Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 4 (3): 61-72, 2012
ISSN 1815-7432 / DOI: 10.3923/rjit.2012.61.72

Firmware for Data Security: A Review

Siva Janakiraman, Rengarajan Amirtharajan, K. Thenmozhi and John Bosco Balaguru Rayappan
Department of Electronics and Communication Engineering, School of Electrical and Electronics Engineering, Sastra University, Thanjavur, 613 401, India

Corresponding Author: Siva Janakiraman, Department of Electronics and Communication Engineering, School of Electrical and Electronics Engineering, Sastra University, Thanjavur, 613 401, India

ABSTRACT

The survey is made to find the various research articles that are published over the past 16 years in the area of security implementations with various embedded systems using three well known electronic data bases SCOPUS, IEEE Xplorer and Science Direct. More precisely, what it would cover? And to know the firmware development for microcontrollers and how to incorporate data security through cryptography is also discussed. The survey also takes into account the works that are carried out independently in specific type of crypto algorithm among the various types such as symmetric, asymmetric, block and stream cipher including the light weight algorithms. The major focus of the study lies in analyzing the feasibility in developing firmware for various cryptography algorithms on low, medium and high end microcontrollers that are well suitable for the development of wireless sensor nodes. Furthermore, this survey would explore various methods and tools used to analyze the important parameters like memory footprint and execution time and also suggest future recommendation for improving data security.

Key words: Symmetric key, public key, Block and stream ciphers, middleware and firmware, steganography

INTRODUCTION

Dedication makes the path to study precisely on the problem to bring out the result as required. This is not only true for men but also for machines. The evident statement can be obtained from the systems which encapsulate a special-purpose dedicated computer to do a set of defined tasks called embedded systems. The specialty of the embedded system is because of the highly specific requirements that it takes upon (Wang *et al*., 2007). Though, the task completion with perfection is an important design aspect of any embedded system it must also consider other design metrics such as low cost, size, weight, power consumption, portability, performance etc., (Tillich and Herbst, 2008). The achievable level of all design metrics during the embedded system design is mainly decided by the microcontroller chip, the main processing core of all embedded systems. Normally, the embedded systems with little memory and limited hardware resources are controlled by the code that resides on a non-volatile memory. The memory may be self contained in the microcontroller chip called on-chip memory or may reside outside the chip called off-chip memory. The set of instructions that controls the working of embedded systems called firmware.

Nowadays embedded systems are increasingly used in many distributive applications due to which the need for them to communicate among themselves and other peripherals has risen in the same magnitude. This communication essentially happens in an open media, where the data

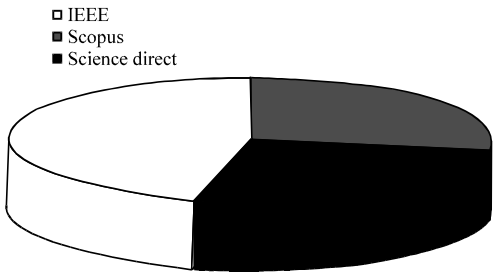

Fig. 1: Publications on e-databases

becomes vulnerable and hence, any private information communicated between any two may be known by others. The necessity to provide digital data security in open access channel invites the science of secret writing called cryptography (Jaberi *et al*., 2012; Al-Somani *et al*., 2006, 2009; Rabah, 2005a-c, 2006). In simple words any procedure that performs some mathematical operations between the data to be communicated (plain text) and a character string of length K-bits (key) to produce the result (cipher text) may be called as encryption and the reverse operation that brings back the plain text as decryption in the world of cryptography. The main goal of any cryptography algorithm(Schneier, 2007) is to provide confidentiality by ensuring none other than the intended recipient can read the message. In addition, the algorithm must posses other security requirements like authentication that facilitates to prove one's identity, integrity to ensure originality of the message and non-repudiation to identify the data origin (Liu *et al*., 2011).

To facilitate the embedded system firmware with the feature of data security, cryptography algorithm implementations are considered with the embedded core processing unit, microcontrollers. This study focuses on the implementations that have been carried out so far for the development of security firmware for embedded systems. To identify the studies that were done over the past 16 years on the secure firmware development, a survey is made on three of the popular electronic databases SCOPUS, IEEE Xplore and Science Direct. The distribution of research article publications on various international journals and conferences related to the topic of cryptography implementations with embedded systems are shown in Fig. 1.

Cryptography methods for data security: Three foremost types of cryptographic algorithms (Abomhara *et al*., 2010; Muda *et al*., 2010) specifically Secret Key Cryptography (SKC) with common key to both encrypt and decrypt, Public Key Cryptography (PKC) using distinct keys for encryption and decryption finally, hash functions for irrevocable data transformation are frequently in use for data security. The use of symmetric set of rules with single key to encipher and decipher the data brings the name Symmetric encryption for the SKC process. The greatest intricacy of this method is the distribution of symmetric key (Abomhara *et al*., 2010) between sender and recipient (Hromkovic, 2009; Muda *et al*., 2010). Stream and Block ciphers are the two sundry procedures of Secret key algorithms. Operations on stream of bits or on a byte or on the word length of a processing unit are carried in stream ciphers at a time whereas, a block of data can be handled at a time by the block cipher (Roman *et al*., 2007). The complication on PKC lies on the development of two mathematically interlinked keys that denies the determination of one key from another. The magical nature of the algorithm is the order in which the keys are used to encrypt and decrypt does not affect the result giving the meaning for asymmetric cryptography. The purpose of

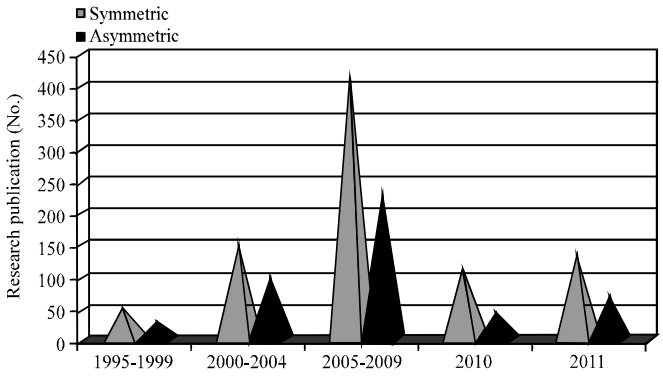

Fig. 2: Research publications on symmetric vs asymmetric ciphers in embedded

Symmetric cipher implementation is mainly to serve on integrity checks, user authentication and non-repudiation (Hromkovic, 2009; Salem *et al.*, 2011). Hash functions serve a different purpose than SKC and PKC schemes. The hash value called message digests, produced by the use of hash functions are used to dole out authenticity and integrity through digital finger print that hides the data and its length to be retrieved.

Among the security features rendered by crypto algorithms, our intention is to know only about the use of cryptography implementations on embedded processor cores for data security. As this is offered by symmetric and asymmetric procedures we limit our review in this study, SKC and PKC schemes alone (Leander *et al.*, 2007). The graph on Fig. 2 shows the results of the year wise survey made on electronic data bases with respect to the research publications on symmetric and asymmetric embedded crypto implementations.

The inference from the above graph makes it clear that the research publications on embedded implementations of cryptography are increasing every year. As can be seen explicitly, the number of publications on symmetric cryptography is nearly twice compared to the publications on asymmetric cryptography. Though, the features of microcontrollers make room for the design of even highly complex systems, there exists infeasibility on the implementation of public key cryptography on low-end embedded cores (Elbehiery and Abdelmouez, 2010). The accomplishment does not materialize due to their huge claim on memory and power (Ganesan *et al.*, 2003). But also, the computational density gives hike to demand on hardware and software resources (Leander *et al.*, 2007).

Symmetric ciphers on the other hand perform well on various microcontroller platforms. Another survey result shown in Fig. 3 manifestly tells that the study carried out on symmetric block cipher with microcontrollers during every year is more compared to the symmetric stream cipher implementations (Leander *et al.*, 2007).

For any embedded system the choice of the microcontroller depends on the kind of application and various design metrics that needs to be satisfied in the system design. The survey shows major number of publications on cryptography implementation with microcontrollers target two applications namely Radio Frequency Identification (RFID) tags and Wireless Sensor Nodes (WSN) (Elbehiery and Abdelmouez, 2010; Vogt *et al.*, 2009). The processing unit in most sensor node is a microcontroller, a highly constrained computer with memory and required interfaces that forms the base for simple applications. Depending on application needs other components like off-chip memory or cryptographic chips can also be there (Roman *et al.*, 2007). In the remaining chapters

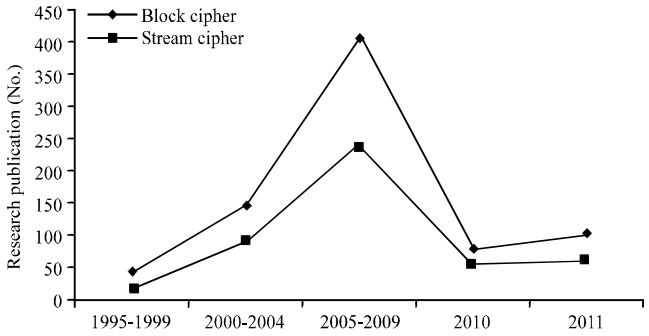

Fig. 3: Research publications on block vs stream ciphers in embedded

our paper makes a review on various cryptography algorithms implemented on diverse range of microcontrollers that are widely used for WSN.

Choice on CPU and crypto algorithms for wireless sensor nodes (WSN): The features offered by the microcontrollers such as cost effectiveness, low power consumption, moderate amount of computational capabilities and memory makes it as the best choice for the development of wireless sensor nodes (Cakiroglu *et al.*, 2010). Requirements such as peripheral interfaces, processing speed, code memory, data memory and also the kind of service to provide to the node determines the type of microcontroller to embed in a sensor node (Elbehiery and Abdelmouez, 2010). Many types of microcontrollers are produced by the same manufacturer to suite variety of applications and performance metrics. Out of them, very few are used by sensor node developers. This makes the nodes from diverse manufacturers to share the same microcontroller, though their architecture completely poles apart. As a result of this the performance of same algorithms on these nodes, will become akin (Roman *et al.*, 2007). Considering the choice of any cryptography algorithm to implement on microcontrollers for WSN the primary optimization goal is efficient implementation. Furthermore, the demands for low power consumption, minimal footprint; low RAM usage and execution time are also to be considered for tiny embedded systems (Tillich and Herbst, 2008). In addition, compute computational intensive operations such as Parings are extremely difficult to do with wireless sensor nodes (Ramachandran *et al.*, 2007).

Amongst the widely used microcontrollers for wireless sensor nodes, the Atmel ATmega128, Texas Instruments (TI) MSP430F14X/16X, Freescale 68HCS08, Microchip PIC18F6720, LPC2378 and Advanced RISC Machine (ARM) ARM920 are considered for most of the standard cryptography algorithm implementations. The distinct features of Advanced Virtual RISC (AVR) compared to other 8-bit microcontrollers are pre-decrement and post-increment of pointer registers that are done free of cost (without consuming additional cycles) also contribute heavily in optimized implementations (Bos *et al.*, 2009). MSP430 being the extreme low power consuming microcontroller also shows acceptable level of performance (Gura *et al.*, 2004).

These microcontrollers can be categorized into three different classes. Microcontrollers with 8-bit architectures belong to class one. Inspite of its limited processing abilities, class one (8-bit) processors are used nowadays since, still it holds a remarkable position in the market. The performance metrics of the class one processors are inferior to the two other classes. The 16-bit class two medium end processor architectures perform better than the lower end 8-bit class one processors but inferior compared to the higher end architectures. The third category is the 32-bit class three

architectures which are the best interms of performance (Hyncica *et al.*, 2011). Though, in general terms we categorize the performance in increasing order among low, medium and high end processors, the word length (data bus width) needed to perform the operations on the selected cryptography algorithm may reverse the above statement. For example, with its highly efficient RISC features, the low end class one ATmega processors may surpass all other architectures for small bus width (Ganesan *et al.*, 2003). Table 1 presented list of microcontrollers used in wireless sensor nodes.

Although, AVR microcontroller is an outstanding candidate for most of the algorithm implementations, the main problem is porting the code to very limited amount of SRAM. As an elucidation for reducing critical SRAM, moving s-boxes and large static data arrays into flash memory. The wise way to implement any data size greater than 8-bit with AVR is to convert all variables to standard integer sizes of AVR. When using low or medium end microcontroller the mismatch between wide bus width required to implement arithmetic for security (32 bit) and available bus width (8 or 16 bits) in addition to lack of convinced operations (e.g., multiply) in the Instruction Set Architecture (ISA) creates other challenges. Also, the presence of native features like variable sized bit wise shift, CISC vs RISC and memories like caches in ISA characteristics aids the performance of specific embedded cores (Ganesan *et al.*, 2003).

From the survey, we found the widely implemented and analyzed block based cryptography algorithm implementations on one or some of the microcontroller architecture categories mentioned above are AES (128/192/256) (Daemen and Rijmen, 2003), Twofish (128/192/256), DES, IDEA, Blowfish, RC2, RC5, RC6, XTEA (Wheeler and Needham, 1995), SEA (Standaert *et al.*, 2006), Sosemanuk. In general, apart from the demand on resources that makes asymmetric cryptography less appropriate for low-end microcontroller implementations also, the lifetime of nodes gets ruthlessly limited due to massive power consumption (Ganesan *et al.*, 2003).

In terms of SRAM consumption, SEA takes much less memory compared to AES and RC6 turns out to be the winner taking less memory size than SEA. Considering the aspect of performance, although AES does better than SEA, still SEA can perform better than RC6 taking less amount of execution time. Another feature that SEA offers is it supports better implementation feasibility on many type of processors with various word size (8-bit, 16-bit and 32-bit) (Cakiroglu *et al.*, 2010). Table 2 presented list of standard algorithms tested on various microcontroller platforms.

As proved by Elbehiery and Abdelmouez (2010), AES was a real performer compared to DES on implementation with the low end microcontroller ATmega32. Both in terms, of execution time as well as key space AES results were well ahead the DES (Fig. 4). On implementing IDEA and RC5 in ATmega128L, the encryption and decryptions timings were found to be identical for both algorithms. Performance of IDEA was well ahead the RC5. During the execution on multiple blocks

Table 1: List of microcontrollers used in wireless sensor nodes

Embedded controller	Word length (bit)	Operating frequency (MHz)	Internal code memory (kB)	Internal data memory (kB)
ATmega128L	8	16	128	8
ATmega32	8	16	32	2
68HCS08	8	40	64	4
PIC18F6720	8	20	128	4
MSP430F14X	16	4	60	2
MSP430F16X	16	8	55	5
LPC2378	32	78	512	58
ARM920	32	180	4000	512

Table 2: List of standard algorithms tested on various microcontroller platforms

Algorithm	Block size (bits)	Key length (bits)
AES	128	128
		192
		256
DES	64	64
RC5	65	128
RC6	128	128
		192
		256
IDEA	64	128

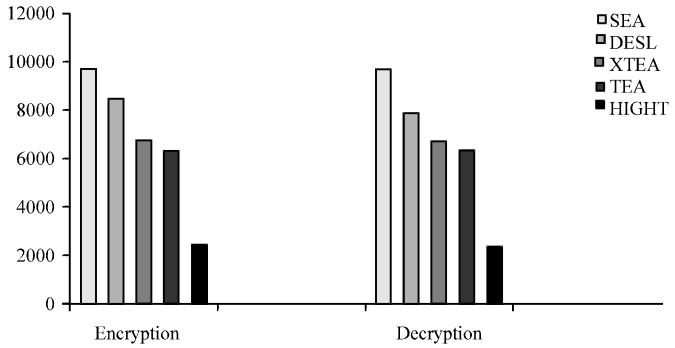

Fig. 4: Execution time (clock cycles)

of plaintext, the variation in performance of IDEA was highly acceptable compared to RC5 since, the execution time was not a multiple of the number of blocks encrypted (Choi and Song, 2006).

The instruction set extensions to AVR proposed by Tillich and Herbst (2008) to enhance the speed and code occupancy for implementing AES-128 which could not be achieved even by highly optimized assembly code with native instructions alone.

Steps were taken to enhance the speed of the AES-128 implemented on 8-bit AVR using the optimization techniques. Here, programming was done using Compute Unified Device Architecture (CUDA) which is an extension to the C language that facilitates parallel programming through Single Instruction Multiple Thread (SIMT) (described in NVIDIA CUDA-Programming Guide). CUDA allows the programmers to define kernel functions and thereby, supports SIMT. Two versions of implementations namely fast and compact were tried. The compact version took less area in RAM whereas, the fast version made a trade-off with RAM usage by consuming RAM area for the storage of S-box (Bos *et al.*, 2009).

LIGHT WEIGHT CRYPTOGRAPHY

Light weight cryptography is emerging as a counterpart for the well known statement that says about the infeasibility in the implementation of asymmetric cryptography with low end microcontrollers. It offers an attractive asymmetric-key cryptosystem for use with resource constrained devices, specifically when sufficient cryptographic operations cannot be guaranteed to provide ample security (Tillich and Herbst, 2008). Trade-offs plays a vital role in the design of light weight cryptography to optimize design metrics of embedded systems namely security, cost and performance. Though, the compromise is possible, always any two metrics security and cost or

security and performance, or cost and performance; may contradict each other. Optimizing all the three at once is physically impossible. Pipelined hardware implementation can provide high-performance consequently increasing the cost, through high area requirement. Utilizing the bit permutations that are virtually free in hardware can result in a low-cost design. As a metric for evaluation on firmware implementations, code and data memory requirements and the clock cycles taken for execution are taken. Another important metric, the power consumption on firmware implementations can be arrived as a rough estimate by multiplying the processing cycles by the average power consumption of the target device.

Two ciphers namely the Tiny Encryption Algorithm (TEA) (Wheeler and Needham, 1995) family and the International Data Encryption Algorithm (IDEA), which consists only of arithmetic operations is the ones that are especially optimized for software architectures. IDEA uses 16-bit words and TEA uses 32-bit words. Simple operations like addition, XOR addition and multiplication are present in IDEA whereas, TEA family needs shifts along with addition and XOR addition. Both algorithms can perform well on low end 8-bit platforms. They both do not need huge memory since there is no substitution box (S-box), present in them (Leander *et al*., 2007).

The renowned asymmetric algorithm RSA (Mousa, 2005), which is known to be suitable for signing and well as encryption, is widely used on numerous applications. It is due high demand on resources for its operation, it is considered as an expensive candidate for resource-constrained devices. This is the reason for the development of new algorithms with more optimizations, such as Elliptic Curve Cryptography (ECC) (Gura *et al*., 2004; Rabah, 2005a; Al-Somani *et al*., 2006; Olorunfemi *et al*., 2007; Roman *et al*., 2007; Al-Somani *et al*., 2009). The ECC is a great opponent in the field of light weight cryptography algorithms. In comparison with the traditional public key cryptosystems like RSA, offers superior speed, less power consumption, memory, bandwidth and smaller key sizes without degradation in security (Tillich and Herbst, 2008). Also as a matter of implementing ECC on microcontrollers with small word length, its performance raises over RSA (Gura *et al*., 2004). When accelerated by appropriate cryptographic hardware though it is an asymmetric implementation ECC on embedded may seem to perform slightly better but still may take excessive time compared to standard symmetric implementations such as Advanced Encryption Standard (AES) performance (Leander *et al*., 2007; Muda *et al*., 2010; Zaidan *et al*., 2010a). Another cipher designed for embedded applications with low cost microcontrollers is Scalable Encryption Algorithm (SEA), belongs to symmetric block cipher approach. Implementation of this can be made with less memory size, limited instruction set and small footprint. To design, is quite simple as it can be performed using simple bit manipulation operations such as XOR, bit/word rotations, modulo addition and s-box (Cakiroglu *et al*., 2010).

Some of the well known light-weight ciphers like Data Encryption Standard (http://csrc. nist. gov/publications/fips/fips46-3/fips46-3. pdf) Light Weight Extension (DESL) (Leander *et al*., 2007), HIGHT (Hong *et al*., 2006), SEA, Tiny Encryption Algorithm (TEA) and Extension to TEA (XTEA) were implemented on AVR ATmega128 and analyzed (Table 3).

Table 3: Specification of light weight algorithms

Cipher	DESL	HIGHT	SEA	TEA	XTEA
Block length	64	64	96	64	64
Key length	56	56	96	128	128
Rounds	16	16	141	32	64

As the implementation results depicts, optimization on the highly constrained resource (memory) is achievable with light-weight cryptography but, the algorithms fails to claim in terms of throughput (Rinne *et al.*, 2007).

Though called as weak, the extremely constrained 4-bit microcontrollers are still being used much in today's applications. It was proven that, even a 4-bit microcontroller can also be an optimal target for the implementation of ultra-lightweight block cipher. To prove this the LWC PRESENT was implemented on ATAM893D microcontroller of ATMEL's MARC4 family. In order to keep the execution time of a block to be 200 ms, the operating frequency is limited to 500 KHz though 2 MHz is the maximum that can be handled by the device (Vogt *et al.*, 2009). On comparison with PRESENT, Hummingbird gives 4-7 times improved performance when speed-optimized and 147 times for size-optimized implementations on similar platforms (Engels *et al.*, 2010).

For the implementation of symmetric ciphers, the use of a non-conventional method called Bit-slicing can be used with an n-bit processor, where it is considered as a collection of n 1-bit execution units operating in SIMD mode. Light-weight Instruction Set Extensions (ISEs) focused on implementation with bit-slicing can be used with any of the above said algorithm to reduce the footprint on code memory, to improve the performance and also which in turn makes the execution time predictable (Grabher *et al.*, 2008).

While, there are several implementation proofs exists, showing the feasibility to create firmware for data security, there exists few exceptions like PIC12F675 microcontroller, the one used in the design of WSN. Being in the category of weak this core requires a separate cryptographic chip so as to provide a nominal protection for data (Roman *et al.*, 2007).

Methods and tools for analysis: To enable the calculation and execution of time in terms of number of clock cycles for the total number of instructions in the firmware program often, we convert the code in high level language like 'C' to assembly (Elbehiery and Abdelmouez, 2010). To analyze the speed of the algorithm on any platform, the use of maximal available operating clock frequency of the core to be used. This may not result in optimal energy consumption but clearly defines the upper bound for performance at each individual platform. Footprint occupied by LWC are presented in Fig. 5. Information on footprint can be the size of the implemented cipher algorithm reported by the linker. The presence of auxiliary functions overhead from library, a varying factor for various algorithms, aids for rise in code size and makes the code size evaluation unfair. As far as block ciphers are concerned, the time taken to encrypt the complete block can be taken. The frequently used method to measure the time is to run the internal timer and to exclude

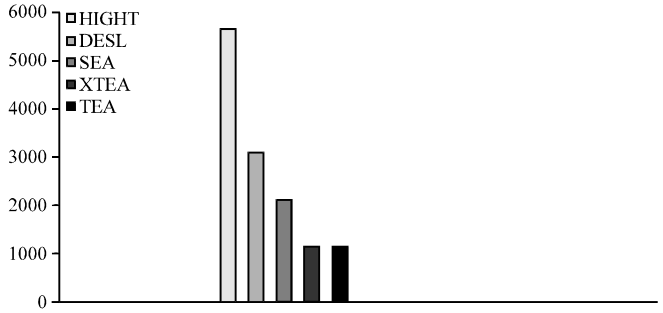

Fig. 5: Footprint occupied by LWC

the overhead produced by interrupt. The easiness of this way makes it to suits well for various platforms (Hyncica *et al.*, 2011). Also, the time to load the plain text from a source in to the memory of the microcontroller and the time taken at the end of encryption to store the cipher text in to memory may get exempted while calculating the execution time of any block cipher in embedded core (Tillich and Herbst, 2008).

Open source tools which are available free of cost are the better choice as it also provides guidelines for operations, addresses problems and updates from the open source community. In this regard AVR provides the well known Integrated Development Environment (IDE) called AVR Studio software platform accompanied with a popular c compiler for AVR micro-controllers, AVR GCC that can be used to evaluate the performance of the cryptography algorithms (Cakiroglu *et al.*, 2010). Another, powerful tool for AVR with enriched features is MikroC PRO development tool. Inclusion of libraries can be done in this to dramatically raise the execution speed. The HEX generated by the MikroC PRO for AVR is compatible with all programmers. The integrated Software Simulator gives us the way to inspect program flow by debugging the executable logic (Hyncica *et al.*, 2011).

The CrossWorks studio IDE for MSP430 comprises of ANSI C compiler, macro assembler, linker/locator, libraries, core simulator, flash downloader, JTAG debugger. Optimizations on code for the smallest size or fastest speed can also be set through this (Engels *et al.*, 2010). IAR Embedded Work Bench offers a sophisticated IDE with debugger. Even, the entire microprocessor designs can also be simulated through PROTEUS, a complete electronics design system, with capabilities to run actual processor machine code in real-time (Elbehiery and Abdelmouez, 2010).

Recommendations: Cryptography and steganography are the well known twins in the world of security. To incorporate security as a firmware on resource constrained devices, the term cryptography is well analyzed by many researchers with low, medium and high end microcontrollers. Due to its high demand on memory for the storage of cover such as image, audio or video; Steganography is considered rarely to provide security using microcontrollers. Microcontrollers are the best candidate to do bit and byte level operations that are needed for the steganography methods like LSB substitution used in spatial domain. Image steganography on spatial domain was suggested by Stanescu *et al.* (2009) with NXP Phillips LPC-2294, an ARM 7 based high end microcontroller unit. They were able to implement simple LSB substitution (Amirtharajan and Balaguru, 2009, 2011; Rajagopalan *et al.*, 2011; Padmaa *et al.*, 2011) on a various sizes of RGB images that can be accommodated in the external memory of 1 MB mapped with LPC 2294. This implementation suggested a new dimension to provide data security in mobiles devices without significant rise in cost. As improvement in execution time which is a desirable factor for all, eliminating the need for external memory, the major factor that limits the performance is our primary recommendation for future study. Numerous steganography algorithms (Amirtharajan and Balaguru, 2009; Rajagopalan *et al.*, 2011; Padmaa *et al.*, 2011; Zaidan *et al.*, 2010b; Amirtharajan *et al.*, 2011; Amirtharajan and Rayappan 2012a,b) on spatial and on frequency domains (Thanikaiselvan *et al.*, 2011) has also been tried with medium and low end microcontrollers. This challenging endeavor will figure out a wise way to establish greater security in wireless sensor nodes without substantial rise in cost.

CONCLUSION

The survey clearly shows that the study on firmware development for embedded system facilitating data security is increasing dramatically in recent years. Although, all kinds of cryptography algorithms were considered by various researchers on a wide spread only block based symmetric ciphers were the major choice for implementation with microcontrollers. Inspite of its higher security, the higher demand on the constrained resources of microcontrollers raised by the asymmetric algorithms made them as least choice in most applications. The article summarizes the different kind of standard and light weight algorithms implemented on various classes of microcontrollers appropriate for wireless sensor nodes. After analyzing the feasibility on security implementations with microcontrollers through cryptography, this paper proposes possible recommendations on using image steganography with microcontrollers for images sensor nodes.

REFERENCES

Abomhara, M., O. O. Khalifa, O. Zakaria, A. A. Zaidan, B. B. Zaidan and H. O. Alanazi, 2010. Suitability of using symmetric key to secure multimedia data: An overview. J. Applied Sci., 10: 1656-1661.

Al-Somani, T. F., M. K. Ibrahim and A. Gutub, 2006. High performance elliptic curve GF (2^m) crypto-processor. Inform. Technol. J., 5: 742-748.

Al-Somani, T. F., E. A. Khan, A. M. Qamar-ul-Islam and H. Houssain, 2009. Hardware/software co-design implementations of elliptic curve cryptosystems. Inform. Technol. J., 8: 403-410.

Amirtharajan, R. and R.J.B. Balaguru, 2009. Tri-layer stego for enhanced security-a keyless random approach. Proceedings of the IEEE International Conference on Internet Multimedia Services Architecture and Applications, December 9-11, 2009, Bangalore, India, pp: 1-6.

Amirtharajan, R. and R.J.B. Balaguru, 2011. Data embedding system. WIPO Patent Application WO/2011/114196. http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=WO2011114196&F=0

Amirtharajan, R., R.R. Subrahmanyam, P.J.S. Prabhakar, R. Kavitha and J.B.B. Rayappan, 2011. MSB over hides LSB: A dark communication with integrity. Proceedings of the 2011 IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA), December 12-14, 2011, Bangalore, Karnataka, India.

Amirtharajan, R. and J.B.B. Rayappan, 2012a. An intelligent chaotic embedding approach to enhance stego-image quality. Inform. Sci., (In Press). 10.1016/j.ins.2012.01.010

Amirtharajan, R. and J.B.B. Rayappan, 2012b. Inverted pattern in inverted time domain for icon steganography. Inform. Technol. J., (In Press).

Bos, J. W., D. A. Osvik and D. Stefan, 2009. Fast implementations of AES on various platforms. Cryptology ePrint Archive, Report 2009/501 (2009), http://eprint. iacr. org/2009/501. pdf.

Cakiroglu, M.,C. Bayilmis, A. T. Ozcerit and O. Cetin, 2010. Performance evaluation of scalable encryption algorithm for wireless sensor networks. Sci. Res. Essays, 5: 856-861.

Choi, K. J. and J. I. Song, 2006. Investigation of feasible cryptographic algorithms for wireless sensor network. Proceedings of the 8th International Conference on Advanced Communication Technology, February 20-22, 2006, Phoenix Park, pp: 1379-1381.

Daemen, J. and V. Rijmen, 2003. The Design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag, Berlin, ISBN-13: 978-3540425809.

Elbehiery, H. M. and M. G. Abdelmouez, 2010. Implementation of new symmetric ciphering on ATMEGA 32. Proceedings of the International Computer Engineering Conference, December 27-28, 2010, Giza, pp: 1-8.

Engels, D., X. Fan, G. Gong, H. Hu and E. M. Smith, 2010. Hummingbird: Ultra-lightweight cryptography for resource-constrained devices. Fin. Cryptogr. Data Security, LNCS, 6054: 3-18.

Ganesan, P., R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller and M. Sichitiu, 2003. Analyzing and modeling encryption overhead for sensor network nodes. Proceedings of the 2nd Workshop on Wireless Sensor Networks and Applications, September 19, 2003, San Diego, CA, USA, pp: 151-159.

Grabher, P., J. Grobschadl and D. Page, 2008. Light-weight instruction set extensions for bit-sliced cryptography. Proceedings of the 10th International Workshop, August 10-13, 2008, Washington, DC., USA.

Gura, N., A. Patel, A. Wander, H. Eberle and C. S. Sheueling, 2004. Comparing elliptic curve cryptography and RSA on 8-bit cpus. Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems, LNCS., 3156, August 2004, Springer, Berlin, Heidelberg, pp: 119-132.

Hong, D., J. Sung, S. Hong, J. Lim and S. Lee *et al.*, 2006. HIGHT: A new block cipher suitable for low-resource device. Cryptogr. Hardware Embedded Syst., LNCS, 4249: 46-59.

Hromkovic, J., 2009. Algorithmic Adventures: From Knowledge to Magic. Springer, Berlin, Heidelberg, ISBN: 9783540859857, pp: 239-276.

Hyncica, O., P. Kucera, P. Honzik and P. Fiedler, 2011. Performance evaluation of symmetric cryptography in embedded systems. Proceedings of the 2011 IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), September 15-17, 2011, Prague, pp: 277-282.

Jaberi, A., R. Ayanzadeh and A. S. Z. Mousavi, 2012. Two-layer cellular automata based cryptography. Trends in Applied Sci. Res., 785: 68-77.

Leander, G., C. Paar, A. Poschmann and K. Schramm, 2007. New lightweight DES variants. Proc. Int. Workshop Fast Software Encrypt., 4593: 196-210.

Liu, Y. Zhou, Y. Xiao and G. Sun, 2011. Encryption algorithm of RSH (round sheep hash) chaoqun. Inform. Technol. J., 10: 686-690.

Mousa, A., 2005. Sensitivity of changing the RSA parameters on the complexity and performance of the algorithm. J. Applied Sci., 5: 60-63.

Muda, Z., R. Mahmod and M. R. Sulong, 2010. Key transformation approach for rijndael security. Inform. Technol. J., 9: 290-297.

Olorunfemi, T. O. S., B. K. Alese, S. O. Falaki and O. Fajuyigbe, 2007. Implementation of elliptic curve digital signature algorithms. J. Software Eng., 1: 1-12.

Padmaa, M., Y. Venkataramani and R. Amirtharajan, 2011. Stego on 2^n:1 platform for users and embedding. Inform. Technol. J., 10: 1896-1907.

Rabah, K., 2005a. Theory and implementation of data encryption standard: A review. Inform. Technol. J., 4: 307-325.

Rabah, K., 2005b. Secure implementation of message digest, authentication and digital signature. Inform. Technol. J., 4: 204-221.

Rabah, K., 2005c. Theory and implementation of elliptic curve cryptography. J. Applied Sci., 5: 604-633.

Rabah, K., 2006. Implementing secure RSA cryptosystems using your own cryptographic JCE provider. J. Applied Sci., 6: 482-510.

Rajagopalan, S., S. Janakiraman, H. N. Upadhyay and K. Thenmozhi, 2011. Hide and seek in silicon: Performance analysis of quad block equisum hardware steganographic systems. Proceedings of the International Conference on Communication, Technology and System Design, December 7-9, 2011, Coimbatore, Tamilnadu, India.

Ramachandran, A., Z. Zhou and D. Huang, 2007. Computing cryptographic algorithms in portable and embedded devices. Proceedings of the IEEE International Conference on Portable Information Devices, May 25-29, 2007, Orlando, FL, pp: 1-7.

Rinne, S., T. Eisenbarth and C. Paar, 2007. Performance analysis of contemporary light-weight block ciphers on 8-bit microcontrollers. SPEED 2007, http://www. ei. rub. de/media/crypto/veroeffentlichungen/2011/01/29/lw_speed2007. pdf.

Roman, R., C. Alcaraz and J. Lopez, 2007. A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. Mobile Networks Appl., 12: 231-244.

Salem, Y., M. Abomhara, O. O. Khalifa, A. A. Zaidan and B. B. Zaidan, 2011. A review on multimedia communications cryptography. Res. J. Inform. Technol., 3: 146-152.

Schneier, B., 2007. Applied Cryptography: Protocols, Algorithm and Source Code in C. 2nd Edn., Wiley, India.

Standaert, F. X., G. Piret, N. Gershenfeld, J. J. Quisquater, 2006. SEA: A scalable encryption algorithm for small embedded applications. Proc. Int. Conf. Smart Card Res. Adv. Appl., 3928: 222-236.

Stanescu, D., V. Stangaciu, I. Ghergulescu and M. Stratulat, 2009. Steganography on embedded devices. Proceedings of the 5th International Symposium on Applied Computational Intelligence and Informatics, May 28-29, 2009, Timisoara, pp: 313-318.

Thanikaiselvan, V., P. Arulmozhivarman, R. Amirtharajan and J. B. B. Rayappan, 2011. Wave (let) decide choosy pixel embedding for stego. Proceedings of the International Conference on Computer, Communication and Electrical Technology, March 18-19, India, pp: 157-162.

Tillich, S. and C. Herbst, 2008. Boosting AES performance on a tiny processor core. Topics Cryptol., LNCS, 4964: 170-186.

Vogt, M., A. Poschmann and C. Paar, 2009. Cryptography is feasible on 4-Bit microcontrollers: A proof of concept. Proceedings of the IEEE International Conference on RFID, April 27-28, 2009, Orlando, FL, pp: 241-248.

Wang, L., H. Zhao and G. Bai, 2007. A cost-efficient implementation of public-key cryptography on embedded systems. Proceedings of the 2007 International Workshop on Electron Devices and Semiconductor Technology, June 3-4, 2007, Tsinghua University, pp: 194-197.

Wheeler, D. J. and R. M. Needham, 1995. TEA: A tiny encryption algorithm. Fast Software Encrypt., 1008: 363-366.

Zaidan, A. A., B. B. Zaidan, A. K. Al-Frajat and H. A. Jalab, 2010a. An overview: Theoretical and mathematical perspectives for advance encryption standard/rijndael. J. Applied Sci., 10: 2161-2167.

Zaidan, B. B., A. A. Zaidan, A. K. Al-Frajat and H. A. Jalab, 2010b. On the differences between hiding information and cryptography techniques: An overview. J. Applied Sci., 10: 1650-1655.