7

NLM-MAC: Lightweight Secure Data Communication Framework Using Authenticated Encryption in Wireless Sensor Networks

Pardeep Kumar and Hoon-Jae Lee
Dongseo University
Republic of Korea

1. Introduction

Wireless sensor networks (WSNs) are widely used intelligent technology in the century that provides user-oriented better solutions for real-time environment. WSNs have wide range of applications, such as, habitat monitoring, surveillance, location tracking, agriculture monitoring, structural monitoring, wild-life monitoring and water monitoring, are few examples (Akyildiz et al., 2002). Furthermore, numerous other applications require the fine-grain monitoring of physical environments which are subjected to critical conditions, such as, fires, toxic gas leaks and explosions. Sensors' sense the environmental data and transmit to the sink node using wireless communication, as shown in figure 1. Thus the novelty of WSNs is providing inexpensive yet effective solutions for monitoring unattended physical environments. In addition, the ubiquitous nature of WSNs makes environmental data access possible anytime, anywhere in an ad-hoc manner.

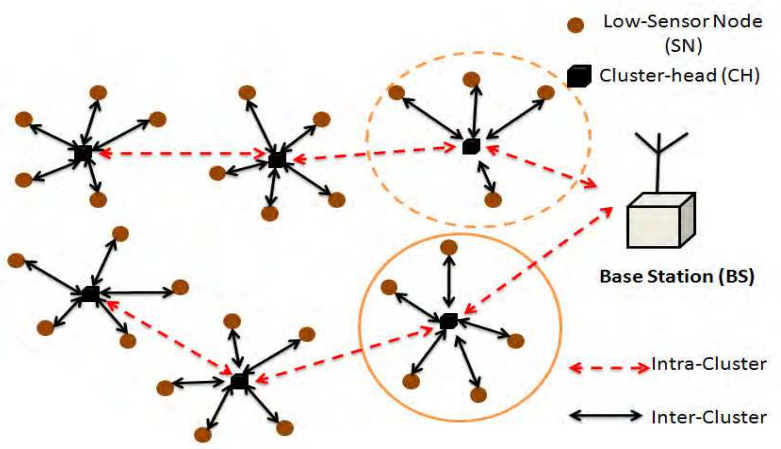


Fig. 1. Wireless sensor networks

A single node consists of on-board sensors, low computation processor, less memory, and limited wireless bandwidth. For example, a typical resource constraint node has 8 MHz microcontroller with 128 KB of read-only memory and 10 KB of program memory (Hill et al., 2000). Furthermore, a node is battery-powered (e.g., AAA batteries), thus it can operate autonomously, if needed. Therefore, a node able to collect the environmental information, processes the raw data, and communicates wirelessly with the sink. Most of WSNs are self-organized that can make self-governing decisions (i.e., turn on/off actuators) and become a part of better distributed management and control system.

The new wireless sensor technology has offered economically viable monitoring solution to many challenging applications (e.g., earthquake monitoring, military, healthcare monitoring, nuclear reactor monitoring, etc). However, deploying new technology without considering security in mind has often susceptible to attacks. As WSNs deals with real-time sensitive data that can be manipulated by any adversary for individual profit. Moreover, wireless nature of sensor node makes network more prone to the attacks. Thus security has always a big concern for wireless communication based applications. In addition, providing security to these resource constraints networks are very tedious task as compared to the resource rich networks, such as, local area networks (LANs) and wide area networks (WANs). While the WSNs security requirements are the same as conventional networks, such as confidentiality, authentication, availability, freshness and integrity. Thus security has emerged as one of the important issues in wireless sensor networks.

Significant cryptographic protocols have been introduced in order to secure the link-layer of wireless sensor networks. These cryptographic schemes are either based on block cipher (i.e., SPINS (Perrig et al., 2001), TinySec (Karlof et al., 2004), MiniSec (Luk et al., 2007)) or on public key cryptosystem (TinyPK (Watro et al., 2004)) and elliptic curve cryptography (TinyECC(Liu & Ning, 2007) and WMECC(Wang et al., 2006)). But due to the fact of limited memory and low computation of sensor nodes these protocol are still expensive in term of memory and computation. Furthermore, block cipher are always centred in cryptology, for instance, data encryption standard (DES) was considered as standard block cipher from 1974-to-2000 (Ahmad et al., 2009). Thereafter, in 2001 Advanced encryption standard (AES) was selected as standard block cipher. In fact the security of AES has been implemented in hardware for sensor nodes (e.g., telosb (Polastre et al., 2005)), and successfully implemented in software as well (Roman et al., 2007). Furthermore, in (Law et al., 2006)) and (Roman et al., 2007), some block ciphers are benchmarked on MSP430 platform and deduced the best block cipher to use in the context of WSNs. In (Roman et al., 2007) authors have surveyed public key cryptography and elliptic curve cryptography primitives for wireless sensor networks. While, the public key cryptosystem and elliptic curve cryptography are computationally expensive and time consuming for sensor networks because they need to generates and verify the digital certificates.

On other hand, stream ciphers have the simple structures, fast computations (i.e., encryption and decryption), but these ciphers are not popular in WSN security. In (Fournel et al., 2007) authors claim that the stream ciphers provide high level security services at low computation time, memory efficient, and easy to implement in software (i.e., few lines of code is required). Moreover, in 2004, the European Union started a project "named eSTREAM" ciphers aim to select a standard stream cipher that has comparable hardware and software security with efficiency (Henricksen, 2008), as AES. In (Fournel et al., 2007)

authors have presented a survey and benchmark on stream cipher for dedicated platform and deduce the well-suited stream cipher for constraints devices. Authors argue that the stream ciphers could be a better solution, and could achieves fast encryption in resource constraint network applications.

In Lim et al., 2007 and Kumar & Lee, 2009, proposed authenticated encryption which is known as Dragon-MAC[1] for wireless sensor networks. In Ahmad et al., 2009, have addressed authenticated encryption schemes, namely, HC128 –MAC, SOSEMANUK-MAC using eSTREAM ciphers for wireless sensor networks. In (Kausar & Naureen, 2009), authors have implemented and analyzed the HC-128 and Rabbit encryption schemes for pervasive computing in wireless sensor network environments. They have simulated lightweight stream ciphers (i.e., only encryption) for WSNs.

Consequently, the stream ciphers are not adequately addressed and implemented in wireless sensor networks applications. As the security services such as data authentication, confidentiality, integrity, and freshness are become critical issues in wireless sensor networks and many exiting WSN applications are lacking of the link layer security. As result, there is still research potential at link layer security that would ensure and provide security services at low cost.

In this regard, this chapter proposes a lightweight secure data framework using authenticated encryption. An NLM-128 stream cipher is used for data or packet confidentiality (Lee et al., 2009). In order to achieve the authentication and integrity services, a message authentication code (MAC) "named NLM-MAC" is incorporated into the sensor packets. The NLM-MAC ensures the message integrity and freshness of the authenticated packets. The proposed framework achieves security services at low computation cost (i.e. memory and time efficient). In order to minimize the computation cost of NLM-MAC algorithm, it is using some of the data already computed on NLM-128 stream cipher. In addition, the chapter discusses the following: (1) importance of security at the WSN link layer; (2) an adversary threat model that can be expected in WSNs; and (3) basic security requirements for wireless sensor networks. We have implemented the proposed framework on real-time test bed and our result confirms its feasibility for real-time wireless sensor applications too. In addition, we compared the proposed framework results with the existing stream ciphers that have been implemented in the resource constraints sensor networks.

The rest of chapter is structured as follows: Section 2 discusses (i) importance of security at the link layer; and (ii) an adversary threat model that can be expected in WSNs. Section 3 discusses the basic security requirements for wireless sensor networks, and Section 4 presents the related works with their weaknesses, if any. Section 5 proposed lightweight authenticated encryption framework for wireless sensor networks, and Section 6 evaluation of proposed framework in term of memory and computation time. In Section 7, conclusions are drawn for proposed authenticated encryption (NLM-MAC) and future directions are given.

2. Important of security at the link layer and adversary network model

This section discusses the importance of security at the link layer and adversary network model for wireless sensor networks.

[1]MAC is representing as message authentication code, otherwise explain.

2.1 Importance of security at the link layer

End-to-end security mechanisms are not possible in sensor network as compared to traditional computer network (e.g., SSH (Ylonen, 1996), IPSec and SSL protocols). These protocols are based on route-centric. In traditional networks, 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 an important role in energy efficient data retrieval (Karlof et al., 2004).

In contrast, for 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 of data processing operation (e.g., data compression, eliminate redundancy and so on) on incoming data packets to be routed towards to the base station. Thus, in-network processing requires intermediate nodes to access, modify, and suppress the contents of messages, if needed. Moreover, it is very unlikely that end-to-end security schemes are used between sensor nodes and base-station to guarantee the message integrity, authenticity and message confidentiality (Karlof et al., 2004). More importantly, the link-layer security architectures can easily detects 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. Hence, security is an imperative requirement at the link layer.

2.2 Adversary network model

WSNs are vulnerable to attacks due to their wireless in nature. In addition the sensor nodes are deployed in hostile or unattended environment, and are not physically protected or guarded. An adversary can directly disturb the functioning of real-time wireless sensor network applications. By applying the adversary model, he/she can handle the application accordingly for their personal benefits. For simplicity, we have divided the adversary model as follows.

- **Data monitoring and eavesdropping:** Since the sensor devices are wireless in nature, and wireless range are not confined. It may happen that an attacker easily snoops data from the wireless channels and have control on network contents, accordingly. Further, he/she may eavesdrop the network contents, such as sensor id, location and others network related information.
- **Malicious node:** An attacker can quietly place his/her malicious node into the network. By deploying malicious node into the network an attacker may control the entire wireless network or may change the route of network.
- **Data corruption:** Any message alteration from the networks, or bogus message injection into the networks could harm to the entire networks. He/she can potentially destroy the whole network and hence, network integrity compromised. Further, an adversary can replay the corrupted messages again and again, by doing so he/she can harm to the critical applications, e.g., healthcare monitoring, military and etc.

3. Security requirements for wireless sensor network at link layer

This section sketches out the important security requirements for WSNs, which are based on the above threat model and link layer requirements, as follows.

- **Confidentiality:** confidentiality, 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 sensor data should be encrypted with secret key. Moreover, the secret key is intended to recipient only, hence achieved confidentiality.
- **Authentication:** Authentication is associated to identification. Entity authentication function is important for many applications and for administrative task. Entity authentication allows verifying the data whether the data is really sent by legitimate node or not. In node-to-node communication entity authentication can be achieved through symmetric mechanism: a message authentication code (MAC) can be computed on secret shared key for all communicated data.
- **Integrity:** Message integrity, which addresses the illegal alteration of messages. To conformation of message integrity, one must have the ability to identify data manipulation by illegal parties.
- **Freshness:** In wireless sensor networks, data confidentiality and integrity are not enough if data freshness is not considered. Data freshness implies that the sensors reading are fresh or resent and thus an adversary has not replayed the old messages.

4. Related work

This section presents the related work for security protocols that have been proposed for wireless sensor networks.

Perrig et al., 2001, proposed a security protocol SPINS for wireless sensor networks. It 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 their 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., 2004, proposed another most popular wireless security architecture known as "TinySec: a link layer security architecture for wireless sensor networks". TinySec achieves low energy consumption and memory usage, and provides access control, message integrity and confidentiality. 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. Authors claims RC5 is more difficult to implement than Skipjack, so authors' selected Skipjack as the default secure block crypto algorithm. 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 is able to detect the illegal packets when they are injected into the network. One of the major drawbacks of TinySec, it does not attempt to protect from replay protection (Luk et al., 2007). The replay protection is intentionally omitted from TinySec (Luk et al., 2007).

MiniSec (Luk et al., 2007) is the first fully-carried out general function security protocol, and implanted on the Telos sensor motes. MiniSec provides two controlling modes, i.e., unicast and broadcast, and recognized as MiniSec-U, MiniSec-B, respectively. Both methods use the OCB-encryption system that allows data confidentiality and authentication. By using counter as a nonce MiniSec provides the replay protection to the sensor nodes. For more details reader may refer to the (Luk et al., 2007).

A TinyPK (Watro et al., 2004) protocol has proposed for WSN. It specifically designed for authentication and key agreement. In order to deliver secret key to the protocol, authors implemented the Diffie-Hellman key exchange algorithm. TinyPK is based on public key cryptography, which is memory consuming and time consuming for sensor networks.

Lim et al., 2007 and Kumar & Lee, 2009, proposed Dragon-MAC for wireless sensor networks. In their schemes, *encrypt-then-MAC* is used, i.e., the sensor data first encrypted and then MAC is computed over the encrypted data. Two keys are used for encryption and authentication, respectively. Authors tested their schemes for Telos B family. The main weakness of Dragon, it is not suitable for some real-time applications, such as healthcare monitoring, military, etc. Because it has 1088 bits of internal states, which are not easy to maintain for the resource hungry sensor nodes.

Zhang et al., 2008 proposed a security protocol for wireless sensor networks that exploits the RC4 based encryption cryptosystem and RC4-based hash function "called HMAC (hashed-message authentication code)" is generated for message authentication.

Ahmad et al., 2009 addressed SOSEMANUK-MAC and HC128-MAC authenticated encryption schemes using eSTREAM cipher for sensor networks. They did not provides any analytical or simulation analysis for their proposed work.

In Kausar & Naureen, 2009, authors have implemented and analyzed the HC-128 and Rabbit encryption schemes for wireless sensor networks environment. They have simulated lightweight stream ciphers (i.e., only encryption) for WSNs, but their cost of encryption schemes are very high (Kausar & Naureen, 2009). More importantly, they implemented only encryption, which is not sufficient for real-time WSN applications.

As we have seen the above, only few security schemes are well implemented and provide better security services to the WSNs. Further, many of stream ciphers are not implemented properly and provide less security services at high computation costs. So, next section present a lightweight secure framework for sensor networks that exploits the stream cipher and provides sufficient security services for WSN applications.

5. Proposed authenticated encryption framework

This section is divided into twofold: (1) introduction of NLM-128 keystream generator cryptographic protocol (Lee et al., 2009); and (2) proposed authenticated framework "named NLM-MAC" for wireless sensor networks which is based on a massage authentication code. The proposed scheme exploits the NLM-128 stream cipher based-security and facilitates the confidentiality, authenticity, integrity and freshness to the air messages.

5.1 NLM-128

A NLM-128 keystream generator proposed by Lee et al. in 2009, which is based on LM-type summation generator, and is designed with both security and efficiency in mind. It is a

combination of a linear feedback shift register (LFSR) and a nonlinear feedback shift register (NLFSR), which are easy to implement in software as well as in hardware. The length of LFSR and NLRSR is 127 bits and 129 bits, respectively. Both, LFSR and NLFSR give 258 bits of internal states to the NLM-128. Further, it takes 128 bits key-length and 128 bits initialization vector (*IV*) to fill the internal states. The simple structure of NLM-128 is shown in 2.

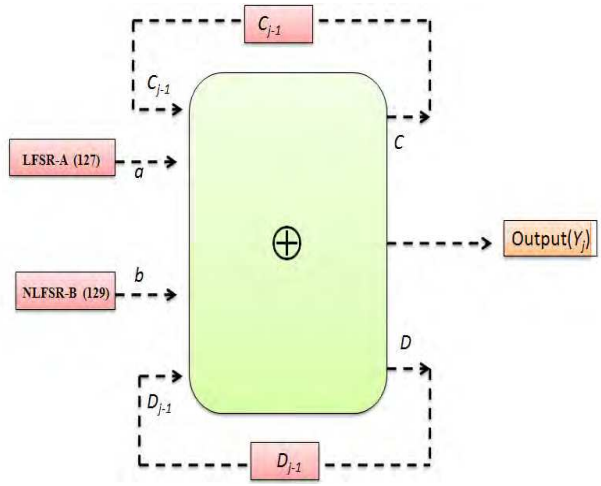


Fig. 2. NLM-128 keystream generator

5.1.1 Keystream generator

The NLM-128 generator generates the output keystream using LFSR and NLFSR sequences, a carry bit (*C*), and a memory bit (*D*). It has two polynomials: a primitive polynomial $P_a(x)$ and irreducible polynomial $P_b(x)$, as following:

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

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

The output of keystream Y_j , C_j and D_j are defined as following:

$$Y_j = (a_j \oplus b_j \oplus c_{j-1}) \oplus d_{j-1} \quad (3)$$

$$C_j = a_j b_j \oplus (a_j \oplus b_j) c_{j-1} \quad (4)$$

$$D_j = b_j \oplus (a_j \oplus b_j) d_{j-1} \quad (5)$$

5.1.2 Key loading and re-keying

Initially, 128-bits key (*key*) and 128- bits initialization vector (*IV*) together feed to 257 internal states of NLM-128. To generate the initial state for keystream generator, it uses generator itself twice, as follows.

- The initial state of LFSR-A is simply obtained by *XORing* of two 128-bits binary strings of the key (*key*) and *IV* , i.e., $LFSR\text{-}A = (Key \oplus IV) \bmod 2^{127}$.
- The initial state of 129 bits for NLFSR-B 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, i.e., $NLFSR\text{-}B = (key << 1) \oplus (0 \mid IV)$.
- Now cipher is runs second time to produce an output string of length 257-bits.

For more detailed specifications and NLM-128 security analysis, reader may refer to the (Lee et al., 2009).

5.2 Proposed authenticated encryption

A secure communication setup is needed in wireless sensor networks between two ends parties (i.e., sensor node and base station). In this regards, this subsection proposed an authentication encryption "named NLM-MAC" that setup secure communication between two ends parties and provides authentication, integrity and confidentiality, to their air messages. The proposed framework effectively utilise: (i) less space for key, and for message encryption, so that application's other functions can have enough room; and (ii) less computation, which helps to increases the network lifetime. The idea of NLM-MAC is very simple: a message authentication code (MAC) is computes over the already encrypted data (i.e., NLM-128), and hence achieve security services, as follows.

5.2.1 Data confidentiality

To achieve the confidentiality, first, NLM-128 keystream generator initialize with 128 bits key length and 128 bits of initialization vector (IV). Later, the keys and IV feed into NLM-128 internal states, which generates 128 bits output keystream, as discussed above (recall section 5.1). Thereafter, the output of NLM-128 keystream generator is *ex-or* with the plaintext that provide data confidentiality. The simplicity and small size of NLM-128 makes it well suitable to the wireless sensor network environments. For NLM-128 security analysis reader may refer to (Lee et al., 2009).

5.2.2 NLM-MAC (authentication and integrity)

A message authentication code (*MAC*) is short piece of information that used to authenticate the two end parties and verify their integrity. For instance, if a sender attached a *MAC* to the message then it must be verified at receiver end in order to manage the access control. The proposed NLM-MAC that is based on Lim et al (2007) and Kumar & Lee (2009) schemes, and offers general security services to the wireless sensor network, as discussed in the section 3. To compute MAC, considers a scenario where a sender (Alice) wants to set up a secure communication with a receiver (Bob), as follows:

- Initially Alice runs NLM-128 and encrypts the plaintext with encryption key (i.e., Key) and initialization vector (*IV*).
- Then Alice computes a *MAC* over the cipher text using MAC-Key (*i.e.*, K_{mac}), the procedure is shown in figure 3.

Pt = plaintext
Ct= Ciphertext
Key= Encryption key
K_{mac}= MAC encryption key
Ct[i]= i^{th} Ciphertext

1. $Ct = E_{key}(Pt)$
2. $\{l, m, n, p\} = K_{mac}$ *(128-bit)*
   IV=
   *[destpan||addr||type||group||count]*2*
3. $\{l, m, n, p\} = Ct[i] \oplus l, m, n, p$
4. $NLM\text{-}MAC = l \oplus m \oplus n \oplus p$
5. *Output MAC(32-bits)*

Fig. 3. NLM-MAC algorithm

- Thereafter, Alice sends *MAC*, cipher text (*Ct*) and current time (*Ta*) stamp to the receiver end (i.e. Bob).
- Upon receiving Bob the message (i.e., *MAC*, cipher text and time stamp)
- Bob first check time stamp and compare *MAC*, if both checks pass then Alice is authentic and decrypt the cipher text with *Key* and obtained the plain text.

5.2.3 NLM-MAC design

The encrypted cipher text (*Ct*) is splitting into 32-bit blocks, and then padding the last word with zeroes, if required. Meanwhile, the *MAC* encryption key (K_{mac}) is fed through variables *l, m, n, p* and then K_{mac} is *XORing* 32-bit Ct with 32-bit of *l*, and hence obtained 32-bit *MAC*.

To integrate our authenticated encryption procedure into the sensor node, we add 2 bytes counter (*ctr*) and 4-bytes *MAC* into default radio stack (TelosB), as shown in figure 4. The 2 bytes *ctr* used to achieve the semantic security and 4 byte *MAC* ensure the authentication and integrity.

Len	Fcfhi	Fcflo	Dsn	DestPAN	Add	Type	Grp	D_len	Data	CTR	MAC
1	1	1	1	2	2	1	1	1	28	2	4

Fig. 4. Modified Telos B node packet format

5.2.4 NLM-MAC analysis

Generally, the initialization vector (i.e., *IV*) must unique for encrypted packets, the unique *IV* does not give additional rooms to an attacker (Karlof, 2004). Therefore, in the proposed framework, an *IV* is taken from the packet header that is modified radio (refer figure 4) and sends to the recipient end. As shown in the figure 4, a two bytes counter (*ctr*) gives 2^{16} variations to the initialization vector (*IV*). By doing so, it guarantees that message encrypted with same key should give different cipher text every time. The four bytes *MAC* length indirectly implies the computation cost which would be needed to forge the *MAC* in chosen cipher text attack. In, (Chang et al, 2007) , (Zoltak et al., 2004) and (Karlof et al., 2004) suggested 4 bytes *MAC* gives well sufficient security, and easy to implement. Further, (Lim et al., 2007) and (Ahmad et al., 2009) suggested that the strongest definition of security for authenticated encryption can be achieved via *Encrypt-then-MAC* approach only. *Encrypt-then-MAC*: $(E_{key}, K_{mac}(Msg) = E_{key}(Msg) \,||\, K_{mac}(E_{key}(Msg))$ always gives privacy and authenticity to the air messages.

5.2.5 Operation of NLM-MAC

The operation of NLM-MAC is very simple, as follows: suppose, Alice simply computes a *MAC* on the encrypted packet with MAC key (k_{mac}) and sends *MAC* packet and cipher text to the Bob. When Bob received the *MAC* packet (i.e., authenticated packet) and cipher text, then Bob verify the *MAC* packet which is sent by Alice. If *MAC* verified then Alice is authentic and no information has been altered in transit. NLM-MAC is an *Encrypt-then-MAC* stream cipher mode (Lim et al., 2007), as shown in figure 5.

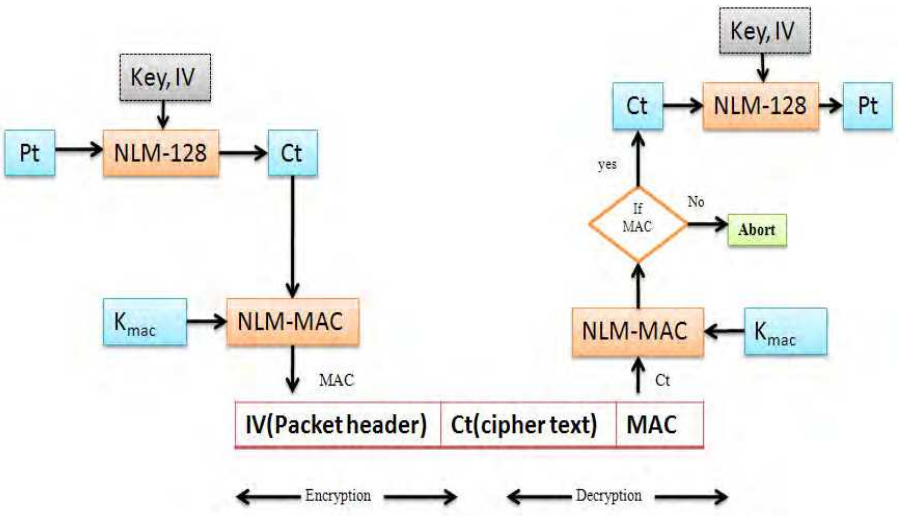


Fig. 5. Flow of NLM-MAC

6. Implementation, evaluation, and security analysis

This section discusses the implementation and evaluation of proposed framework. Further we compare and prove that the proposed scheme is efficient in term of resources consumption (i.e., memory and time efficiency) with existing schemes.

6.1 Experimental set up and implementation

In order to check the feasibility of NLM-MAC, we embedded the proposed scheme on real-time test bed, which ran on two Telos B motes and one personal computer (Intel 3.166GHz) as base station. We have implemented NLM-MAC using TinyOS, an event-driven open source operating system, which is specially designed for wireless sensor networks. The application called "secure chitchat application", and is written in NesC language. The secure chitchat application tested on Telos B sensor node that has a 16-bit, 8MHz MSP430 processor having 48 KB of programme space and 10 KB of flash memory. Further, the specifications of Telos B motes are shown in the table 1.

TelosB specification	
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.

Table 1. Telos B node specification

The experimental set up is depicted in figure 6, where sensor node 'A' acts as sender and the sensor node 'B' as receiver and vice versa. Personal computer (PC) is playing an important role as base station.

Fig. 6. Experimental set up

6.2 Evaluation

This subsection evaluates the secure chitchat application that integrated with NLM-MAC based security services. For evaluation we have considered mainly, memory and CPU execution time. As shown in table 2, our entire code uses: (i) without security 11 KB of ROM and 450 Bytes of RAM; (ii) with encryption 12.4 KB ROM (i.e., 12.4-11= 1.3KB) and 559 Bytes

RAM (i.e., 559-450 = 109 bytes); and (iii) with NLM-MAC 13.74 KB ROM (i.e., 13.74-12.4 = 1.4KB extra from encryption) and 632 Bytes RAM (i.e., 73 bytes extra from encryption). Further, the proposed scheme takes 13.35 ms time for encryption and 16.74ms for NLM-MAC operation. It is easy to see from the table 2 that the proposed scheme leaves ample space for other application's functions.

Description	ROM (BYTES)	RAM (Bytes)	Execution Time (ms)
Without security scheme	11,412	453	-
NLM-128 (Only Encryption)	12, 442	559	13. 53
NLM-MAC	13,749	632	16.74

Table 2. Occupied memory and execution time of NLM-MAC

In addition, to evaluate the simple performance of symmetric encryption and authentication (i.e. NLM-MAC) on data packets, we conducted some performance evaluation tests. As shown in the experimental set (fig 6), we simply sent 1000 data packets from sensor node A to sensor node B without any packet loss and vice versa. In order to measure the throughput of the proposed scheme, the packet size ranges from 20 bytes to 100 bytes, with an incremental of 20 bytes, as depicted in the figure 7. In only encryption case, the throughput is 23Kbps (i.e., for 20 bytes) to 25.9Kbps (for 100 bytes); and in NLM-MAC operation, it is 13.6Kbps (i.e., for 20 bytes) to 18.5Kbps (for 100bytes), which is reasonable for secure wireless sensor networks.

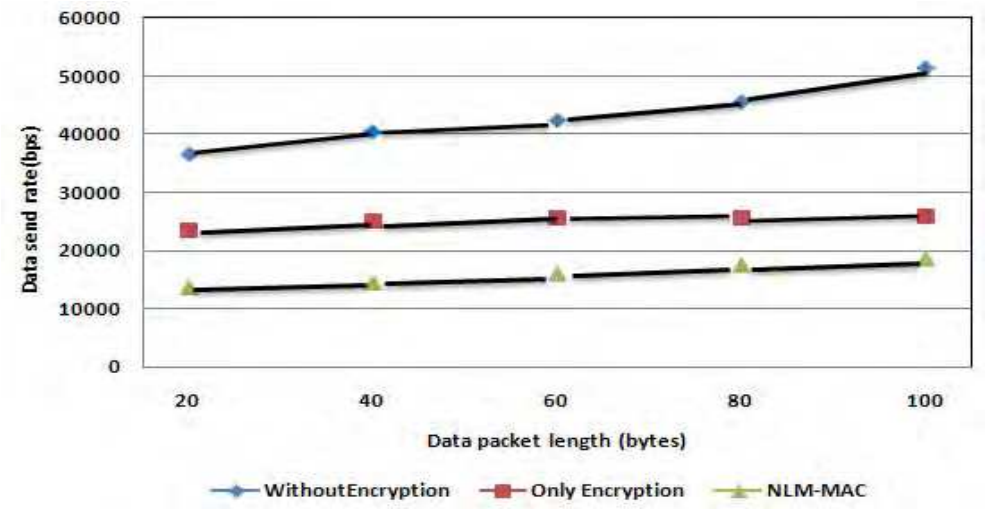


Fig. 7. Data throughput for without Encryption, only encryption, and NLM-MAC.

6.2.1 Memory and execution time comparisons with other exiting stream ciphers

This subsection compares NLM-128 with some existing stream ciphers that have been implemented or simulated in wireless sensor networks, recently. We compared the memory efficiency of proposed scheme with Lim et al.(2007), Kumar & lee (2009) and Kausar & Naureen (2009). Lim et al.(2007) and Kumar & lee (2009) have implemented Dragon stream cipher that support to the link layer security on TelosB sensor platform. Kausar & Naureen (2009) have simulated HC-128 and Rabbit stream cipher on TinyOS and TOSSIM environment for sensor networks. As shown in table 3, the encryption operation of HC-128 simulation is very expensive and it required much memory (i.e., 32.5KB of ROM and 10KB of RAM) and relatively low computation time (.049 ms). Whereas, the proposed scheme required only 12.44KB of ROM and 559bytes of RAM for message encryption, and 13.53 ms of computation time, which is practical on real-time test bed.

		Dragon encryption (Lim et al.,2007)	Dragon encryption (Kumar& Lee, 2009)	Rabbit encryption (Kausar & Naureen, 2009)	HC-128 encryption (Kausar & Naureen,2009)	**Proposed NLM-128 encryption**
M E M O R Y	Random-access memory (RAM)	18 KB	17.5 KB	14 KB	32.5KB	**12.44KB**
	Read-only memory (ROM)	964 Bytes	915 Bytes	1KB	10KB	**559 Bytes**
Execution time(ms)		17.88	16.25	.039	.049	**13. 53**

Table 3. Memory and execution time comparisons for encryption operation with other stream ciphers.

The table 4 shows the memory comparison for MAC operation. As shown in the table 4, the NLM-MAC needs only 13.7KB of ROM and 632Bytes of RAM; whereas, in (Lim et al., 2007) Dragon-MAC needs 18.9KB of ROM and 982Bytes of RAM; and in (Kumar & Lee, 2009) Dragon-MAC needs 18.13KB of ROM and 948Bytes of RAM. Moreover, NLM-MAC requires 16.74ms computation time for MAC operation, which is significantly low as compared to Lim et al., 2007 and Kumar & Lee, 2009. Whereas, in Kausar & Naureen, 2009, authors did not implemented or analyzed MAC operation, which is paramount operation in WSN security.

Consequently, it is very clear from table 3 and table 4 that the NLM-128 and NLM-MAC operations are memory efficient as compare to existing schemes.

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 5. Analytically, standard Telos radio stack packet transmission time is 2.016 ms and NLM-MAC radio stack packet transmission time is 2.208 at 250 kbps bandwidth.

		Dragon-MAC (Lim et al.,2007)	Dragon–MAC (Kumar& Lee, 2009)	**Proposed NLM-MAC**
MEM-ORY	RAM	18.9KB	18KB	**13.7KB**
	ROM	982 bytes	948 bytes	**632** bytes
Execution time(ms)		21.40	20.35	**16.74**

Table 4. Memory and execution time comparisons for MAC operation with other stream ciphers.

Description	**Pay-load (Bytes)**	**Packet Over-head (Bytes)**	**Total Size (Bytes)**	**Trans-mission time (ms)**	**Over-head inc. %**
TinySec-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	–
NLM-MAC	24	45	69	2.208	9.5

Table 5. Latency analysis

6.3 Security analysis

Based on the above experimental set up, we believe that the proposed NLM-MAC uses NLM-128 in a secure way and uses its strength and makes achieve more secure features, i.e., authentication and integrity. NLM-MAC has achieved basic requirement as discussed in section 3 and protect the air messages from an attacker, as follows.

- Data confidentiality: The proposed framework achieves NLM-128 based data confidentiality through encrypting air messages.
- Data authentication: The proposed framework facilitate data authentication through the *MAC* verification.
- Data integrity: The proposed NLM-MAC also guarantees the data integrity through data authentication verification.

Furthermore, all the operations in proposed schemes are simply uses *XOR* operations, which is cost effective.

7. Conclusions

This chapter tested the feasibility of stream cipher in sensor network where energy and computation time are important factors. We have designed NLM-MAC scheme for resource constrained devices. The proposed scheme employs on some of already computed data underlying NLM-128 cipher. The salient features of NLM-128 keystream generator 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 Telos B sensor nodes for wireless sensor networks. The implementation of its features can revolutionize the security primitives in wireless sensor networks. As conclusion, this chapter found that the lightweight stream ciphers also can be a substitute of the block ciphers. Furthermore, the remaining feature of NLM-128 can be enhanced and implemented in wireless sensor networks as per the applications scenarios.

8. References

Ahmad, S. ; Wahla, A. & Kausar, F. (2009). Authenticated Encryption in WSN using eSTREAM Ciphers, Proceeding of ISA 2009, LNCS 5576, pp. 741-749.

Akyildiz, I. F. ; Su, W. ; Sankarasubramaniam, Y. & Cayirci, E. (2002). A Survey on Sensor Networks, *IEEE Communications Magazine*, 40(8), pp. 102-114.

Henricksen, M. (2008). Tiny Dragon : An Encryption Algorithm for Wireless Sensor Networks, Proceeding of 10th IEEE International Conference on High Performance Computing and Communications (HPCC'10), Dalian, China, pp. 795-800.

Hill, J. ; Szewczyk, R. ; Woo, A. ; Hollar, S. ; Culler, D. & Pister K.(2000). System Architecture directions for networked sensors, Proceedings of ACM ASPLOS IX, pp. 93-104.

Fournel, N. ; Minier, M. & Ubeda, S. (2007). Survey and Benchmark of Stream Ciphers for Wireless sensor networks, WISTP, 2007, LNCS 4462, pp. 202-214.

KarlOff, C. ; Sastry, N. & Wagner, D.(2004). TinySec : A Link Layer Security Architecture for Wireless Sensor Networks. Proceedings of 2nd ACM Conference on Embedded Networked Sensor Systems(SenSys 2004). Baltimore, MD.

Kausar, F. & Naureen, A. (2009). A Comparative Analysis of HC-128 and Rabbit encryption schemes for pervasive computing in WSN environment, Proceeding of ISA 2009, LNCS 5576, pp. 682-691.

Kumar, P. & Lee, H. J. (2009). A secure data mechanism for ubiquitous sensor networks with Dragon cipher, Proceeding of IEEE 5th International Joint conference INC, IMS and IDC, Seoul, South Korea.

Law, Y. W. ; Doumen, J. & Hartel, P. (2006). Survey and Benchmark of Block Ciphers for Wireless Sensor Networks, ACM Transactions on Sensor Network(TOSN), pp. 65-93.

Lee, H. J. ; Sung S. M. & Kim, H. R. (2009). NLM-128, an Improved LM-Type Summation Generator with 2-Bit memories, in the Proceeding of Computer Sciences and Convergence Information Technology (ICCIT'09), Seoul, South Korea, pp. 577-582

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

Liu, A. & Ning, P. (2007). TinyECC : A Configurable Library for Elliptic Curve Cryptography in wireless Sensor Networks. North Carolina State University, Department of Computer Science, Tech. Rep. TR-2007-36.

Luk, M. ; Mezzour, G. ; Perrig, A. & Gligor, V. (2007). MiniSec : A Secure Sensor Network Communication Architecture. Proceeding of IPSN'07, Cambridge, USA.

OpenSSL. http://www.openssl.org (Accessed on 12th september 2011).

Perrig, A. ; Szewczyk, R. ; Wen, V. ; Culler, D. & Tygar, J. D. (2001). SPINS : Security protocol for sensor networks. Proceeding of 7th international conference on Mobile Computing and Networks (MOBICOM 2001), Rome, Italy.

Polastre, J. ; Szewczyk, R. & Culler, D. (2005). Telos : Enabling ultra-low power wireless research, Proceeding of Sensor Network (IPSN'2005), pp. 364- 369.

Roman, R. ; Alcaraz, C. & Lopez, J. (2007). A Survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Network Nodes, Mobile Netw Appl (2007), DOI 10.1007/s11036-007-0024-2.

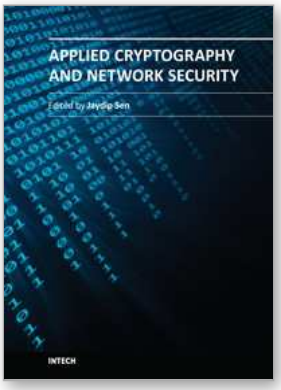
Security Architecture for the Internet Protocol. RFC2401, 1998. (Accessed on 10th september 2011).

Wang, H. ; Sheng, B. ; Tan, C. C. & Li, Q. (2007). WM-ECC : an Elliptic Curve Cryptography Suite on Sensor Motes. College of William and Mary, Department of Computer Science, Tech Rep. WM-CS-2007-11.

Watro, R. ; Kong, D. ; Cuti, S-F. ; Gardiner, C. ; Lynn, C. ; & Kruus, P. (2004). TinyPK : Securing Sensor Networks with Public Key Technology. Proceeding of Security of Ad-hoc and Sensor Networks, Washington DC, USA.

Ylonen, T. (1996). SSH-Secure Login connections over the internet, Proceeding of 6th USENIX Security Symposium, San Jose, California, 1996.

Zoltak, B. (2004). An Efficient Message Authentication Scheme for Stream Cipher, Cryptology ePrint Archieve 2004. (Accessed on 19th september 2011).



Applied Cryptography and Network Security
Edited by Dr. Jaydip Sen

ISBN 978-953-51-0218-2
Hard cover, 376 pages
Publisher InTech
Published online 14, March, 2012
Published in print edition March, 2012

Cryptography will continue to play important roles in developing of new security solutions which will be in great demand with the advent of high-speed next-generation communication systems and networks. This book discusses some of the critical security challenges faced by today's computing world and provides insights to possible mechanisms to defend against these attacks. The book contains sixteen chapters which deal with security and privacy issues in computing and communication networks, quantum cryptography and the evolutionary concepts of cryptography and their applications like chaos-based cryptography and DNA cryptography. It will be useful for researchers, engineers, graduate and doctoral students working in cryptography and security related areas. It will also be useful for faculty members of graduate schools and universities.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821