Dragon-MAC: Securing Wireless Sensor Networks with Authenticated Encryption

Shu Yun Lim[1], Chuan Chin Pu[1], Hyo Taek Lim[2], and Hoon Jae Lee[2]

[1] Graduate School of Design and Information Technology,
Dongseo University, Busan 617-716, South Korea
{shuyun, ccpu98}@dit.dongseo.ac.kr
[2] Division of Computer and Information Engineering,
Dongseo University, Busan 617-716, South Korea
{htlim, hjlee}@dongseo.ac.kr

Abstract. Sensor networks offer economically viable monitoring solutions for a wide variety of applications. In order to combat the security threats that sensor networks are exposed to, a cryptography protocol is implemented at sensor nodes for point-to-point encryption between nodes. Disclosure, disruption and deception threats can be defeated by authenticating data sources as well as encrypting data in transmission. Given that nodes have limited resources, symmetric cryptography that is proven to be efficient for low power devices is implemented. Data protection is integrated into a sensor's packet by the means of symmetric encryption with the Dragon stream cipher and incorporating the newly designed Dragon-MAC Message Authentication Code. The proposed algorithm was designed to employ some of the data already computed by the underlying Dragon stream cipher for the purpose of minimizing the computational cost of the operations required by the MAC algorithm. In view that Dragon is a word based stream cipher with a fast key stream generation, it is very suitable for a constrained environment. Our protocol regarded the entity authentication and message authentication through the implementation of authenticated encryption scheme in Telos B wireless sensor nodes.

Keywords: Authenticated encryption, Sensor network security, Dragon stream cipher, eStream project, Telos B sensor node.

1 Introduction

Recent advances in sensor network technology have open up security challenges in data transmission over wireless medium. With its wide deployment in hostile environments and security-critical applications, data security and integrity must be well taken care of. Nonetheless, security addition in a resource constrained environment with minimum overhead is a great challenge. Asymmetric encryption and digital signature are claimed to be too expensive to be feasible, even symmetric encryption protocol has to be used sparingly. Related research, TinySec [1] has created efficient link layer security protocol with an efficient block cipher and keying mechanism that is tailored for Mica2 sensor platform. Hence we are using another

approach, which is to design our security package for Telos RevB sensor platform with a symmetric stream cipher. By looking at the advantages of Dragon [2] stream cipher, we found it to be very suitable to safeguard wireless sensor networks. Dragon is very fast in key stream generation. It is faster than many of its counterparts for instance RC4 in software implementation. Besides, Dragon is able to produce throughputs of gigabits per second in both modern software and hardware. Acting as a fast key stream generator and only requiring around four kilobytes of memory, it is very appropriate for sensor nodes that have limited code size for application and security.

Besides ensuring data confidentiality, our scheme uses a message authentication code to achieve two-party authentication and data integrity. Considering that the Dragon state update function (*F* function) is integrated with a high non-linearity virtual S-box, generating a MAC from this design structure can be very competitive in practical applications. Dragon-MAC has retained the structure of Dragon stream cipher and shared its *F* function for MAC generation. This reversible mapping of 192-bit to 192-bit function is able to supply 4 bytes output that served as a MAC. We hope to introduce this security primitive and its composition method for achieving a more complete solution for sensor network security. Such approach results in the simplification of the complete authenticated encryption process, obtaining high efficiency in software implementations and, according to the further discussed analyses, ensuring a sufficient level of security.

2 Sensor Network

2.1 Sensor Networks Environments

Sensor nodes are characterized as a severely resource-constraint device in terms of energy, memory and computational ability. A typical sensor has around 2 to 10kb RAM and less than 128kb flash memory. Some sensors are operated by standard AAA batteries and their lifetime is limited. On account of its limited storage space and energy supply, lightweight modules for sensor nodes are aggressively sought. Further more, these devices are not tamper resistant and due to the distributed nature of these nodes, physical manipulation and monitoring of them made even difficult.

2.2 Sensor Network Security Primitives

The nature of broadcast communication allows interception, eavesdropping and alteration of message in sensor network broadcast medium. Sensor networks suffer from the same threats with conventional wireless networks and what's more, they are vulnerable to resource consumption attacks. This kind of attack is aiming at draining the resources like power and network bandwidth. Considering the threats that this sensor nodes are exposed to, we have to provide authenticity, integrity and privacy to secure the network environments.

Security is a broadly used term encompassing the characteristics of access control, integrity, and confidentiality. By putting control on the accessibility, only authorized nodes should be able to participate in the network. Authorized nodes are designated as those nodes that possess a shared group key. Besides, a recipient of a message needs to be assured that the message came from its stated source and a message should only be accepted if it was not altered in transit. This can help to prevent man-in-the-middle attacks where an adversary overhears, alters, and re-broadcasts messages. Lastly, all communications need to be kept private so that eavesdroppers cannot intercept, study or infer the content of messages.

Unlike the conventional network security protocol, sensor networks cannot afford to have the luxury of long key and *IV* (initialization vector) length. We carefully formatted our *IV* based on the packet header that cannot be eliminated in the transmission therefore security parameters and overhead can be tuned down. This will be discussed later in Section 3.3.

2.3 Related Works

There is relatively little work in the area of securing sensor networks than their mobile ad-hoc networks counterparts. C. Karlof et al. introduced link layer security architecture named “TinySec” for wireless sensor networks. Data transfer that relies on carrier sense, which let the nodes detect if other nodes are currently transmitting, are particularly vulnerable to DoS. Link-layer security architecture can prevent this type of attack by detecting unauthorized packets when they are first injected into the network, thus putting a stop to energy and bandwidth waste. TinySec support two different security options: Authenticated Encryption and Authentication only. In authentication encryption mode, data payload is encrypted and the entire packet is secured by a MAC. In contrast, a packet is only secured with MAC in authentication mode thereby decreases the power consumption. The authors had tested the cipher performance of two 64-bit block ciphers Skipjack and RC5 on the security architecture. Even tough RC5 outperformed Skipjack, Skipjack is selected as the default crypto algorithm for the security block since it is not patented. On the other hand, P. Ganesan et al. [3] have presented its experimental measurements to indicate cryptographic cost. RC4 is shown to have outperformed RC5 for Motes Atmega platform contrary to the choice of RC5 for the Motes project.

In addition to TinySec, University of California at Berkeley has developed a security building block, SPINS [4], consisting of SNEP to secure point-to-point communication and μTESLA for efficient broadcast authentication. SNEP is intended to achieve message confidentiality through encryption. All cryptographic primitives in this scheme are constructed out of a single block cipher for code reuse. RC5 was chosen for Atmega motes due to memory constraints. This block cipher could also serve as a hash function.

While public key cryptosystems is commonly believed to be inefficient for use on low-power devices, BBN Technologies had came out with a solution that can secure sensor networks through Public Key Technology named TinyPK [5]. TinyPK is also known as "Lightweight Security for Wireless Networks of Embedded Systems". The public-key protocols allow authentication and key agreement between a sensor

network and a third party as well as between two sensor networks. Authentication is done based on Diffie-Hellman key agreement protocol. The project has completed an implementation of RSA on the motes.

Furthermore, C. N. Zhang et. al. [6] had presented a light weight protocol based on RC4 stream cipher, which is a one-way hash function for key generation and data authentication. It has several modifications on RC4 for instance using offset and reversible states of RC4 and changing an operation to construct the one-way-hash function. This scheme is adopted to generate new keys to handle massive member join and move requests efficiently. Since only RC4 is used for data transmission, key generation and data authentication it is believed to be an ideal solution for resource limited environment.

3 Proposed Authenticated Encryption Scheme

3.1 Definition of Dragon-MAC scheme

The usual approach for securing communications in a network is to establish an end-to-end trust relationship between the sender and the receiver of a message. Concerns are placed upon entity authentication and message authentication. In the first case, the mere identity of a communication peer is verified, with message authentication codes; while in the latter case, the origin of a message and the integrity of its content are assured, with the help of data encryption. 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 Dragon *F* state update function. This feature significantly reduces the excessive program space required by the MAC algorithm.

Dragon [2] is a new word based stream cipher submitted to the European ECRYPT Stream Cipher Project. It is selected as a Focus Phase 3 candidate as of April, year 2007. The Dragon is designed with both security and efficiency in mind to satisfy the need for lightweight algorithms, dedicated to hardware environments where the available resources are heavily restricted. This stream cipher is constructed on a 1024 bits word-based NLFSR, a state update function (*F* function) and a 64 bit memory (M). It is able to generate 64-bit of pseudorandom keystream per round, operating on key sizes of 128 and 256 bits. The *F* function plays its role in both key setup and 32-bit output keystream generation. It is diffused by using a network of modular and binary addition through *G* and *H* function. This design constructed large internal states of 1024-bit and expected period of 2^{576}.

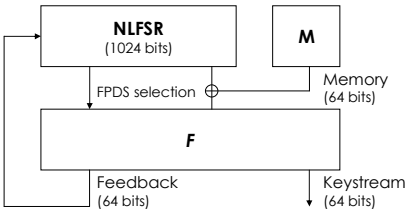

Fig. 1. Dragon structure.

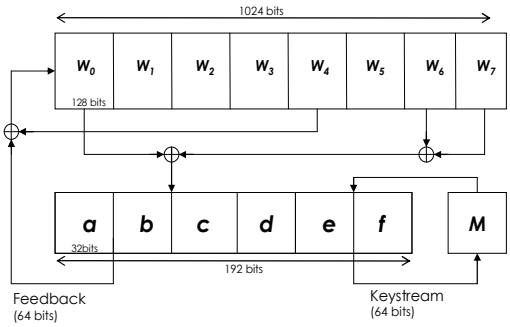

Fig. 2. Initialization.

3.2.1 Initialization

The exact specification can be referred to [2]. The 1024-bit internal state is divided into eight 128-bit words, denoted w_0 to w_7. *Key* and *IV* is feed into the internal states and going through 16 iterations of the *F* function as shown in Figure 3. The feedback of Dragon consists of four words of the *F* function outputs, means that *F* can mix *K* and *IV* effectively in a minimum number of rounds. A smaller number of rounds in key setup translate directly into high rekeying performance. This makes Dragon very competitive in practical applications that require frequent rekeying. For instance, wireless networks that usually use the frame header as *IV*.

3.2.2 Keystream Generation

Keystream generation uses the same component for initialization can simplify analysis and increases implementation efficiency. In this case the large NLFSR of 1024-bit is divided into thirty-two 32-bit words, B_i, $0<_i<31$. In each iteration 6 words from the internal states supplies input to *F* function. Of the six inputs to the *F* function, four are taken directly from the keyed internal states where two are taken from the 64-bit counter. This counter is acting as a memory in key initialization process. Each round of the keystream generation results in the output of 64-bit word, updated state *B* and memory *M*.

3.2.3 State Update Function

This component that involved in both key setup and keystream generation is a reversible mapping. It takes in 32-bit words and produces 32-bit outputs after going through the pre-mixing layer, s-box layer and post-mixing layer. Meanwhile, *G* and *H* function are constructed from two 8 x 32-bit S-boxes S_1 and S_2 to provide algebraic completeness and high non-linearity. This carefully-designed S-box lies at the heart of this state update function with low autocorrelation, best known non-linearity of 116 and optimum algebraic degree of 6 or 7.

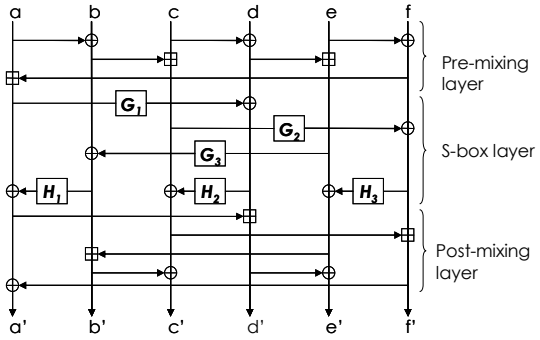

Fig. 3. State Update Function (*F Function*).

3.3 Dragon-MAC design

The scheme was designed to conform to the general security requirements discussed in Section 3.2. It makes use of the high non-linearity and low autocorrelation effect provided by S-boxes in the construction of the *F* state update function. The 8X32 S-boxes that is employed in *F* state update function have been designed heuristically to satisfy a range of important security related properties. Assuming that the underlying primitives are secure, it is possible to build a proof of a given notion of security of the MAC algorithm.

Table 1. Dragon MAC generation

Let *Pt* denote the plaintext
Let *Ct* denote the ciphertext
Let K_e denote the encryption Key
Let K_m denote the MAC encryption Key
Let *Ct[i]* denote the *i-th* 32-bit word of ciphertext

1. $Ct = E_{Ke}(Pt)$
2. $\{a, b, c, d\} = K_m$ *(128-bit)*
   $\{e, f\} = \{0x00004472, 0x61676F6E\}$
   $IV = [\ destpan \,||\, addr \,||\, type \,||\, group \,||\, ctr\ ] * 2$
3. for *i* from 16 down to 1 perform step 4
4. $\{a', b', c', d', e', f'\} = F(\ Ct[i] \oplus a, b, c, d, e, f)$
5. $Dragon\text{-}MAC = a \oplus b \oplus c \oplus d \oplus e \oplus f$
6. Output *MAC* (32-bit)

The message blocks are formed by splitting the ciphertext *Ct* encrypted under K_e into 32-bit words, then padding the last words with zeroes if necessary. Meanwhile, the MAC encryption key K_m is fed into the *F* function structure through input *a, b, c,* and *d. F* function will go through 16 clock cycles to XOR 32-bit *Ct* with 32-bit *a* and 32-bit MAC can be obtained by XORing all the outputs of *F* function. We had modified the default radio stack to incorporate our authenticated encryption scheme. We are adding 2 bytes of *ctr* to achieve semantic security and appending an additional of 4 bytes MAC for authentication. In practical applications the *IV* should be message-unique for messages encrypted with the same key so that it will not give additional advantage to an attacker. Since the *IV* is taken from the packet header of the modified radio packet format and sent in clear to the decrypting party, the 2 bytes counter *(ctr)* can give a variation of 2^{16} to the *IV*. This security property is very desirable to ensure that message encrypted with the same key should give different cipher text each time.

len	fcfhi	fcflo	dsn	destpan	addr	type	grp	data_len	data
1	1	1	1	2	2	1	1	1	28

Fig. 4. Telos B Sensor nodes radio packet format.

len	fcfhi	fcflo	dsn	destpan	addr	type	grp	data_len	ctr	data	MAC
1	1	1	1	2	2	1	1	1	2	28	4

Fig. 5. Dragon-MAC packet format.

The length of the MAC, indirectly implies the computational effort which would be required to forge the MAC in a chosen ciphertext attack. Zoltak et. al [8] and Karlof et al. [1] suggested that MAC length, d = 4 bytes provide a well sufficient security level and enable a comfortable implementation of the system. This 4 bytes MAC is practical and sufficient in the context of sensor networks.

3.4 Operation Mode

To send an authenticated packet, a sender simply computes a MAC on the packet with the agreed key K_m and encrypted message. When a node gets a packet, it can verify that the packet was sent by the corresponding node and no information has been altered in transit. Dragon-MAC is a Encrypt-then-MAC stream cipher mode. It is only defined for use with 128-bit Dragon stream cipher. Here the MAC algorithm reuses the state update function to produce a tag of the encrypted message. Bellare et. al. [9] suggested that it is possible to achieve the strongest definition of security for authenticated encryption only via the Encrypt-then-MAC approach. Compared to MAC-then-Encrypt and Encrypt-and-MAC scheme, Encrypt-then-MAC $E_{ke,Km}(M)=E_{ke}(M) \,||\, T_{km}(E_{ke}(M))$ can always provide privacy and authenticity.

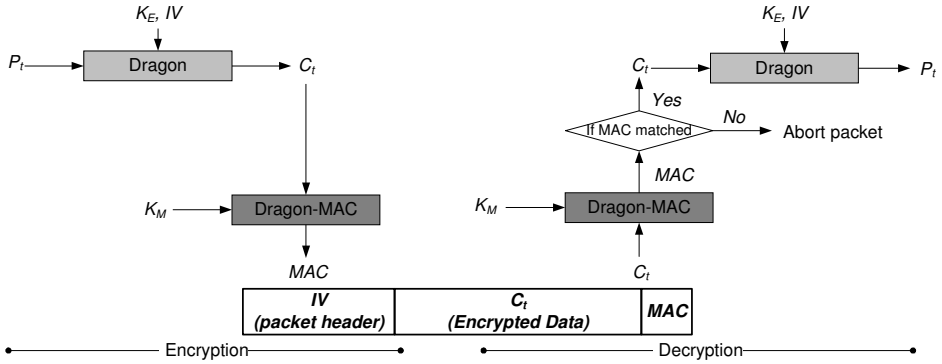

Fig. 6. Authenticated encryption process flow.

4 Implementation and Evaluation

By employing our security architecture we are able to prevent most of the depicted threats while considering the constraints that the sensor network demands. It provides confidentiality as well as integrity for the communicated information and ensures authenticity of the sensor nodes. Confidentiality is achieved by encryption the messages. This prevents any illegitimate disclosure of information; furthermore, Dragon-MAC ensures the integrity of the messages.

The implementation of our security scheme is tightly coupled with TIP700CM Telos RevB node. It is one of the popular and commercially available sensors which are marketed by Maxfor Co. Ltd. The MCU model is Texas Instrument MSP430F1611, which can operate at maximum frequency of 8 MHz but in this implementation default frequency of 4 MHz is set. Each sensor has program memory of 48Kb and 10 Kb RAM, a TI (Chipcon) CC2420 radio and USB (UART) interface. The radio transmits at 250Kbps data rate. By utilizing various sensors we can realize a Single Frequency Network (SFN) application which uses 2.4 GHz band. TIP700CM is an ultra low power wireless module for ubiquitous applications by integrating humidity, temperature and light sensors and providing flexible interconnection with peripherals.

Table 2. Telos RevB sensor nodes specification.

TIP700CM	
MCU Model	TI MSP430F1611
Type	16 Bit RISC
Program memory	48 Kbytes
RAM	10 Kbytes
EEPROM	3wire serial: 128 Bytes (1Kbits)
Radio Model	CC2420(2.4Ghz)
Data Rate	250Kbps
Sensor	Light/Temperature/Humidity
PC Interface	USB
Interface	USB (UART)

As a part of a proof of concept implementation, we had implemented Dragon stream cipher in 420 lines of nesC coding, taking up 964 bytes RAM and 18 Kbytes of program space. By doing so, this primitive had achieved a secure pair wise communication between neighbouring nodes. We had also ported the Dragon-MAC scheme on the TIP700cm sensor nodes. It helped to realize a two-party authentication and data integrity for the transmitted packet. This authentication scheme requires an additional 982 bytes of RAM and 18.9 Kbytes of ROM. Employing point-to-point security techniques is the most secure mode of operating sensor network. Several symmetric key operations are necessary at each hop so it could be a drawback if such computations are energy intensive or slow. To evaluate the execution cycles required for each component such as Dragon encryption, decryption and message authentication, a timer is used to measure the execution time. The timer is the MSP430 internal built-in timer triggered at a time. Two time-flags are allocated at the beginning and ending of the targeted component code respectively. After execution, the difference between the two records are calculated and forwarded to a PC through UART connection and displayed on monitor. Time to execute cipher operations on a 4 MHz TIP700CM sensor node marked at 17.88 ms and 21.40 ms respectively for Dragon and Dragon-MAC.

Table 3. Implementation results

Description	**ROM (bytes)**	**RAM (bytes)**	**Execution Time** (cycles/byte)
Without security scheme	13228	437	-
RC4 (Encryption only)	14372	502	12.32
Dragon (Encryption only)	18000	964	17.88
Dragon-MAC (En. & Auth.)	18900	982	21.40

Table 4. Expected latency overhead incurred

	Application Data (byte)	**Packet Overhead (byte)**	**Total size (byte)**	**Time to transmit (ms)**	**Increase over current stack**
Telos radio stack	24	39	63	2.016	-
Dragon-MAC	24	45	69	2.208	9.5%
TinyOS radio stack	24	39	63	26.2	-
TinySec-AE	24	42	68	28.3	7.9%

Increase in the packet length also lengthens the transmission time. Analytically, it requires 2.016 ms to transmit a standard radio packet and 2.208 ms to transmit a Dragon-MAC radio packet on 250 Kbps application bandwidth. Since Telos B sensor node delivers a higher data rate than Mica's node 19.2 Kbps, it generally transmit packet at a faster rate with lower power consumption. With an increase of less than 10% in transmission time, it is considered modest compared to the protection that it can offer.

5 Conclusions

This paper presents a successor to Dragon stream cipher, which is a Message-Authentication Scheme dedicated only to the Dragon stream cipher. Besides, we also demonstrated the feasibility of software-based cryptography building block in sensor networks by implementing Dragon stream cipher in real sensor test bed. Dragon is selected as the cryptography primitive owing to its fast key stream generation and secured nature. Several directions for future research arise from our solution. First, we intend to simulate our approach in other embedded sensors platform to determine the flexibility that our security scheme is able to cope with. Another interesting question is to determine how much further stream ciphers can go together with the widespread deployment of wireless sensor networks.

6 Acknowledgements

This research was supported by University IT Research Center Project of Korea and by the Program for Training of Graduate Students in Regional Innovation.

References

[1]Chris Karlof, Naveen Sastry, David 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.

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

[3]Prasanth Ganesan, Ramnath Venugopalan, Pushkin Peddabachagari, Alexander Dean, Frank Mueller and Mihail Sichitiu, "*Analyzing and Modeling Encryption Overhead for Sensor Network Nodes*" Workshop on Wireless Sensor Networks and Applications (WSNA '03) with MobiCom'03, Sep 2003.

[4]Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, 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.

[5]Ronald Watro, Derrick Kong, Sue-fen Cuti, Charles Gardiner, Charles Lynn1 and Peter Kruus, *"TinyPK: Securing Sensor Networks with Public Key Technology*", Workshop on Security of ad hoc and Sensor Networks, Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, Washington DC, USA 2004.

[6]C.N. Zhang and Q. Yu, "*An RC4 based Light Weight Secure Protocol for Sensor Networks*", from proceeding on Wireless and Optical Communication Multi Conference 2006.

[7]R. Rivest, "*The RC5 encryption algorithm*", in Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, pages 86-96, Springer-Verlag, 1995.

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

[9]M. Bellare and C. Namprempre, "*Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm*", Lecture Notes in Computer Science Vol. 1976, T. Okamoto ed, Springer-Verlag, 2000.

[10]B. Schneier, "*Applied Cryptography*", Second edition, John Wiley & Sons, 1996. National Institute of Standards and Technology, "*SKIPJACK and KEA Algorithm Specifications*", ver. 2, 29 May 1998.

[11]TIP50CM sensor node specifications. Available at http://www.maxfor.co.kr/eng/

[12]Germano Guimaraes, Eduardo Souto, Djamel Sadok, Judith Kelner, *"Evaluation of Security Mechanisms in Wireless Sensor Networks," icw*, pp. 428-433, 2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05), 2005.