TriSec: A Secure Data Framework for Wireless Sensor Networks Using Authenticated Encryption

Pardeep Kumar, Sang-il Cho, Dea-Seok Lee, Young-Dong Lee and Hoon-Jae Lee, *Member, KIMICS*

Abstract— **Wireless sensor networks (WSNs) are an emerging technology and offers economically viable monitoring solution to many challenging applications. However, deploying new technology in hostile environment, without considering security in mind has often proved to be unreasonably unsecured. Apparently, security techniques face many critical challenges in WSNs like data security and secrecy due to its hostile deployment nature. In order to resolve security in WSNs, we propose a novel and efficient secure framework called TriSec: a secure data framework for wireless sensor networks to attain high level of security. TriSec provides data confidentiality, authentication and data integrity to sensor networks. TriSec supports node-to-node encryption using PingPong-128 stream cipher based-privacy. A new PingPong-MAC[1] (PP-MAC) is incorporated with PingPong stream cipher to make TriSec framework more secure. PingPong-128 is fast keystream generation and it is very suitable for sensor network environment. We have implemented the proposed scheme on wireless sensor platform and our result shows their feasibility.**

Index Terms— **Wireless sensor networks, security, TriSec, stream cipher, Confidentiality, Authentication, Integrity, PingPong-MAC.**

I. INTRODUCTION

THIS paper is a comprehensive and modified new version of [1]. Whilst continuing its original format and content, all of the sections have been enlarged, incorporating more in-depth analysis and characteristics discussion. There are more technical details incorporated with discussion regarding the suitability and usefulness of the system, supported by more current research.

Wireless sensor networks (WSNs) is a combination of sensor nodes, base-station and communication interface. These sensor networks considered as a discrete group of independent nodes with low cost, low power, limited computation and less memory. They communicate wirelessly over limited frequency and low bandwidth [2]. Unlike traditional networks, sensor networks are deployed in hostile environment to execute their tasks.

Nowadays, WSNs have become popular solutions to many challenging domestic, commercial, military, healthcare, environmental and agricultural applications. These sensor nodes collectively monitor the area and generate a substantial amount of data which is transmitted it to the base-station using one node to another node via RF signals and routing algorithms.

Providing security in sensor networks is not an easy task. The data redundancy, less memory and limited energy are the main constraints characteristics of WSNs. These characteristic raise challenges as compared to conventional security networks measures. WSN seems more vulnerable to attacks, which are more difficult to launch on wired networks. Thus, security has emerged as one of the most important issues for sensor network applications.

Although significant numbers of research have been conducted in order to overcome the potential threats in sensor networks, but these are not adequate. Many implemented WSNs systems are suffering from the lack of link layer security features. In other cases, where the security features are implemented, those are too resource consuming. None of the existing system is able to simultaneously ensure low energy, memory consumption and provide complete security. Thus, there is a need for a better security system which can combine low operational costs with a high security performance. The objective of this research is to propose energy efficient security primitives for wireless sensor network environment. The proposed solution will be able to ensure complete security to the WSNs system and at the same time it will need low energy and memory consumption without compromising the level of security.

So in this paper, we have taken up above resource challenges and proposed a new scheme called TriSec: energy efficient secure data framework for wireless sensor network. In TriSec, a data protection is incorporated into a sensor packet frame by means of symmetric encryption with the PingPong-128 stream cipher and PP-MAC message authentication code. In order to minimize the computation cost of PP-MAC algorithm, TriSec employing on some of the data already computed on PingPong-128 stream cipher.

Manuscript received March 12, 2010; revised March 17, 2010; accepted March 28, 2010.

HoonJae Lee is with the Division of Computer and Information Engineering, Dongseo University, Busan, 617-716, Korea (Email: hjlee@dongseo.ac.kr)

[1] MAC is used as message authentication code, otherwise explained.

TriSec framework using another approach and provides a robust security package for resource constraint platforms. In our scheme, a symmetric stream cipher is used called PingPong-128 [3] for ubiquitous application. TriSec provides three security services namely: data confidentiality, data authentication and data integrity. In addition, a Message Authentication Code (MAC) gives two-party data authentication as well as data integrity.

The rest of the paper is organized as follows: In section II we review the related work. In section III we recall some background of sensor node and security issues for WSN. In section IV, we discuss for security at link layer. In section V we describe security goals of TriSec secure data framework. In section VI gives detailed design of proposed TriSec. In section VII experimental setup, implementation, evaluation of TriSec and compare our results with others found in the literatures. In section VIII , we conclude the security of TriSec.

II. RELATED WORKS

In this section we review the related works for security schemes that have been proposed for sensor networks.

Perrig et al. [5] proposed a security protocol SPINS for wireless sensor networks. SPINS consists of two secure building blocks. (1) Secure Network Encryption Protocol (SNEP), provides two party data authentication (point-to-point) communication. (2) micro-Timed Efficient Streaming Loss-Tolerant Authentication Protocol (μ-TESLA), provides efficient authenticated broadcast communication. In this scheme, all cryptographic primitives are constructed based on a single block cipher scheme. Author selected RC5 block cipher because of its small code size and high efficiency. RC5 is also suitable for ATmega platform because of memory constraints. A hash function is used with block cipher.

Karlof et al. [6] proposed another most popular wireless security architecture known as "TinySec: a Link Layer Security Architecture for Wireless Sensor Networks". TinySec carried out for wireless sensor networks, achieves low energy consumption and memory usage. TinySec provides access control, message integrity and confidentiality. In sensor networks, data travels on carrier sense in which node check, if another node is also currently broadcasting, than node will be vulnerable to Denial of Service (DoS) attack. TinySec security architecture gives protection from Dos attack and able to detect illegal packets when they are injected into the network. TinySec consists of two building blocks: (1) Authenticated Encryption mode denoted as TinySec-AE. In this mode, the data packet payload is encrypted and the whole packet is secured by a Message Authentication Code (MAC). (2) Authentication only denoted as TinySec-Auth. In this mode, the entire packet is authenticated with a MAC, but the whole data packet is not encrypted. Author has tested two 64-bit block ciphers, i.e. Skipjack and RC5 for Authenticated Encryption mode and Authentication only mode. RC5 is more difficult to implement than Skipjack, so author selected Skipjack as the default secure block crypto algorithm. One of the major drawbacks of TinySec, it does not attempt to protect from replay protection [7]. The replay protection is intentionally omitted from TinySec [7].

MiniSec [7] is the first fully-carried out general function security protocol for the Telos sensor motes. MiniSec provides two controlling modes i.e. unicast and broadcast, hence, recognized as MiniSec-U and MiniSec-B, respectively. Both of the methods are using the OCB-encryption system to allow data confidentiality and authentication by using a counter as a nonce. MiniSec provides the replay protection to the sensor nodes.

These researches [5]-[7] have been implemented at efficient link layer security protocol with an efficient block cipher and keying mechanism.

A TinyPK [8] protocol also has been proposed for WSN. TinyPK is designed specifically to allow authentication and key agreement between resource-constrained sensors. The protocol is designed to be used in conjunction with other symmetric encryption based protocols as TinySec. In order to deliver secret key to the protocol, author implemented the Diffie-Hellman key exchange algorithm.

Lee et al. [9]-[11] has proposed Dragon-MAC for wireless sensor networks. In this scheme Encrypt-then-Mac is used by means data is firstly encrypted then MAC is computed. Two keys *ke, km* are used for encryption and authentication, respectively. Authors have implemented the schemes only for Telos family [9], [10].

Ahmad et al. [12] analyzed known authenticated encryption schemes using eSTREAM cipher for sensor networks.

III. SENSOR NODE AND SECURITY ISSUE

Sensor node is a basic unit of sensor networks; these nodes are characterized as resource hungry device in terms of energy, memory and computational ability. A typical sensor (MICA) has around 8MHz microcontroller, 2 ~ 10kb RAM and less than 128kb flash memory. Some sensors are powered by standard AA batteries and their lifetime is limited. On account of its limited storage space and energy supply, lightweight modules for sensor nodes are aggressively sought.

Due to these constraints it is very difficult to directly apply the traditional security schemes to sensor network. All security schemes requires additional amount of resources for software implementation, such as data memory, code space and energy to power the sensor.

Furthermore, these devices are easy to physically destroyed due to the distributed nature, physical manipulation and monitoring of them made even difficult.

IV. LINK LAYER SECURITY

End-to-end security mechanisms are not possible in sensor network as compared to traditional network such as SSH [15], IPSec [16] and SSL [17] protocols. These protocols are based on route-centric multi-hop communication, in which the intermediate router only need to view the packet header and it is not necessary for them to have access to packet bodies. They are considered inappropriate since they are not allowed in-network processing and data aggregation which plays important role in energy efficient data retrieval.

In sensor networks, it is important to allow intermediate nodes to check message integrity and authenticity because they have many-to-one multi-hop communication nature. The intermediate nodes carry out some form of data processing on incoming data packets to be routed towards the base station. Thus, in-network processing requires intermediate nodes to access, modify, and suppress the contents of messages; it is very unlikely that end-to-end security schemes are used between sensor nodes to base-station to guarantee the message integrity, authenticity and message confidentiality [6]. Link-layer security architectures can detect unauthorized packets when they are first injected into the network, whereas in end-to-end security mechanisms, the network may route packets injected by an adversary many hops before they are detected. These kinds of attacks waste the energy and bandwidth.

V. SECURITY GOALS OF TRISEC

TriSec provides three basic link layer security services: message[2] confidentiality, message authenticity and message integrity.

A. Message confidentiality

Message confidentiality is a service, in which message is used by only authorized users. In sensor networks, message should not be leaked to neighboring node because sensor deals with very sensitive data. In order to provide the security the message should be encrypted with secret key. Secret key is intended to recipient only, hence achieved confidentiality.

B. Authentication

Authentication service is associated to identification. Entity authentication function is important for many applications and for administrative task. Entity authentication allows to receiver, to verify that the data is really sent by authenticated sender or not. In node-to-node communication entity authentication can be achieved through symmetric mechanism: a message authentication code (MAC) is compute on secret shared key for all communicated data.

C. Message Integrity

Message integrity is a service, which addresses the illegal alteration of message. To conformation of message integrity, one must have the ability to identify data manipulation by illegal parties.

VI. DESIGEN OF TRISEC

A TriSec framework is designed based on massage authentication code, which provides message confidentiality, authenticity, and message integrity to sensor nodes. Our scheme is employing on PingPong-128 stream cipher based-privacy.

A. Definition of PingPong-MAC (PP-MAC)

To establish an end-to-end trusted and secure communication between sender and receiver for a message, message authenticity and message integrity are achieved by message authentication code (MAC).

The authentication scheme is based on an internal state being transformed along with the progress of encryption progress. This results from the fact that the scheme employs PingPong-128 algorithms. This feature substantially reduces the excessive program space needed by the MAC scheme.

B. PingPong-128

A PingPong-128 [3] stream cipher is proposed by Hoon Jae Lee and Kevin Chen in 2007. PingPong family is based on summation generator stream cipher with addition of mutual clocked control structure. This algorithm is designed with both security and efficiency in mind to satisfy the need for lightweight algorithms.

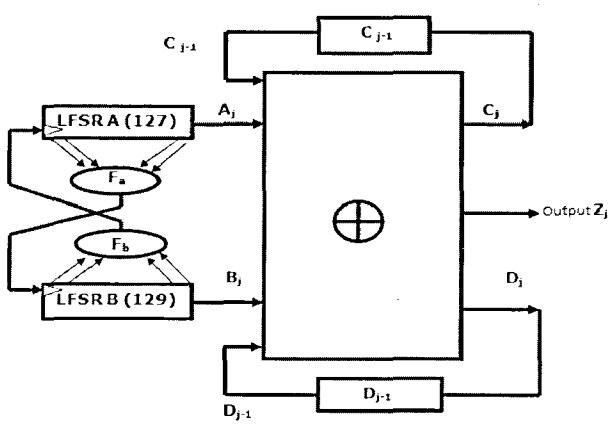


Fig. 1 PingPong-128 keystream generator.

PingPong is highly secure algorithm, dedicated to hardware environment and easy to implement in software.

[2] sometime messages are represented as data.

PingPong-128 is a bit based stream cipher as shown in figure 1. This stream cipher is constructed on two mutually clocking LFSRs and a single memory bit. PingPong-128 accept key as 128-bits and 128-bits initialization vector to feed the internal states. It generates an output block of 128 pseudo-random bits from a combination of the internal states, for each iteration. PingPong-128 has 257-bits of internal state.

1) Description of Keystream Generation

The PingPong-128 [3] generator produces the output keystream by combining the LFSR sequences and the memory sequence. PingPong-128 has two mutually clocking LFSRs *La*, *Lb*, and a single bit of memory *c*. Two primitive polynomials, Pa(x) and Pb(x) are following:

$$P_a(x) = x^{127} \oplus x^{109} \oplus x^{91} \oplus x^{84} \oplus x^{73} \oplus x^{67} \oplus x^{66} \oplus x^{63} \oplus x^{56} \oplus x^{55} \oplus x^{48} \oplus x^{45} \oplus x^{42} \oplus x^{41} \oplus x^{37} \oplus x^{34} \oplus x^{30} \oplus x^{27} \oplus x^{23} \oplus x^{21} \oplus x^{20} \oplus x^{19} \oplus x^{16} \oplus x^{13} \oplus x^{12} \oplus x^{7} \oplus x^{6} \oplus x^{2} \oplus x^{1} \oplus 1 \quad (1)$$

$$P_b(x) = x^{129} \oplus x^{125} \oplus x^{121} \oplus x^{117} \oplus x^{113} \oplus x^{109} \oplus x^{105} \oplus x^{101} \oplus x^{97} \oplus x^{93} \oplus x^{89} \oplus x^{85} \oplus x^{81} \oplus x^{77} \oplus x^{73} \oplus x^{69} \oplus x^{65} \oplus x^{61} \oplus x^{57} \oplus x^{53} \oplus x^{49} \oplus x^{45} \oplus x^{41} \oplus x^{37} \oplus x^{33} \oplus x^{29} \oplus x^{25} \oplus x^{21} \oplus x^{17} \oplus x^{13} \oplus x^{9} \oplus x^{5} \oplus 1 \quad (2)$$

Two linear feedback shift register (LFSR) *La* and *Lb* are mutually clock controlled by the function *fa*(La) and *fb*(Lb). Two mutual clock controlled structure is used to provide irregular clocking of LFSRs which will increase the non-linearity of the output keystream.

2) Clock control functions

Two clock controlled function are defined as:

$$fa(La) = 2 * La42(t) + La85(t) + 1 \quad (3)$$

$$fb(Lb) = 2 * Lb43(t) + Lb86(t) + 1 \quad (4)$$

As shown in figure 1, at instant time *j*, the output of the LFSR A (*La*) and LFSR B (*Lb*) are denoted by *aj* and *bj* respectively while *cj* and *dj* represents the memory bit. The memory bits are defined by the function *fc* and *fd* respectively and at time *j* these functions are as follows:

$$cj = fc(aj, bj, cj-1) = ajbj \oplus (\text{aj} \oplus \text{bj})\text{cj-1} \quad (5)$$

$$dj = fd(aj, bj, dj-1) = bj \oplus (\text{aj} \oplus \text{bj})\text{dj-1} \quad (6)$$

The output of the keystream generator is obtained by combining the output of the LFSR sequences and the memory bit sequence. The output sequence at time *j* is denoted by *zj* and defined as:

$$zj = aj \oplus bj \oplus cj-1 \oplus \text{dj-1} \quad (7)$$

3) Key Initialization and Rekeying

For detailed specification, refer to [3]. Initially, 128-bits key (*ke*) and 128- bits initialization vector (*IV*) together feed to 257 internal states of PingPong-128. To generate the initial state generator used itself twice.

- The initial state of *La* is simply obtained by XORing of two 128-bits binary strings of the key (*ke*) and *IV* by means $La = (Ke \oplus IV) \bmod 2^{127}$.
- The initial state of 129 bits for *Lb* is simply obtained by assuming the 128-bits key are embedded into 129-bits word and shifted one bit left. Then XORing with the *IV* embedded into 129 word with a leading zero, by means $La = (ke<<1) \oplus (0|IV)$.
- Now cipher runs a second time to produce an output string of length 257-bits.

It is very unlikely if LFSRs will be initialized with all the zero states. By employing the PingPong algorithm itself, it makes advantage for fast implementation.

C. PP-MAC Design

The scheme is designed to achieve the general security requirement discussed in section VI-B. It is assuming that the underlying primitives of PingPing-128 are secure. It is possible to build a proof of the given notion of security of the MAC procedure as shown in Table 1.

TABLE 1
PINGPONG MAC PROCEDURE

Let *Pt* be denoted as Plaintext
Let *Ct* be denoted as Ciphertext
Let *ke* be denoted as encryption key
Let *km* be denoted as MAC encryption key
Let *Ct[i]* be denoted *i-th* 32-bit word of ciphertext.

1. $Ct = E_{ke}(Pt)$.
2. $\{a, b, c, d\} = km(128\text{-}bit)$
   $IV = [destpan||addr||type||group||counter]*2$
3. $\{a, b, c, d\} = Ct[i] \oplus$ a, b, c, d;
4. $PingPong\text{-}MAC = a \oplus b \oplus c \oplus d$.
5. ***Output MAC(32-bit)***

The encrypted ciphertext (*Ct*) is splitting into 32-bit blocks, and then padding the last word with zeroes, if required. Meanwhile, the MAC encryption key (*km*) is fed through variables *a, b, c, d* and then this key is XORing with 32-bit Ct with 32-bit of *a*, and 32-bit MAC can be obtained by XORing of all (*a, b, c, d*) outputs.

To integrate our authenticated encryption procedure into sensor network, we are adding 2 bytes of counter

(*ctr*) and 4-bytes MAC into default radio stack as shown in figure 2. A 2 bytes ctr is used to achieve the semantic security and 4 byte MAC is authentication.

Len	Fcfhi	Fcflo	Dsn	DestPAN	Add	Type	Grp	D_len	Data	CTR	MAC
1	1	1	1	2	2	1	1	1	2S	2	4

Fig. 2 Modified radio stacks of Telos Rev B.

D. PP-MAC Analysis

Practically, if IV should be message-unique for encrypted message with the same key, then it will not give additional rooms to an attacker. Since the IV is taken from the packet header of modified radio and sent to the decryption end, the 2 bytes counter (*ctr*) gives 2^{16} variations to the IV. This security property is very necessary to guarantee that message encrypted with same key should give different ciphertext every time. The MAC length indirectly implies the computation cost which would be needed to forge the MAC in chosen ciphertext attack. Chang et al [4], Zoltak et al [14] and Karlof et al [6] suggested the MAC length, MAC=4 bytes gives a well sufficient security and easy to implement. Practically 4 bytes of MAC is sufficient to wireless sensor network.

E. Operation mode of PP-MAC

The operation of PP-MAC is as follows: A Sender party[3] simply computes a MAC on the packet with MAC key (*km*) and encrypted message, and then authenticated packet will send to receiver party. When receiver node received the authenticated packet, then he/she can verify that the packet is sent by corresponding node or not and no information has been altered in transit. PP-MAC is an Encrypt-then-MAC stream cipher mode as shown in figure 3.

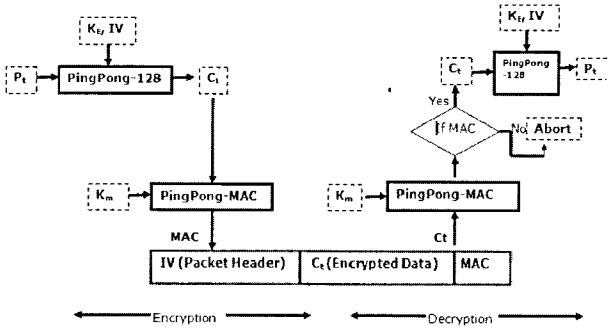


Fig. 3 Operation mode of PingPong-MAC.

Bellare et al [14], Lim et al [10] and Ahmad et al [12] suggested that the strongest definition of security for authenticated encryption can be achieved via Encrypt-Then-MAC approach only.

Encrypt-then-MAC:

$Eke,km(Msg)=E_{ke}(Msg)||T_{km}(E_{ke}(Msg))$ always provide privacy and authenticity to message [10].

[3] Party represent as a sensor node.

VII. EXPERIMENTAL SETUP, IMPLEMENTATION AND EVALUATION

A. Experimental Setup

Figure 4 shows the experimental setup. In the experiment sensor mote 'A' acts as a sender node, while sensor mote 'B' serves as base-station between the sender and PC receiving mote A's wireless packets. Mote B's is serially connected to PC via USB.

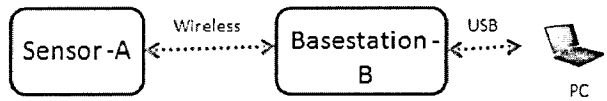


Fig. 4 Experimental Setup.

B. Implementation

For the implementation of proposed secure data framework, we are avoiding some of the related constraints to wireless sensor network. The memory occupied, power consumed are the main issues related to sensor node that we have taken up into consideration. Proposed framework gives PingPong-128 based-privacy as well as reliability for the information and also ensures the authenticity to sensor nodes. Messages encryption provides the Confidentiality. PP-MAC also ensures the integrity of the messages. For the implementation of PP-MAC, we choose TelosB [18] sensor node. TelosB is a very low power communication wireless component for ubiquitous applications. The specifications of sensor used in this proposed method's implementation, are shown in Table 2.

We have implemented TriSec framework on TinyOS [19], an event-driven open source operating system specially designed for wireless sensor networks. The code is written in nesC [20] for portability reasons.

TABLE 2
TELOSB SPECIFICATIONS

TelosB	
ITEMs	**DESCRIPTION**
Processor	16-bit RICS
Internal Memory	10-kb RAM
Flash Memory	48-kb ROM
Multi-Channel Radio	2.4-GHz(CC2420)
Interface	USB (UART)
Sensors	Temperature, Humidity, Light, etc.

C. Evaluation

As a proof of implementation, we had implemented PingPong-128 stream cipher in wireless sensor networks and results are shown in Table 3.

Our scheme is taking 598 bytes RAM and 16,474 bytes of program space for encryption. By doing so, we achieved a secure pair wise communication between neighboring sensor nodes. We also had implemented PP-MAC in Telos rev B sensor mote. It achieves two-party authenticated communication and data integrity for transmitted packet. PP-MAC required additional 25 bytes RAM and 955 bytes of ROM.

TABLE 3
EXPERIMENTAL RESULTS

Description	ROM (Bytes)	RAM (Bytes)	Execution Time (ms)
Without security scheme	15,354	470	-
PingPing-128 (Encryption only)	16,474	598	14.43
PingPong-MAC (Encrypt. & Auth.)	17,429	623	18.35

To evaluate the execution cycles required for each component such as PingPong-128 encryption, decryption and message authentication. A MSP430 internal built-in timer (*localTimer*) interface is used to calculate the execution time. Two time variables are allocated at the beginning and ending of targeted component respectively. After measuring the difference between two packets are calculated and forwarded to a PC through UART connection. Time to execute cipher operations on the 4MHz TelosB sensor node marked at 14.43 ms and 18.35 ms for PingPong-128 encryption and PP-MAC, respectively.

Furthermore, we have calculated the expected latency overhead incurred, if the packet length is increased then transmit time is also increased, as shown in Table 4. Analytically, standard Telos radio stack packet transmission time is 2.016 ms and PingPong-MAC radio stack packet transmission time is 2.208 at 250 kbps bandwidth.

D. Memory Comparison with other existing security architectures

Lee et al. [9] has implemented Dragon stream cipher to support link layer encryption on TelosB sensor platform. As shown in figure 5, it takes around 915 bytes of RAM and 17,583 bytes of ROM for encryption only.

Kausar et al. [21] has simulated HC-128 and Rabbit stream cipher on TinyOS and TOSSIM environment for sensor networks. It has been found that, the implementation of HC-128 is very expensive for in term of memory requirement as shown in figure 5.

TinySec [6] is a software package implemented to support link layer encryption on MicaZ sensor platform. As shown in figure 5, it consumes 768 bytes RAM and 7,146 bytes of ROM.

Our new proposed TriSec is taking only 598 bytes of RAM and around 16k of ROM for encryption operation in TelosB sensor platform.

TABLE 4
EXPETED LATENCY OVERHEAD INCURRED

Description	Pay-load (Bytes)	Packet Over-head (Bytes)	Total Size (Bytes)	Trans-mission time (ms)	Over-head inc. %
Tiny Sec-AE	24	42	68	28.3	7.9
TinyOS stack	24	39	63	26.2	_
Telos radio stack	24	39	63	2.016	_
MiniSec	24	25	49	1.568	-
PingPong-MAC	24	45	69	2.208	9.5

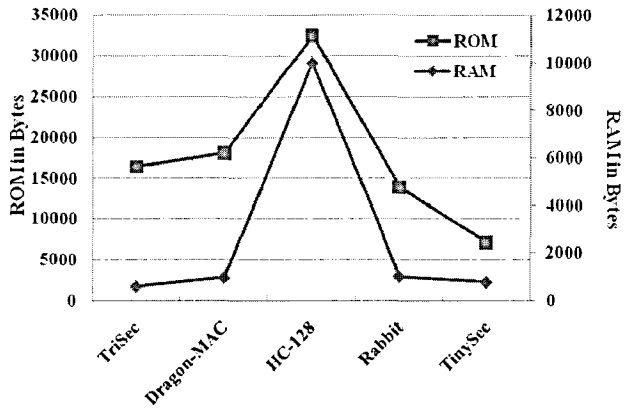


Fig. 5 Comparison of TriSec with existing schemes.

VIII. CONCLUSIONS

This paper addresses security in sensor network where energy and computation power present important role. We have designed PingPong-MAC (PP-MAC) algorithm for resource constraints devices. PP-MAC is employing on some of already computed data underlying PingPong-128 cipher. The salient features of PingPong-128 Method are its fast key generation and fast software implementation, good primitives for security such as encryption, authentication, decryption and data integrity.

The entity verification and message authentication have been tested through the performance of authenticated encryption schemes using TelosB sensor nodes for wireless sensor networks.

The implementation of its features can revolutionize the security primitives in wireless sensor networks. The remaining feature of PingPong-128 can be enhanced and implemented in wireless sensor networks as per the applications scenarios.

REFERENCES

[1] P. Kumar, S-I Cho and H. J. Lee, "PingPong-MAC: Secure Ubiquitous Sensor Network with Authenticated Encryption." In proc. of 2nd International Conference on Interaction Sciences, 2009, pp. 256-261.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramamiam and E. Cayirci, "A Survey on Sensor Networks," IEEE Communication Magazine, August 2002, pp. 102-114.

[3] H. J. Lee and K. Chen, "A New Stream Cipher for Ubiquitous Application," ICCIT, 2007, South Korea.

[4] C-C Chang, D. J. Nagel and S. Muftic, "Balancing Security and Energy Consumption in Wireless Sensor Networks" MSN 2007, LNCS 4864, pp 469-480.

[5] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks", Proceedings of 7th Annual International Conference on Mobile Computing and Networks (MOBICOM 2001), Rome, Italy July 2001.

[6] C. Karloff, N. Sastry and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks", Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD, November 2004.

[7] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: A Secure Sensor Network Communication Architecture" IPSN'07, April 2007, Cambridge, Massachusetts, USA.

[8] R. Watro, D. Kong, S-F Cuti, C. Gardiner, C. Lynn and P. Kruus, "TinyPK: Securing Sensor Networks with Public Key Technology", Workshop on Security of Ad-hoc and Sensor Networks, Washington DC, USA 2004.

[9] P. Kumar and H. J. Lee, "A Secure Data Mechanism for Ubiquitous Sensor Network with Dragon Cipher", 2009, IEEE 5th International Joint Conference on INC, IMS and IDC.

[10] S. Y. Lim, C. C. Pu, H. T. Lim, and H. J. Lee, "Dragon-MAC: Securing Wireless Sensor Networks with Authenticated Encryption", 2007. [http://eprint.iacr.org/2007/204.pdf]

[11] K. Chen, M. Henricksen, W. Millan, J. fuller, L. Simpson, E. Dawson, H. J. Lee and S. Moon, "Dragon: A fast word based stream cipher". 2005/006, ECRYPT Stream Cipher Project Report.

[12] S. Ahmad, A. Wahla and F. Kausar, "Authenticated Encryption in WSN Using eSTREAM Ciphers", ISA 2009, LNCS 5576, pp. 741-749.

[13] S. Mahlknecht, "Energy-Self-Sufficient Wireless Sensor Networks for the Home and Building Environment", Doctor's thesis, Technical University of Vienna, 2004.

[14] B. Zoltak, "An Efficient Message Authentication Scheme for Stream Cipher", Cryptology ePrint Archive 2004.

[15] T. Ylonen, "SSH- Secure login connection over the Internet", 1996, 6th USENIX Security Symposium.

[16] Security Architecture for the Internet Protocol. RFC2401, 1998.

[17] OpenSSL. http://www.openssl.org

[18] http://www.maxfor.co.kr/online%20brochure.pdf

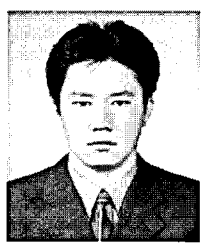
[19] http://www.tinyos.net/

[20] http://www.tinyos.net/tinyos-1.x/doc/nesc/ref.pdf.

[21] F. Kausar and A. Naureen, "A Comparative Analysis of HC-128 and Rabbit Encryption Schemes for Pervasive Computing in WSN Environment", 2009, ISA 2009, LNCS 5576, pp.682-691.

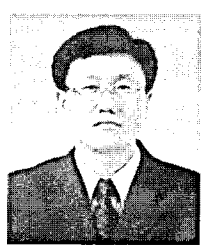
Pardeep Kumar received BE in computer science and engineering in 2002 from Institute of Technology & Management, Haryana and M-Tech in computer science & engineering in 2006 from Choudhary Devilal University, Haryana, India. Currently he is pursuing PhD. in Ubiquitous-IT from Dongseo University, Busan, South Korea. His area of interest is security in sensor network, Body area Network, and computer network.

Sang-Il Cho received his B.S. degree from Kyungwoon University, M.S. and Ph.D. degree from Dongseo University, Korea, in 2003, 2005 and 2010, respectively. In 2010, he joined part-time lecturer, Dongseo University, Korea. His current research interests are computer network security, cryptographic protocol engineering.



Dae-Seok Lee received BSc. Degrees in Computer Engineering, MSc. degree in Computer Network and Ph.D degree in WSN from Dongseo University, Korea, in 2002 and 2006. Since 2010 to now, he is research in regional innovation center for ubiquitous appliance, Dongseo University. His research interests include Ubiquitous Healthcare, Wireless Sensor Network in healthcare monitoring and System Monitoring.

Young-Dong Lee received his B.S., M.S. and Ph.D. degree from Dongseo University, Korea, in 2004, 2006 and 2009, respectively. In 2009, he joined the BK21 project team for u-healthcare technology development, Dongseo University, Korea where he is now a research professor. His current research interests are body sensor networks, vital signs monitoring, and reliability analysis for ubiquitous healthcare.



Hoon-Jae Lee received BS, MS, and PhD Degrees in electronic engineering from Kyungpook National University, Daegu, Korea in 1985, 1987, and 1998, respectively. He is currently an associate professor in the School of Computer and Information Engineering at Dongseo University. From 1987 to 1998, he was a research associate at the Agency for Defense Development (ADD). His current research interests include developing secure communication system, side-channel attack and USN/RFID security.