IMPACTO DE MECANISMOS DE SEGURIDAD EN SENSORES IEEE 802.15.4

Carolina Tripp, Jordi Casademont Serra
Departamento de Ingeniería Telemática
Universidad Politécnica de Cataluña (UPC)
C/Jordi Girona 1-3. 08034 Barcelona, España
ctripp@entel.upc.edu

Resumen **- En la actualidad son muchos los mecanismos de seguridad que el estándar IEEE 802.15.4 permite a las redes inalámbricas de sensores [1] Dicho estándar define las especificaciones de la Capa de Acceso al Medio y la Capa Física de los dispositivos inalámbricos de área personal. La última revisión corresponde al 2006. Dichas revisiones y actualizaciones son hechas por el grupo de trabajo 802.15.**

Sin embargo estos mecanismos consumen recursos como memoria y batería, que son un poco limitados en estos dispositivos. Además de contribuir a los retardos en la comunicación. Por lo cual, en el presente trabajo se presenta de manera práctica el impacto que el uso de mecanismos de seguridad tienen en el desempeño de este tipo de redes. Para ello se hizo una comparación de dicho desempeño de manera teórica basándose en lo presentado en [2], con los valores óptimos apegados a lo especificado en el estándar IEEE 802.15.4, en contraste con pruebas reales. Para estas pruebas se hizo uso del sistema operativo TinyOS [3] y de las operaciones de seguridad MAC (Capa de Acceso al Medio) ofrecidas por el chip CC2420 usado en las motas TelosB. Además se presenta el desgaste de la batería, el cual es otro punto importante que se desea conservar en los sensores.

Palabras clave **- Sensores, TinyOS, MAC, TelosB.**

I. INTRODUCCION

Desde hace varios años ya, las comunicaciones inalámbricas pasaron a ser parte de la vida cotidiana, hasta el punto de que en la actualidad estamos completamente conectados, ya sea con nuestro móvil o cualquier otro dispositivo inalámbrico con el cual estamos en constante envío o recepción de información.

Los avances actuales en las comunicaciones inalámbricas son las que han permitido que estos dispositivos puedan desarrollarse. Gracias al estándar IEEE 802.15.4 [4] es posible su conectividad, ya que define las características de la Capa Física (PHY) y la Capa de Acceso al Medio (MAC) para los dispositivos de área personal (LR-WPAN, *Low-Rate Wireless Personal Area Networks*).

Sin embargo, este tipo de dispositivos son susceptibles, al igual que las redes inalámbricas tradicionales al ataque sobre la información transmitida. Es por ello que asegurar la información es una de las principales preocupaciones, ya que el canal de comunicación no requiere la participación física de un cable. Y debido a sus características (baja potencia, reducida capacidad de procesamiento y memoria) hacen muy difícil el uso de métodos criptográficos conocidos.

El SmartRF CC2420 [5] es un chip IEEE 802.15.4/ZigBee, operando en la banda de 2.4 GHz con velocidad de datos de 250 Kbps. Es actualmente uno de los chips más populares para trabajar en redes inalámbricas de sensores. CC2420, tiene el soporte de hardware para el formato de trama del estándar IEEE 802.15.4.

Una de las principales características del chip CC2420 es el de soportar operaciones de seguridad, como cifrado, descifrado, autenticación e integridad. Es capaz de realizar dichas operaciones a nivel MAC, entre las cuales se incluyen CTR (*Counter Mode)* que proporciona cifrado, CBC-MAC (*Cipher Block Chaining-Message Authentication Code)* el cual permite comprobar la autenticidad del emisor y la integridad del mensaje y CCM (*Counter Mode with CBC-MAC*) el cual es una combinación de los dos anteriores. Cada unas de estas, basadas en el cifrado AES [6] (*Advanced Encryption Standard*) usando claves de 128 bits.

El presente trabajo está organizado en cuatro apartados, los cuales presentan una breve descripción de la seguridad que puede usada en las motas de las pruebas, seguido por la la implementación de la misma. Continuando con la presentación de los resultados, para finalizar con la presentación de los resultados finales de las pruebas.

II. SEGURIDAD IN-LINE

CC2420 puede realizar operaciones de seguridad (cifrado, descifrado, autenticación e integridad) a nivel MAC dentro de las tramas TxFIFO (transmisión) y RxFIFO (recepción). Estas operaciones son llamadas operaciones de seguridad *In-line* [5].

Los distintos modos de trabajar son:

- ✧ Sin seguridad
- ✧ CTR (cifrado / descifrado)
- ✧ CBC – MAC (autenticación e integridad)
- ✧ CCM (autenticación, integridad y cifrado / descifrado)

III. IMPLEMENTACION

Para la realización de las pruebas prácticas, se elaboró un pequeño programa que realizaba envíos de datos de un sensor a otro, uno programado como estación base o receptor y otro como simple emisor, tomando como base un ambiente ideal, es decir solo una fuente y un receptor, por lo cual no había colisiones. Se consideró el tamaño máximo de datos de usuario en cada caso. Puesto que no se debe superar el *payload* permitido (127 bytes) ya que los sensores no realizarían ninguna operación.

Se usó la implementación de seguridad CC2420 soportada para TinyOS 2, la cual permitió la evaluación de la red con respecto a su desempeño bajo CTR (cifrado / descifrado), CBC – MAC (autenticación e integridad) y CCM (integridad, autenticación y cifrado / descifrado). En el caso de CBC y CCM pudiendo usarse distintos tamaños para el código MAC (4, 8 ó 16).

Esta implementación permite elegir entre los tres modos de seguridad, en TinyOS la función fue llamada SecAMSend y provee una interface que proporciona estas opciones, mediante los comandos:

- ✧ call CC2420Security.setCtr(*key*, *payload*);
- ✧ call CC2420Security.setCbcMac(*key*, *payload*, *MAClen*);
- ✧ call CC2420Security.setCcm(*key*, *payload*, *MAClen*);

Ellas habilitan la seguridad seleccionada antes del envío de cada paquete.

El primero ***key,*** permite al usuario seleccionar la clave, recordando que hay espacio en memoria para dos claves.

El segundo parámetro ***payload***, establece el número de bytes del *payload* que se desea no se tomen en cuenta para el cifrado (en el caso de CTR y CCM) y la autenticación (en el caso de CBC y CCM). Por defecto este parámetro es 0, ya que lo normal es iniciar luego de las cabeceras y tomar en cuenta todo el *payload* para iniciar estas operaciones.

El tercer parámetro ***MAClen***, es usado en CBC-MAC y CCM para especificar la longitud del código MAC (código de autenticación de mensaje) [7]. Los valores pueden ser elegidos entre los valores 4, 8 y 16.

Una vez indicados los valores respectivos en las funciones, se puede llamar la interfaz de envío AMSend de manera normal.

IV. RESULTADOS

Primero se comprobó que las cabeceras añadidas dependiendo del mecanismo usado fueron los correctos. En la Tabla 1 se puede ver la cantidad del *payload* ***n*** (datos enviados) y la longitud final de la trama según el mecanismos implementado.

Con los datos de la Tabla 1 podemos ver cuál es la longitud máxima (en bytes) de datos de usuario que pueden ser enviados dependiendo del tipo de seguridad que se desea implementar, pues se debe recordar que el tamaño máximo de la trama (incluyendo cabeceras) no podrá ser mayor a 127 bytes. En la tabla aparecen en negritas aquellas tramas que sobrepasan esta regla. De esta manera podemos darnos cuenta que mientras Sin Seguridad pueden enviarse 113 bytes de datos de usuario, en el caso de CCM16 el máximo sería de tan solo 92 bytes.

n	**NO SEC**	**CTR**	**CBC-MAC-4**	**CBC-MAC-8**	**CBC-MAC-16**	**CCM4**	**CCM8**	**CCM16**
60	74	79	78	82	90	83	87	95
65	79	84	83	87	95	88	92	100
70	84	89	88	92	100	93	97	105
75	89	94	93	97	105	98	102	110
80	94	99	98	102	110	103	107	115
85	99	104	103	107	115	108	112	120
90	104	109	108	112	120	113	117	125
95	109	114	113	117	125	118	122	**130**
100	114	119	118	122	**130**	123	127	**135**
105	119	124	123	127	**135**	**128**	**132**	**140**
110	124	**129**	**128**	**132**	**140**	**133**	**137**	**145**
115	**129**	**134**	**133**	**137**	**145**	**138**	**142**	**150**

Tabla 1. Longitud final de tramas.

A. *Impacto de la Seguridad en los Retardos de Transmisión*

Se hizo la programación de las motas TelosB [8], estas hacían el envío de mil tramas con diferentes longitudes de datos de usuario, desde 60 bytes hasta el máximo soportado por cada mecanismo de seguridad. Esto nos daba el tiempo total del envío de estas tramas y a partir de esos resultados se pudo promediar el tiempo en milisegundos para cada una de ellas.

En los resultados mostrados en la Tabla 2 se puede ver que la diferencia de tiempos entre enviar una trama sin cifrar y una trama cifrada (CTR) es 0.72 milisegundos, con respecto a una autenticada (CBC MAC-16) es de 1.5 milisegundos, y con respecto a una trama cifrada y autenticada (CCM16), la diferencia incrementa a 1,99 milisegundos.

Como se puede observar en las Figuras 1 y 2, la distribución en cuanto a resultados es bastante parecida, más no los rangos obtenidos. Es decir que se puede observar claramente la diferencia entre una trama sin ningún tipo de seguridad, la cual no tiene *overhead* extra ni procesos añadidos al envío en comparación con aquella a la cual se le ha implementado algún tipo de seguridad.

Pero en cuanto a los rangos de throughput se puede observar una diferencia de alrededor del 7 y 16% entre el caso teórico y el caso práctico. Esto está dado por los tiempos de *backoff*. Es decir, en el caso teórico se considera una *backoff* fijo independiente de los envíos, de los tamaños de trama y de cualquier otro factor que se presentara. Esto no fue así al realizar las pruebas reales. Ya que TinyOS está programado para que los *backoff* sean tan aleatorios como sean posibles y esto hace que los *backoff* no sean fijos. Esto provocó que el throughput tenga este decremento considerable.

Bytes	No-SEC	CTR	CBC-MAC-4	CBC-MAC-8	CBC-MAC-16	CCM-4	CCM-8	CCM-16
60	9,31	10,03	9,85	10,17	10,81	10,36	10,67	11,30
65	10,00	10,46	10,36	10,67	11,29	10,81	11,17	11,79
70	10,45	10,81	10,74	11,08	11,77	11,17	11,47	12,18
75	10,80	11,23	11,14	11,48	12,19	11,57	11,98	12,59
80	11,21	11,67	11,60	11,52	12,60	12,03	12,32	13,00
85	11,64	12,14	11,95	12,32	13,01	12,46	12,82	13,44
90	12,14	12,63	12,45	12,80	13,42	12,97	13,31	13,97

Tabla 2. Retardos en los envíos en mseg.

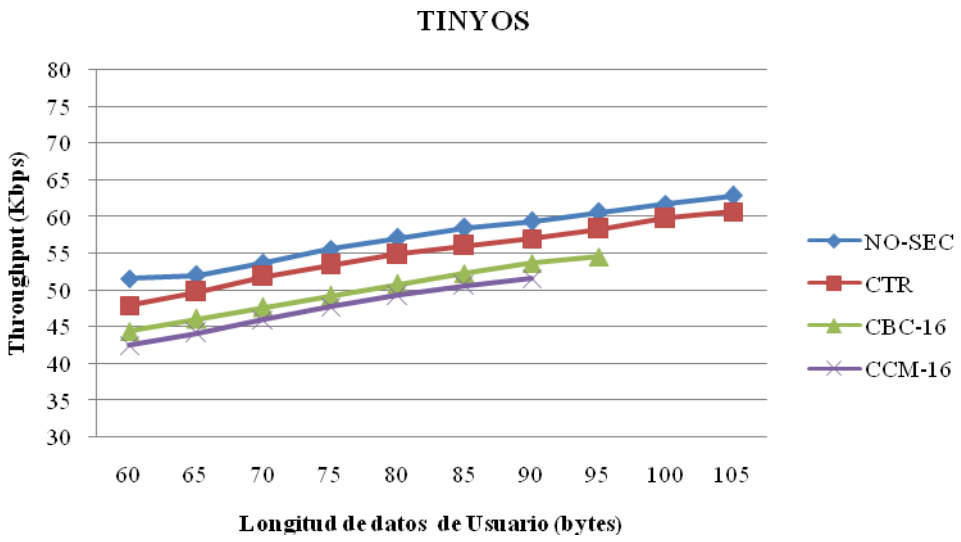

Fig. 1. Throughput efectivo, caso práctico (TinyOS). NO ACK.

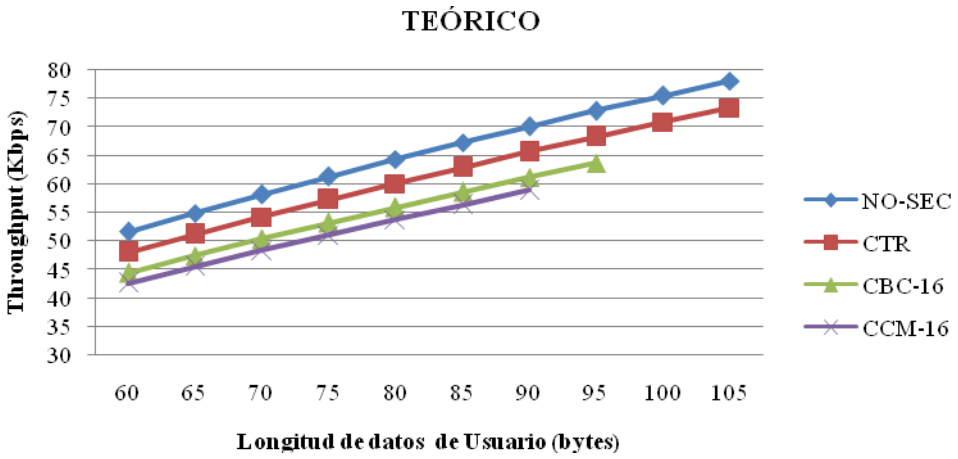

Fig. 2. Throughput efectivo, caso teórico. NO ACK.

B. Impacto de la Seguridad en el Consumo de Energía

En el caso de la energía requerida, se hizo uso de un analizador de potencia para poder observar el comportamiento de consumo de la carga al momento que los sensores están transmitiendo o recibiendo información. Esto haciendo diferentes lecturas, se inició haciendo pruebas cuando los sensores trabajan sin ningún tipo de seguridad, en decir solo enviando datos a una estación base.

Este analizador funciona como una fuente de alimentación de energía, que permite la medición de voltaje e intensidad. Las mediciones se hicieron tanto en el nodo transmisor como aquel programado como Estación Base, el cual recibe toda la información que el emisor está generando.

Se realizó una prueba con las motas programados para envíos de tramas de 90 bytes de usuario sin seguridad, pasando a la misma cantidad de envíos con la misma longitud de datos de usuario pero bajo la implementación de CCM16 y finalizando con un tiempo en reposo, es decir, sin actividad alguna.

Como resultado de las lecturas con el analizador se pudo obtener que como media, la corriente usada para transmitir paquetes sin cifrar es de 24 mA y en recepción 26 mA. En el caso del envío de datos bajo la seguridad CCM16 la media en los envíos es de 22 mA y de igual manera que en el caso anterior 26mA en recepción. Esto sin hacer uso de ACK y aplicando un voltaje de 3 V. En la Figura 3, se puede distinguir claramente como en los envíos sin seguridad hay una carga de 24 mA, cuando pasa a CCM16 hay una disminución de 2 mA y en el caso de reposo baja hasta 4.7 mA, en el caso de transmisión.

Estos resultados se deben a que en el caso de no presentar el añadido de ningún método de cifrado o seguridad, la carga es constante y el envío se hace con mayor rapidez, requiriendo una energía menor que en el caso contrario.

En el caso de recepción, mientras no recibe ningún paquete se encuentra en un nivel de carga de 24 mA en promedio. Esto porque siempre está en estado de espera, es decir en un constante monitoreo de recibir información, por ello su descenso no es mayor. Esto se puede observar claramente en la Figura 4, en la cual se observa constante la recepción de datos en 26 mA y el pequeño descenso en caso de no recibir nada.

Esto demostró que aunque la carta necesaria en mayor en caso de no usar seguridad, como los envios se hacen en un tiempo menor la energía consumida es menor que en el caso de usar seguridad CCM16.

En la Tabla 3 se resumen el desgaste de energía, el cual depende de los tiempos de envío que se necesita para cada caso, esto se obtuvo anteriormente con el analizador. Se pudo comprobar cómo la recepción con seguridad CCM16 gasta más energía, esto se debe a que en caso de implementar un tipo de seguridad los tiempos de envíos y recepción son mayores.

Una batería convencional usada en las motas, tiene una capacidad de 2000mA·hora, y tomando en cuenta los datos obtenidos anteriormente en cuando a la transmisión de tramas y cargas correspondientes, se puede llegar a la conclusión de que la cantidad de envíos dependiendo la longitud de datos de usuario, lo cual se muestra en la Tabla 4.

Esto es multiplicando el tiempo de transmisión por la carga necesaria, en este caso 22 o 24 mA según corresponda. El valor obtenido esta dado en mA·seg, entonces a continuación se pasa a mA·Hora y se divide entre los 2000 que soporta la batería.

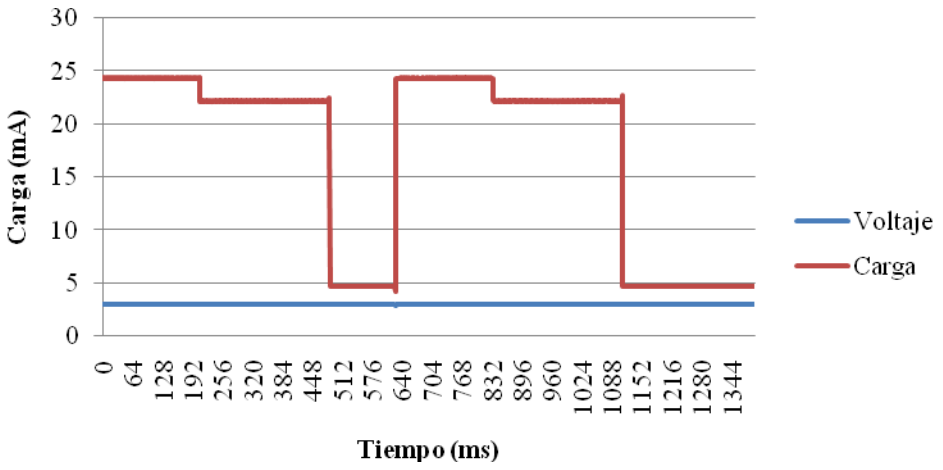

Fig. 3. Transmisión de paquetes Sin Seguridad y CCM16.

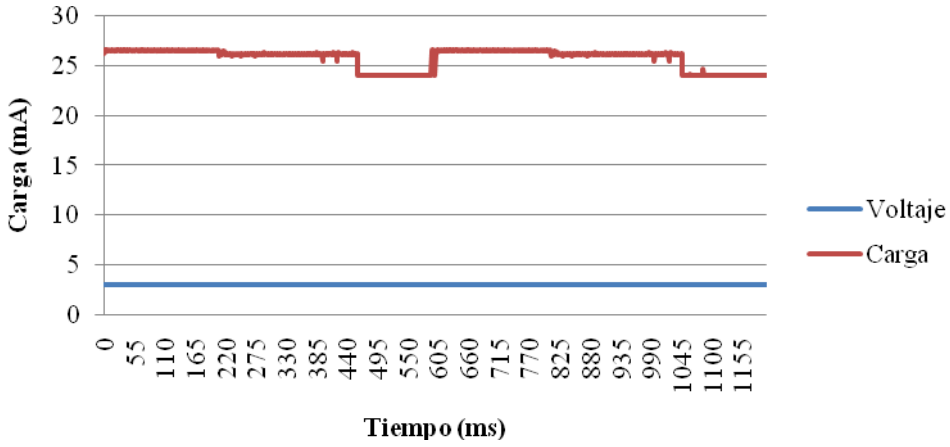

Fig. 4. Recepción de paquetes Sin Seguridad y CCM16.

	NOSEC			CCM16		
bytes	**Tiempo (t)**	**Energía (mJ) TX**	**Energía (mJ) RX**	**Tiempo (t)**	**Energía (mJ) TX**	**Energía (mJ) RX**
60	0,0093	0,6696	0,7254	0,0113	0,7458	0,8814
65	0,0100	0,72	0,78	0,0118	0,7788	0,9204
70	0,0104	0,7488	0,8112	0,0122	0,8052	0,9516
75	0,0108	0,7776	0,8424	0,0126	0,8316	0,9828
80	0,0112	0,8064	0,8736	0,0130	0,858	1,014
85	0,0116	0,8352	0,9048	0,0134	0,8844	1,0452
90	0,0121	0,8712	0,9438	0,0140	0,924	1,092
95	0,0126	0,9072	0,9828	--	--	--
100	0,0130	0,936	1,014	--	--	--
105	0,0134	0,9648	1,0452	--	--	--

Tabla 3. Energía requería en transmisión y recepción.

	Tramas (x 10^6)	
bytes	**NOSEC**	**CCM16**
60	116	98
65	109	93
70	103	89
75	97	85
80	92	81
85	87	78
90	83	75

Tabla 4. Cantidad de tramas transmitidas con una batería AA.

V. CONCLUSIONES

Luego del análisis de los resultados de las pruebas se vio la importancia de tomar en cuentas todas las restricciones en este tipo de redes de comunicación al momento de proporcionar seguridad en ellas, puesto que sus características son muy diferentes a las redes inalámbricas tradicionales, por lo cual se deben tener consideraciones especiales al momento de implementar la seguridad, como el desgaste de la batería, el bajo alcance, poca memoria disponible, etc.

Aún con estas restricciones se sabe que no es imposible la implementación de seguridad; como se mencionó existen varias maneras de ofrecer seguridad dependiendo el tipo de sensor a utilizar, ya que no todos soportan las mismas funcionalidades.

Finalmente pudimos comprobar la carga que necesitan los sensores para los envíos, tanto en recepción como en transmisión. A partir de esto obtener y presentar que en el caso del uso de seguridad se tiene un mayor desgaste de energía el cual genera un mayor consumo de batería. Esto relacionado con los tiempos de envío necesarios.

AGRADECIMIENTOS

Este trabajo ha sido elaborado con el apoyo del CDTI, Ministerio de Industria, a través del proyecto Segur@ y a la Universidad Autónoma de Sinaloa.

REFERENCIAS

[1]. Healy, M, Newe, T y Lewis, E. Efficiently securing data on wireless sensor network. *SENSOR07*. 2007.

[2]. Gomez, Carles, Casademont, Jordi y Paradells, Josep. Theoretical Study on the Impact of Security Mechanisms on Performance of IEEE 802.15.4 and ZigBee higher layers. Barcelona : 2008.

[3]. Portal TinyOS Comunity. [En línea] http://www.tinyos.net/.

[4]. Society, IEEE Computer. Wireless Mediunm Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Network (WPAN's). New York : 2066.

[5]. Chipcon. [En línea] http://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf.

[6]. Rodríguez Henríquez, Francisco, y otros. Cryptographic Algorithms on Reconfigurable Hardware. : Springer, 2006.

[7]. López Trejo, Emmanuel. Implementación Eficiente en FPGA del Modo CCM usando AES. México, DF. : Septiembre 2005.

[8]. Crossbow. [En línea] http://www.xbow.com/Productos/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.