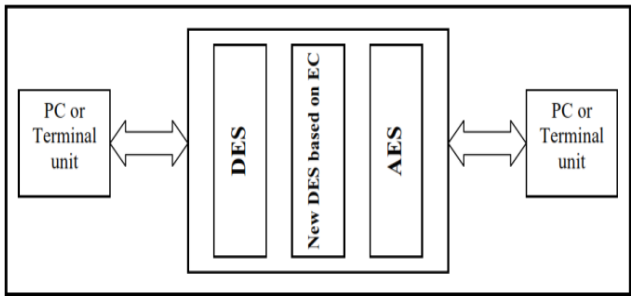

Fig 4. Block diagr am of Multi symmetr ic cipher ing system module

The use of microcontrollers and configurable processors has become an increasingly interesting option for embedded system development. Microcontrollers offer most of the features needed to implement even the most complex designs [15].

Our suggested embedded system is based on microcontroller that is considered the processor to carry out the required algorithms for encryption and decryption. Our embedded system has LCD to show all the processes implemented on the module also it contains a key pad for selecting the appropriate algorithm depending on the degree of the security of the incoming data that can be connected to the module via one of the three communication protocols presented here in our structure shown in fig 5. These communication protocols are USB, Parallel port RS232 or Ethernet RJ45.

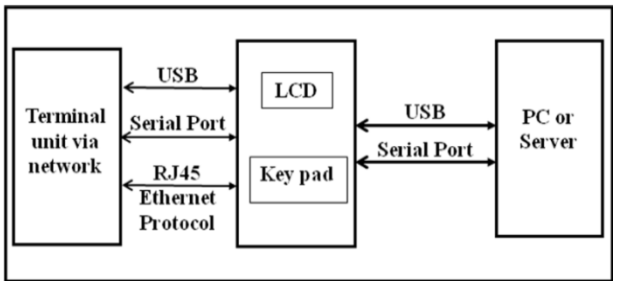

Fig 5. Structure of Multi symmetric ciphering system module

This paper has built the structure of multi symmetric module using the LV32MX v6™ development system shown in fig. 6 that provides a development environment for experimenting with PIC32™ microcontrollers from Microchip®. The system includes an on-board programmer PIC32Flash™ which serves as an interface between the microcontroller and a PC. You are simply expected to write a program, generate a .hex file and program your microcontroller using the mikroPROGSuite for PIC™ program. Numerous modules, such as TFT display with a resolution of 320x240, on-board 2x16 LCD, serial EEPROM module etc, are provided on the board and allow you to easily simulate the operation of the target device. The PIC32FLASH™program for programming provides a complete list of all supported microcontrollers.

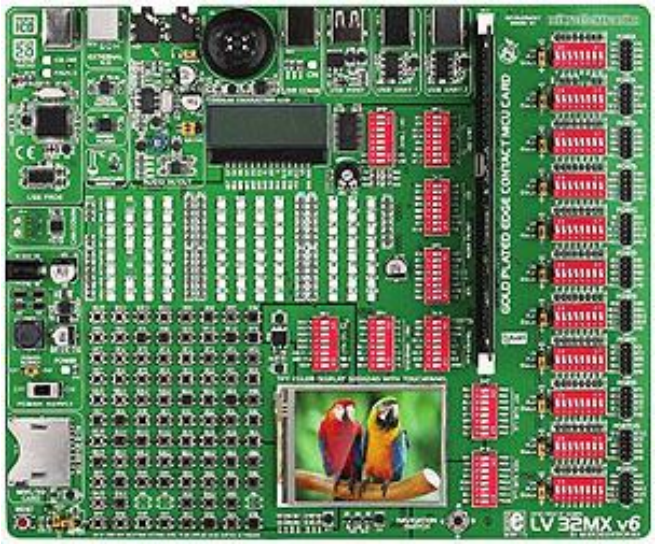

Fig 6. The LV32MX v6™ development system PCB structure

The schematic of LV32MX v6™ is shown in fig 7.

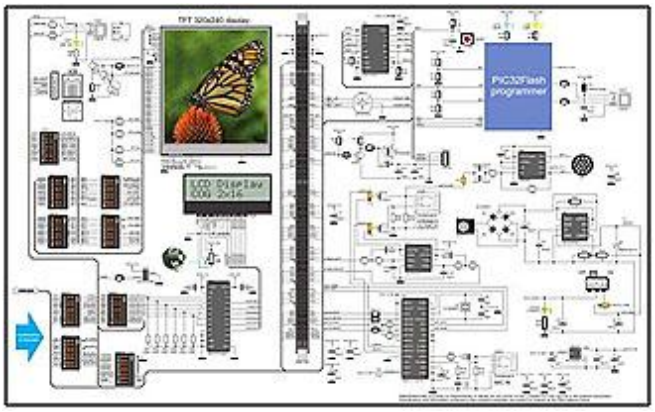

Fig 7. Schematic of the LV32MX v6 development system

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Such device has been designed carefully and due attention has been paid to choosing components thereof. This is why the most important parts of the system are manufactured in SMD (Surface Mounting Device) technology. Besides, the components are mounted by machines of the last generation using lead-free alloy for soldering. What we have now as a result is a high quality and reliable product that meets the world's highest standards being applied in electronics industry and environmental protection.

E. Modes of Operations

This module also has four modes of operations as in the following:

Self test mode

The module will be set in this mode using a specific code pressed on the key pad. In this mode, the module will be tested by connecting data to the input of the module and select one of the encryption algorithms on the module to encrypt the data then we will connect the output (the decrypted data) to the input again and select the corresponding decryption algorithm to get the incoming data again. By accomplishing this connection, we will guarantee that the selected system is working properly. Apply this test on each algorithm in the module.

Connection test mode

This mode is for testing on the connections between the module and the PC or any other terminal unit connected to our module. i.e. if the data coming from the PC or any other unit is connected to the module via one of the mentioned communication protocols like (RS232, USB and RJ45) as shown in Fig. 5, the type of connection will be appeared on the LCD presented on the module. This test is suggested to guarantee the connections between the module and the PC or any other unit, are good.

Operational mode

In this mode, the module delivers the incoming data from PC or any individual unit from the working network and according to the incoming code, the module will select the required system and select the encryption or decryption algorithm.

Development mode

In this mode, the algorithms on the module can be enhanced or modified using the USB cable that will be connected to the PC as shown in Fig 5. We can modify and update the algorithms via this mode.

V. Experimental Results

In this section, we are going to compare between the regular DES and DES-EC. The measurements that have been used for the comparison are the key size and the execution time of encryption/decryption schemes. The encryption time is considered the time that an encryption algorithm takes to produce a cipher text from a plaintext. Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total plaintext in byte encrypted divided by the encryption time [20]. For testing our module, the input size is 64 bits. Table II shows the comparison between DES-EC as a new symmetric ciphering techniques and DES according to the mentioned measurements.

Table II. DES vs. New DES based on EC

	DES	DES-EC	AES
Key size (bits)	56	192	192
Block size (bits)	*64*	*128*	*128*
Execution time (sec.)	Encryption 0.05	Encryption 0.12	Encryption 0.14
	Decryption 0.032	Decryption: 0.11	Decryption: 0.13
Speed (byte/sec)	Encryption 160	Encryption 133.33	Encryption 114.285
	Decryption 250	Decryption 145.45	Decryption 123.077

VI. Conclusion

Our main target in this paper is building a prototype that includes three different algorithms for encryption and can select the appropriate one to encrypt data according to the degree of its security level, also we could guarantee the security of the sent or received data. The module is carried out with lower cost than the modules that are currently used. From the experimental result, we can conclude that the new symmetric algorithm, "DES-EC", turned DES into secured algorithm that can be used in many applications. Also that will give strength to our module as a whole.

References

[1] "Cracking DES. Secrets of encryption research, wiretap politics and chip design", Electronic Frontier Foundation, May 1998.
[2] Ghada Abdelmouez M., Fathy S. Helail, and Abdellatif A. Elkouny, "New DES based on Elliptic Curves," World Academy of Science, Engineering and Technology, 63, March 2010.
[3] Ghada Abdelmouez M., Fathy S. Helail, and Abdellatif A. Elkouny, "Elliptic Curve Cryptographic based DES Reinforcement," ICGST-CNIR Journal, Volume 10, Issue 1, December 2010.
[4] Paar C, Jan Pelzl, Understanding Cryptography A Textbook for Students and Practitioners , Springer-Verlag Berlin Heidelberg, Germany, pp 70-71, 2010.
[5] "Data Encryption Standard (DES)", FIPS PUB 46-3 Federal Information Processing Standards Publications, U.S. Department of Commerce/National Institute of Standards and Technology, October 25, 1999.
[6] H. Darrel, A Menezes., S. Vanstone: Guide to Elliptic Curve Cryptography, Springer- Verlag New York, Inc., 2004.
[7] Paar C.: Implementation Options for Finite Field Arithmetic for Elliptic Curve Cryptosystems ECC, Worcester Polytechnic Institute, USA, 1999.
[8] E. Matthew, The Weierstrass Theory For Elliptic Functions, Department of Mathematics, MACS Herriot Watt University, Edinburgh, The Burn 2007.
[9] N. Torri, K. Yokoyama, Elliptic Curve Cryptosystems, Fujtsu Sci. Tech. J. pp. 140-146, December 2002.
[10] A. Stuart, Encryption and Security: the Advanced Encryption Standard, EDN, October, 2002.
[11] Poettering.AVRAES:TheAESblockcipheronAVRcontrollers,2006. http://point-at-infinity.org/avraes/
[12] S. Wiliam, Cryptography and Network Security Principles and Practices, Fourth Edition, Prentice Hall, November 16, 2005.
[13] Hugo Fruehauf , "Encryption Fundamentals," in Zyfers, October 2001.
[14] Thomas Wollinger, Jorge Guajardo, and Christof Paar, "Cryptography in Embedded Systems: An Overview," Embedded World 2003 Exhibition and Conference, pp. 735-744, Design & Elektronik, Nuernberg, Germany, February 18-20, 2003.
[15] Archana Ramachandran, Zhibin Zhou, and Dijiang Huang, Computing Cryptographic Algorithms in Portable and Embedded Devices, in Proceedings of the IEEE International Conference on Portable Information Devices (PORTABLE), pages 1-7, 2007.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[16] Chung-Chu Chia and Shuenn-Shyang Wang, "Efficient design of an embedded microcontroller for Advanced Encryption Standard," Proceeding of the 2005 Workshop on Consumer Electronics and Signal Processing (WCEsp 2005).
[17] Anton Kargl, Stefan Pyka and Hermann Seuschek, "Fast Arithmetic on ATmega128 for Elliptic Curve Cryptography," context of the SMEPP project (Secure Middleware for Embedded Peer-to-Peer Systems, FP6 IST-5-033563).
[18] Axel Sikora, "Implementing DES/3DES with Atmel FPSLIC," ATMEL Corporation, Security Application, University of Cooperative Education Loerrach, Germany, 2002. www.atmel.com
[19] D. S. Abdul Elminaam, H. M. Abdul Kader, M. M. Hadhoud, Performance Evaluation of Symmetric Encryption Algorithms, Communications of the IBIMA Volume 8, 2009 ISSN: 1943-7765.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)