EVALUATING GALOIS COUNTER MODE IN LINK LAYER SECURITY ARCHITECTURE FOR WIRELESS SENSOR NETWORKS

Vivaksha Jariwala[1] and Dr. D. C. Jinwala[2]

[1]Department of Computer Engineering, C. K. Pithawalla College of Engineering and Technology, Surat, India
vivakshajariwala@gmail.com
[2]Department of Computer Engineering, S. V. National Institute of Technology, Surat, India
dcjinwala@gmail.com

ABSTRACT

Due to the severe resource constraints in the Wireless Sensor Networks (WSNs), the security protocols therein, should be designed to optimize the performance maximally. On the other hand a block cipher and the mode of operation in which it operates, play a vital role in determining the overall efficiency of a security protocol. In addition, when an application demands confidentiality and message integrity, the overall efficiency of a security protocol can be improved by using the Authenticated Encryption (AE) block cipher mode of operation as compared to the conventional sequential encryption and authentication. Amongst the AE block cipher modes, the Galois Counter mode (GCM) is the latest recommended AE mode by the NIST. In this paper, we attempt at evaluating the performance of the GCM mode in the link layer security protocol for a WSN viz. TinySec and compare it with the default conventional block cipher modes of operation used therein. To the best of our knowledge ours is the first experimental evaluation of Galois Counter Mode with Advanced Encryption Standard Cipher at the link layer security architecture for WSNs.

KEYWORDS

Wireless Sensor Networks, Link Layer Security, Block Cipher, Encryption, Authentication, Galois Counter Mode.

1. INTRODUCTION

The Wireless Sensor Networks [1] (WSNs) are characterized by severe constraints in computational, storage and energy resources. In addition, due to the wireless communication and the deployment in ubiquitous environment, ensuring the communication and physical security in WSNs is non-trivial. [2]. Further due to the in-network processing and the subsequent data-centric multihop communication, apart from the end-to-end security protocols at the application layer (SSH-SSL [3], IPsec[4]), link layer security is also necessary. The link layer encryption-decryption further increases the security overhead due to multiple invocations of the security related operations. Therefore, it is necessary that the link layer security protocols in WSNs are carefully tuned to achieve minimal overhead while giving the optimum performance.

Now, in any secure communication, there is a need for considering two security goals minimally viz. *confidentiality* and *integrity*. For the WSNs deployed in ubiquitous environment for *sensing, processing* and *communication*, the security attributes desired are only *message integrity* or *confidentiality* and *message integrity* [5]. There is compelling evidence that support for confidentiality alone without authentication is meaningless [6]. Hence, the security protocol in WSN may support either confidentiality or confidentiality and message integrity.

DOI : 10.5121/ijnsa.2010.2405

The conventional approach to support confidentiality with message authentication is based on *composition* i.e. sequential operations of encryption and message authentication. Such approach increases the eventual block cipher calls as a separate call to the block cipher is required for message authentication after encryption [7].

However, the alternative approach is to use integrated encryption as well authentication operations - known as Authenticated Encryption (AE) [8]. The common AE modes of interest are the Offset Codebook Mode (OCB) [9] and the Counter with Cipher Block Chaining (CCM) [10]. However, the latest addition to these is the Galois Counter Mode [11], now recommended by NIST as a standard AE mode of operation [12].

On the other hand, the first link layer security architecture fully developed and popularly experimented and researched is TinySec [13]. TinySec uses the CBC [14] block cipher mode for confidentiality as well as CBC-MAC [15] for message integrity and authentication. Thus, for those applications demanding confidentiality as well as message authentication, the composition based approach is to be employed in TinySec i.e. first encrypt the message and then attach a MAC on the sender side and compute & verify the MAC and then decrypt the valid packet in the receiver side. We believe that the resultant overhead can be improved if the AE block cipher modes are employed. Hence, in this paper we implement the GCM mode for the TinySec and experimentally evaluate the performance of the GCM mode against the TinySec defaults of composition of CBC-mode and CBC-MAC.

Our results clearly signify that the GCM mode is the preferred mode for link layer security protocols when the underlying applications demand confidentiality as well as message authentication. To the best of our knowledge ours is the first attempt in implementing and benchmarking the storage requirements of GCM mode with AES cipher, in the link layer security framework in WSNs.

The remainder of this paper is organized as follows. In section 2 we introduce block cipher modes and the AE modes as well as survey the related work in the area. In Section 3, we describe our methodology of implementation and experimental setup. In section 4, we present and analyze the performance results. Finally, in section 5 we end with the conclusion and the scope for future work.

2. THEORETICAL BACKGROUND

2.1. Block Cipher Mode of Operation

Authenticated Encryption Systems are those which perform authentication and confidentiality simultaneously. Till now, it was not that much popular. People were using confidentiality and authentication one after another that is simply known as generic composition. But recently there have been a number of new construction which achieve this two goals simultaneously, often much faster than generic composition solutions.

In generic composition we need to invoke block cipher 2*m times where m is the number of block in which plain text is divided.

In Authenticated Encryption [6] total two ways are there. Single pass combined mode and two pass combined mode. In single pass combined mode we need to invoke block cipher m+log(m) times. Where m is used for encryption and during encryption it will produce some seed value which can used for authentication that requires only log(m) invocation of block cipher. The first single passed combined mode introduced is IAPM [16]. OCB [9] is also one of the single pass authenticated encryption mode. But after that all researchers who are working on this made their work patented. Now the interest of researchers has moved in another direction which is the two-pass combined mode. The two-pass combined modes represent a class of algorithms with performances not so far from the single-pass ones, but all with no intellectual property restriction.

Galois Counter Mode [11] is two-pass combined mode of authenticated encryption mode, which we have evaluated here with AES-Cipher for link layer security architecture in WSNs.

2.2. Related Work

In this section we discuss other attempt at evaluating the block cipher and their modes of operation and emphasize of our work, here.

Mode of operation is one of the important parameter for block cipher. Need for mode of operation is described in detail in [7]. Generic composition of mode of operation and two types of authenticated encryption modes are described in [8].

Authenticated encryption and pros and cons of that and comparison of some existing block cipher mode of operation are explained in [17]. Single passed combined mode OCB is explained in [9] which is not patent free mode. Two passed combined mode EAX [18] and CCM [10] are patent free, but difficult to implement in hardware. Galois Counter Mode of operation is explained in detail in [11], which is patent free, high performance, online, easily implemented in hardware. There have been many evaluations of block ciphers and mode of operation for that. But none of them focus on the security of link layer architecture.

In [19] evaluations of ciphers are done. But they have not considered rijndael AES.

Law in [20] presents a detailed evaluation of the block ciphers. The evaluation is based on security properties, storage and energy efficiency of the ciphers. But this work does not consider GCM mode of operation and evaluation of the ciphers is not done within any link layer architecture.

In [21], evaluation of CCM and OCB mode is done for link layer security framework for wireless sensor networks, but GCM mode is not considered.

In [22], authors has presented AES-GCM core. They have implemented this core by taking into account two main aspects that it should provide a real throughput, capable of feeding a Gigabit Ethernet, and should be implemented in a commercial FPGA as a part of System-on-a Chip(SoC)[22]. But this work does not evaluated for any security of link layer architecture for WSNs.

In [23] also authors have implemented Galois Counter Mode on general purpose processors.So in summary we can say that none of the evaluation considers the evaluation of GCM with AES in link layer security architecture for WSNs, as we attempt to do here.

3. IMPLEMENTATION METHODOLOGY

In this section we present Tools employed, methodology used, test application and call-graph that we have generated.

3.1. Tools for Implementation

We have used TinySec link layer security framework in the TinyOS 1.x operating environment [24] with nesC[25] as the language of implementation. We have implemented GCM with AES for Mica2 motes. We made changes in TinySec to incorporate GCM in it. TinySec[13] is the link layer security protocol available with all releases of TinyOS. Additionally, TinySec is integrated into the TOSSIM [26] simulator, which runs on an Intel x86 platform.

We implemented GCM algorithm in nesC[25] code, the programming language used for TinyOS. Incorporating GCM in TinySec requires creation of certain modules and interfaces as well as modifications in some files of TinyOS.

We have used TOSSIM as simulator to simulate our implementation. We have used TestTinySec as sample application, which comes bundled with TinyOS, to test our implementation. For energy and CPU cycle analysis, we use Avrora[27], an instruction level event simulator. Using results obtained from TOSSIM and Avrora, we evaluate performance of Galois Counter Mode.

3.2. Methodology

Our evaluation is based on 2 step approach:

First, we simulated the performance of the ciphers and modes implemented in nesC, in the link layer architecture TinySec. We have used TOSSIM as the WSN simulator. The nesC compiler gives us output, the RAM and ROM requirements of the application under consideration.

We have determined throughput in bits/sec and the energy consumed using the Avrora simulator.

Interface

The interface specifies a set of named functions, called commands, to be implemented by the interface’s provider and a set of named functions, called events, to be implemented by the interface’s user. We have change BlockCipherMode interface file in original tinyOS version to incorporate our mode of operation.

Configuration

A nesC component is either a module or a configuration. Configuration file is responsible for wiring between different modules. We make changes in configuration file of TinySec viz. TinySecC. In this file implementation clause specifies a list of C declarations and definitions called translation-unit. We declare our module AESM as Cipher and GCMModeM as Mode.

Modules

As we have discussed, commands defined in interface must be implemented by provider of the interface. We implement these commands in our GCMModeM module.

3.3. Test Application

Every application in TinyOS is the collection of modules, configurations and interfaces. We use TestTinySec as sample application. This application comes bundled with TinyOS and used for testing operation of TinySec protocol. It implements a counter which is incremented every time tic. This counter value is sent to each node in the network. All messages are encrypted and authenticated over the air.

Sample Application with Original Configuration

Figure 1 shows the partial component graph of TestTinySec application. Component graph for pc can be generated by typing “make docs pc” at the command line in application’s directory. The main module file TinySecM is responsible for invoking all security related commands. TinySecC is the configuration file that shows wiring between different modules. It is shown in component graph that TinySecM uses interfaces provided by CBCModeM, CBCMAC, SkipJackM and LedsC.

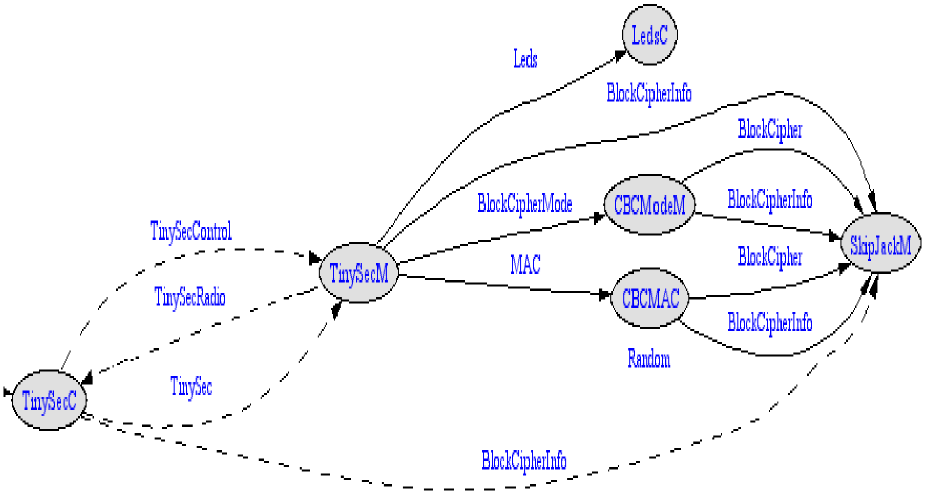

Figure 1. Part of Component Graph of Original TestTinySec Application.

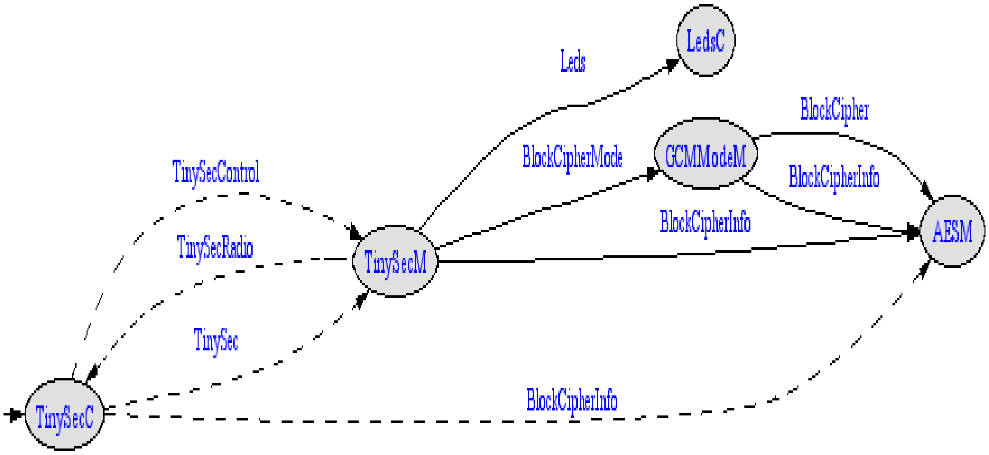

Figure 2. TestTinySec Application with AES - GCM

Application with GCM mode with AES

Figure 2 shows part of a component graph of TestTinySec application for GCM mode with AES cipher. As shown in Fig., we have implemented GCMModeM module and AESM which provides Authenticated Encryption mode of operation. This module is responsible for initializing context structure for GCM mode. As shown in Figure, TinySecM uses BlockCipherMode interface from GCMModeM modules and calls the command implemented therein to provide operation.

4. Performance Results and Analysis

In this, we attempted to add GCM mode with AES in TinySec. In order to do this, we implemented GCM with AES and tried to evaluate these based on different metrics viz.

memory, energy and CPU cycles. In this section, we show our experimental results for this schemes based on above mentioned metrics.

4.1. Storage Requirements

We have simulated our implementation GCM mode with AES for mica2 platform in TOSSIM.

RAM and ROM Usage

The Figure 3 shows ROM requirement for CBC mode with skipjack (Original configuration), CBC mode with AES and GCM mode with AES. From above figure we can clearly say that GCM-AES require more ROM than compared to default configuration (CBC-SKIPJACK), but it provides both confidentiality and authentication. It requires less ROM than CBC-AES.

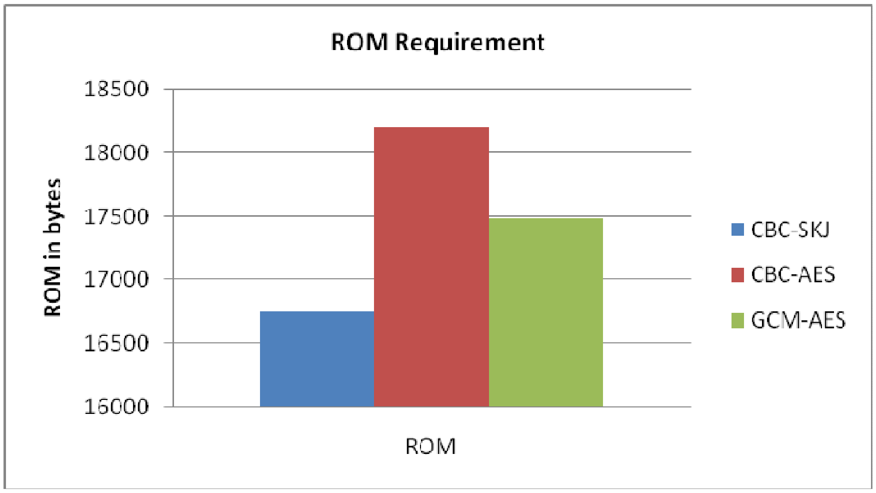

Figure 3. ROM Requirement

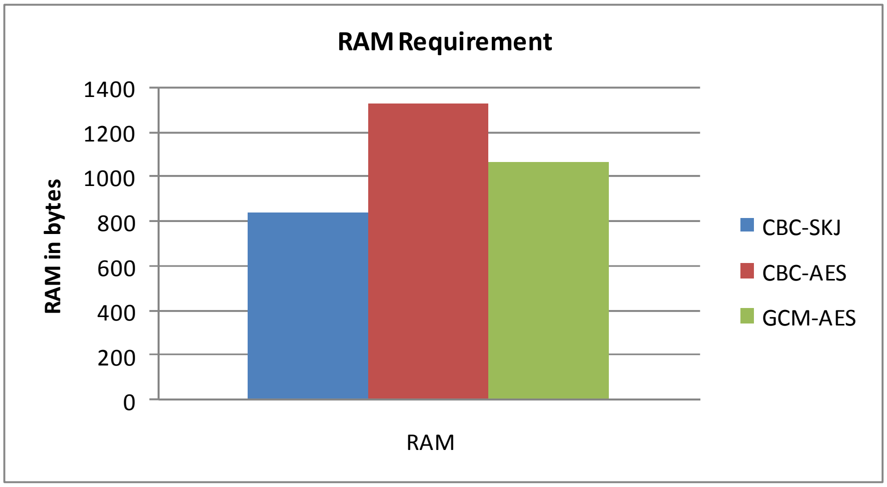

Figure 4. RAM Requirement

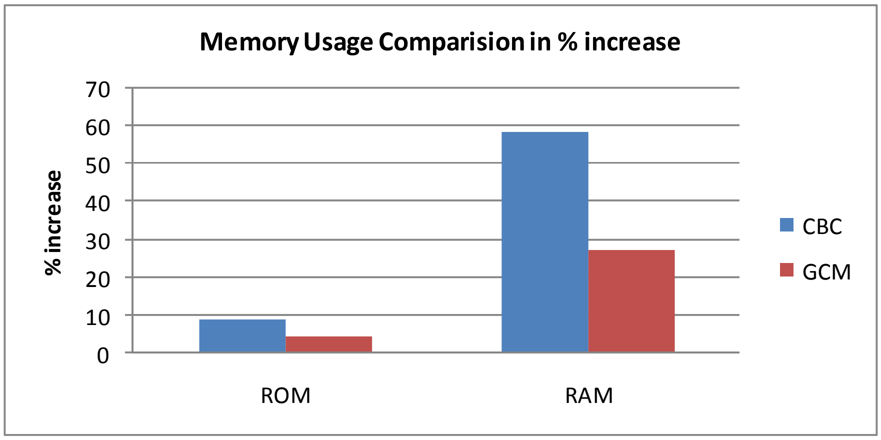

Figure 5. Memory usage Comparison in %increase

The Figure 4 shows RAM requirement for CBC mode with skipjack (Original configuration), CBC mode with AES and GCM mode with AES. From above figure we can clearly say that GCM-AES requires less RAM than CBC-AES.

Figure 5 shows memory usage comparison in % increase for CBC-AES and GCM-AES compared to original TinySec CBC-Skipjack. From figure we can see that for MICA2 motes with only 4KB of RAM, and overhead of only 27.38%, when using the GCM mode with AES implementation. The advantage is the increased security due to standard AES 128-bit cipher wired in GCM modes of operations.

4.2. CPU Cycle Usage

The Figure 6 shows CPU Cycle requirement for CBC mode with skipjack (Original configuration), CBC mode with AES and GCM mode with AES.

The Figure 7 shows %increase in CPU Cycle with compared to original configuration. We can see that measured CPU cycles in the CBC and GCM mode of operation when using AES-128 bit block ciphers are 379.74% and 211.24% more respectively, over the Skipjack cipher with CBC block cipher mode of operations. So we can say that when using GCM mode, penalty in terms of increased CPU cycle is much lesser than when using CBC mode of operation with AES -128 bit cipher.

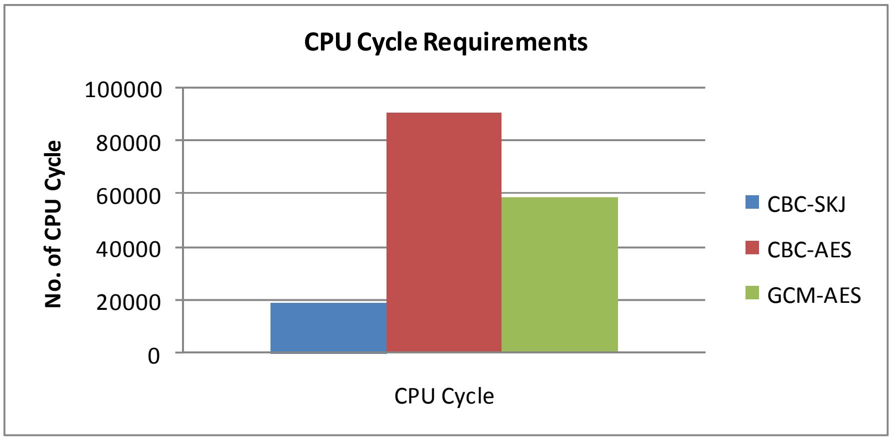

Figure 6. Requirement of CPU Cycle

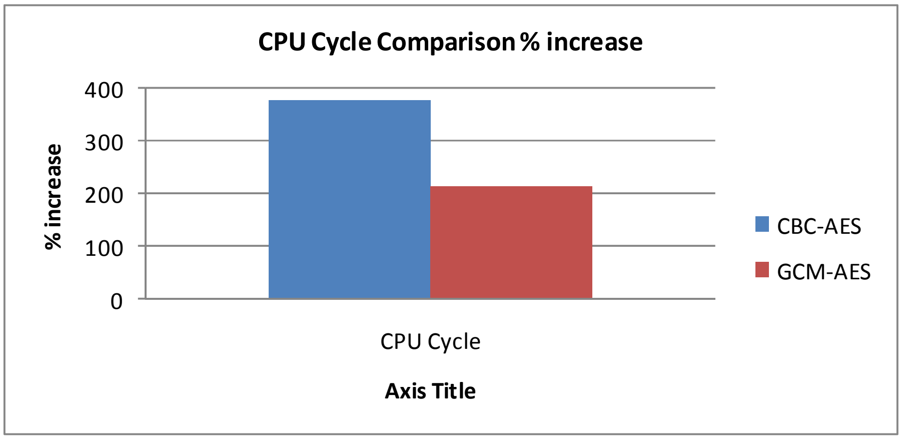

Figure 7. %increase in CPU Cycle

4.3. Throughput

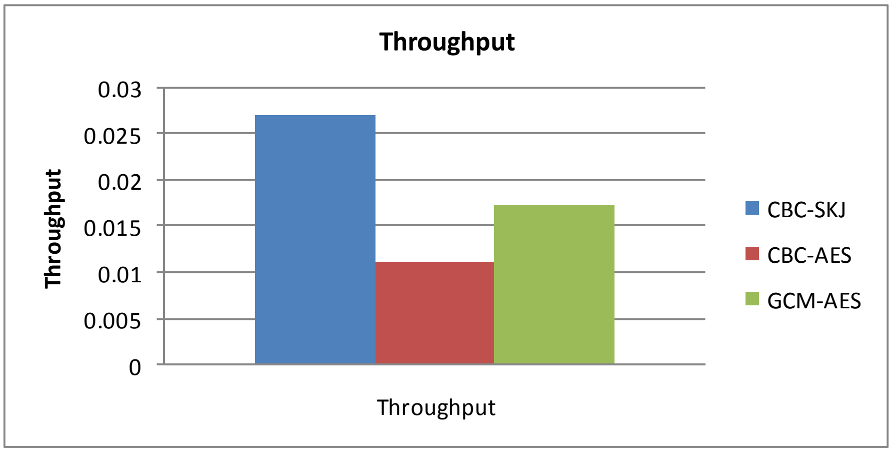

Figure 8. Throughput

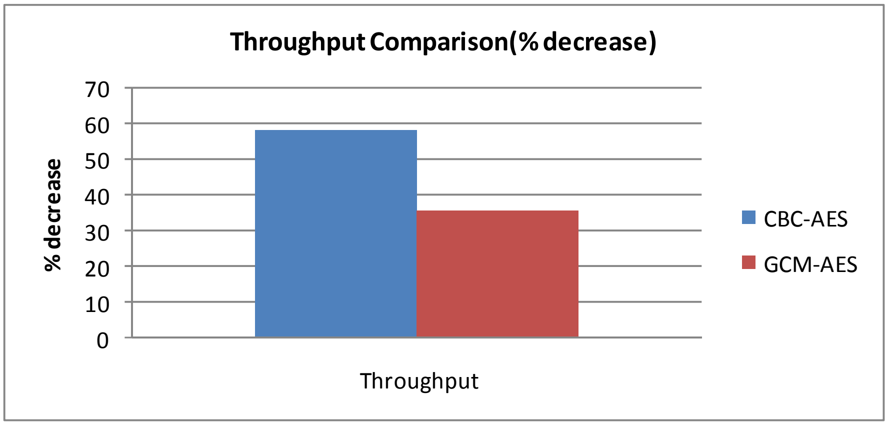

Figure 9. Throughput Comparison

4.4. Energy

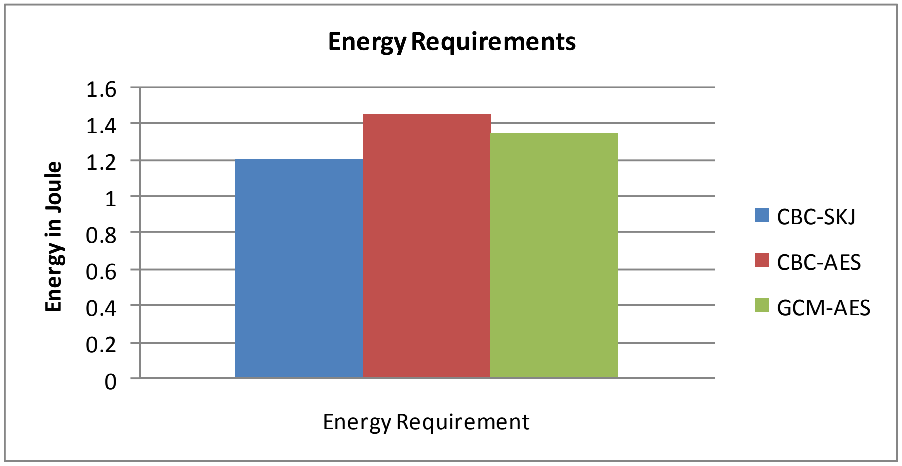

Figure 10. Energy Requirements

The Figure 10 shows Energy requirement for CBC mode with skipjack (Original configuration), CBC mode with AES and GCM mode with AES.

Figure 11 shows Energy comparison in % increase for CBC-AES and GCM-AES compared to original TinySec CBC-Skipjack. From the figure we can say that our results indicate lesser penalty in energy consumption for GCM mode than compared to original version, which is more significant.

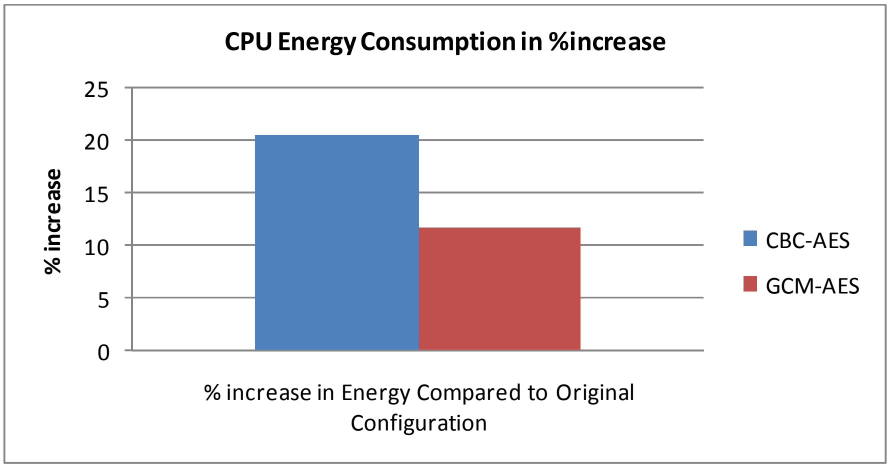

Figure 11. Energy Consumption in %increase

5. Conclusion and Future Work

In this, an attempt is made to investigate Galois Counter Mode of operation with 128-bit AES cipher in link layer security protocol, TinySec, for wireless sensor network. As compared to the default TinySec configuration offering the CBC mode for encryption and CBC-MAC for authentication, our perception that the authenticated encryption mode GCM would work well, indeed is justified. As per the experimental result that we obtain, using the GCM mode in TinySec entails only 12% increase in energy and 28% increase in RAM usage, while offering encryption as well as authentication as compared to default configuration of TinySec. Hence, if the underlying application demands confidentiality as well as message integrity, the GCM mode is preferable. This work can further be extended with (a) exploring any possibilities of further refining Galois Counter Mode to make it more efficient (b) since GCM is the only authenticated encryption mode that offers independent authentication (using GMAC), the feasibility of using GMAC (in place of CBC-MAC) is to be investigated (c) whether parallelizable operation of GCM in resource constrained environment is feasible or not that is to be investigated.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their comments which greatly enhanced this paper.

REFERENCES

[1] Wireless Sensor Networks: Getting Started Guide: Crossbow Technology Incorporated, http://www.crossbow.com

[2] Sasikanth Avancha, Jeffrey Undercoffer, Anupam Joshi, John Pinkston, “Security for Wireless Sensor Networks”, Kluwer Acedemic Publishers, Norwell, MA,2004

[3] Viega J, Chandra P, Messier M, “ Network Security with Openssl. O’Reilly & Associates, Inc.(2002).

[4] IPSec: Request For Comments. RFC 2401, RFC 2402, RFC 2406, RFC 2408, http://www.ietf.org/rfc/rfc240n.txt

[5] Devesh Jinwala, Dhiren Patel, K S Dasgupta; “A Security Attributes driven taxonomy of Wireless Sensor Network Applications”; *International Conference on Sensors and Related Networks (SENET07);* sponsored by VIT, Indian Nuclear Society and University of Applied Sciences, Germany; at Vellore Institute of Technology (VIT), Vellore, Dec 2007

[6] M. Bellare and C. Namprempre, “Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, Advances in Cryptology – ASIACRYPT 2000, volume 1976 of Lecture Notes in Computer Science, Pages 531-545, Springer-Verlag, Berlin Germany, Dec. 2000

[7] Laurent Haan, “Block Cipher modes of operation”, Available online at http://localhost/progressive-coding/tutorial.php?id=3&print=1.

[8] M. Bellare, P. Rogaway, and D. Wagner, “A conventional authenticated-encryption mode.”, Available online at http://eprint.iacr.org/2003/069/ , 2003

[9] Rogaway P, Bellare M, Black J, Krovetz T, “OCB: a block-cipher Modes of Operation for efficient authenticated encryption”, In ACM transaction on Information and System Security, pp.365-403, ACM, NY (2003).

[10] NIST Special Publication 800-38C: Recommendation for Block Cipher Modes of Operation : The CCM Mode for Authentication and Confidentiality , http://csrc.nist.gov/publications/nistpubs/SP800-38C/SP800-38C_updated-July20_2007.pdf.

[11] D. A. McGrew, J. Viega, “The Security and Performance of the Galois/Counter Mode (GCM) of Operation”, INDOCRYPT 2004, LNCS 3348, pp. 343-355, Springer-Verlag Berlin Heidelberg, 2004.

[12] NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation : The Galois Counter Mode (GCM) and GMAC , http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

[13] Chris Karlof, Naveen Sastry, and David Wagner,” TinySec: A link layer security architecture for wireless sensor networks”, in Proc. of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), November 2004.

[14] NIST Special Publication 800-38a: Recommendation for Block Cipher Modes of Operation Methods and Techniques http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[15] Mihir Bellare, Joe Kilian, Phillip Rogaway, “The security of the cipher block chaining message authentication code”*, Journal of Computer and System Sciences*, Vol 61 Isssue 3, pp.:362-399, December 2000.

[16] IAPM – Integrity Aware Parallelizable Mode from Wiki.avilable at http://wikipedia.org/

[17] D. A. McGrew, J. Viega, “The Galois/Counter Mode (GCM)”, Technical Report, Submitted to NIST Modes of Operation Process, May 31, 2005.

[18] M. Bellare, P. Rogaway, D. Wagner,”The EAX Mode of Operation”, in Proc. Of Fast Software Encryption ’04, Springer-Verlag (2004).

[19] Luo X, Zheng K, Pan Y, Wu Z,”Encryption algorithme comparison for Wireless Networked Sensors”, in Proc. Of the IEEE International Conference on Systems, Man and Cybernetics, pp 1142-1146, IEEE (2004).

[20] Law Y, Doumen J, Hartel P,”Survey and Benchmark of Block Ciphers for Wireless Sensor Networks”, ACM Transaction on Sensor Networks, Vol 2, Issue 1, 65-93(2006).

[21] Devesh Jinwala, Dhiren Patel, Kankar Dasgupta, “ Optimizing the block cipher and modes of operation overhead at the Link Layer Security framework in Wireless Sensor Networks “, in Proceeding of ICISS 2008, LNCS 5352, pp. 258-272,2008, Springer – Verlag , Berlin Heidelberg 2008.

[22] Becker Jürgen, Woods Roger, Athanas Peter, Morgan Fearghal, Lázaro Jesús, Astarloa Armando, Bidarte Unai, Jiménez Jaime, Zuloaga Aitzol, “AES-Galois Counter Mode Encryption/Decryption FPGA Core for Industrial and Residential Gigabit Ethernet Communications Reconfigurable Computing: Architectures, Tools and Applications”, Lecture Notes in Computer Science,2009, Springer Berlin / Heidelberg, pp. 312-317.

[23] Gueron, S. and Kounavis, M., “Efficient implementation of the Galois Counter Mode using a carry-less multiplier and a fast reduction algorithm”, *Inf. Process. Lett.* 110, 14-15 (Jul. 2010), 549-553. DOI= http://dx.doi.org/10.1016/j.ipl.2010.04.011,2010.

[24] J. Hill, et al., "System Architecture Directions for Networked Sensors," Proc. 9thIntl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), ACM Press, 2000, pp. 93-104.

[25] David Gay, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, and David Culler. The nesC language: A holistic approach to network embedded systems. In Programming Language Design and Implementation (PLDI), June 2003.

[26] Philip Levis and Nelson Lee. “ TOSSIM: A Simulator for TinyOS Networks Version 1.0 June 26, 2003”

[27] Ben L. Titzer, Daniel Lee, and Jens Palsberg. Avrora: Scalable sensor network simulation with precise timing. In Proc. of the 4th Intl. Conf. on Information Processing in Sensor Networks (IPSN), Los Angeles, CA, April 2005.

Authors

Ms. Vivaksha Jariwala was born on 23rd November 1980. She is lecturer in Computer Engineering with C. K. Pithawalla College of Engineering and Technology, Surat (India). She is research scholar at Sardar Vallabhbhai National Institute of Technology, Surat (India). Her major areas of interest are Information Security Issues in Resource Constrained Environment and Software Engineering.

Dr. Devesh Jinwala was born on 3rd July 1964. He is an Associate Professor in Computer Engineering with Sardar Vallabhbhai National Institute of Technology, Surat (India). His research work is on Configurable Link layer Security Protocols for Wireless Sensor Networks. His major areas of interest are Information Security Issues in Resource Constrained Environment, Algorithms & Computational Complexity and Software Engineering.