Gestión eficiente de permisos en redes de sensores inalámbricos

E. Mercadal, A. Freijó, G. Navarro-Arribas, J. Borrell
DEIC, Departamento de Ingeniería de la Información y de las Comunicaciones - UAB, Universitat Autònoma de Barcelona. Bellaterra (Spain)
Email: {emercadal,afreijo,gnavarro,jborrell}@deic.uab.cat

***Resumen*—Las redes de sensores inalámbricos de bajo consumo son una tecnología en auge estos últimos años. La poca capacidad de cálculo de los nodos que la componen, así como sus altas restricciones en uso de energía, condicionan en gran medida la aplicación de técnicas criptográficas comunes en entornos no restringidos. En este artículo presentamos un método basado en funciones *hash*, con el que se limita el acceso a cada nodo de una WSN solo a dispositivos autorizados, y se ejemplifica en un escenario de emergencias médicas donde los sensores inalámbricos crean redes de víctimas.**

I. Introducción

La seguridad, tanto de los datos como de acceso a la propia red, en las redes de sensores inalámbricos (WSN - *Wireless Sensor Networks*) de bajo consumo, puede ser, en ocasiones, una característica a tener en cuenta en su despliegue. La baja potencia de cálculo de los nodos en este tipo de redes hace que las técnicas habituales para protegerlas tengan que ser revisadas y, en muchos casos desechadas, antes de poder ser aplicadas.

A pesar de estos inconvenientes es deseable y recomendable proteger las WSN desplegadas contra accesos indeseados a su información, tanto en su lectura como en su modificación y creación.

Este tipo de redes se puede aprovechar de los agentes móviles, facilitando la reprogramación de nodos o de la propia red (p.e. esquemas de encaminamiento). Se envía un agente con el nuevo código que se copia a los nodos deseados, éste código se utiliza para tareas como seguir un evento a través de la red o devolver los datos recogidos a la estación base.

Hasta ahora, los agentes se han usado en las WSN principalmente:

- Como simples contenedores de mensajes enviados desde el sensor hasta la estación base. Por ejemplo en la monitorización de sensores remotos de MAPS [1] o en la detección de fuegos de Agilla [6].
- De acuerdo a un patrón de dos fases: una propagación inicial desde la estación base a la WSN, seguida de una fase de vuelta a la estación base. Ejemplos son la recogida de datos globales de Agent Framework [18]; la búsqueda en gradiente de ActorNet y Agent Framework [9], [18]; la recolección de datos en In-Motes [8], o la monitorización de contenedores de carga en Agilla [6].
- Para monitorizar y seguir eventos locales. Por ejemplo, el seguimiento de eventos de Agent Framework [18], o el seguimiento de incendios y navegación en entornos dinámicos de Agilla [6].

En [13] se presenta un nuevo caso de uso para las WSN en el que se utilizan los sensores disponibles en los nodos para monitorizar el estado de las víctimas de catástrofes con un gran número de víctimas (MCI - *Mass Casualty Incident*). Todo el proceso de monitorización se realiza sin tener en cuenta aspectos de seguridad ni privacidad de los datos transmitidos o almacenados por los participantes en la gestión de la catástrofe.

Parece claro pues, que la protección de los datos recogidos por los sensores que contienen información sobre el estado de salud de cada una de las víctimas es necesaria. Los sistemas generales de protección de agentes utilizados en sistemas multi-agente sin restricciones de cómputo, espacio o energía, son en muchos casos difícilmente portables, al menos en su forma actual, a las altamente restringidas WSNs.

El cifrado de los datos recogidos por los sensores en el propio nodo no parece una solución viable, pues el tiempo requerido para el cifrado, junto con la cantidad de batería consumida limitan muchísimo más el sistema haciéndolo poco útil en la práctica, y poco fiable. Hay que tener en cuenta que cada nodo está alimentado por dos pilas de 1.5V y dispone de un procesador MSP430F1611 de 16 bits [15]. Así pues, propuestas como [2], [10] quedan desechadas por orientarse a soluciones de este tipo.

No siendo una solución adecuada cifrar los datos recogidos por los sensores se debe, al menos, limitar su acceso a todo dispositivo ajeno a la aplicación. Esto es, solo permitir su lectura a dispositivos previamente autorizados en el centro de control de emergencias (ECC - *Emergency Control Center*) y, más importante, evitar que los datos guardados en los nodos puedan ser falseados por terceros. Asegurar la integridad de los datos recogidos puede ser, en escenarios de emergencias, más importante que evitar que sean leídos por usuarios no autorizados.

En este artículo presentamos el uso de estructuras basadas en funciones hash, como los filtros de Bloom, para proporcionar un esquema de autorización sencillo y eficiente. El objetivo de este trabajo es mostrar la aplicabilidad de este tipo de esquemas en las redes de sensores.

El artículo esta estructurado de la siguiente manera: primero describimos brevemente la codificación de permisos usando filtros de Bloom (Sección II), a continuación discutimos el escenario de aplicación de estos permisos (Sección III) y

acabamos con una breve conclusión del trabajo expuesto (Sección IV).

II. Codificando permisos en filtros de Bloom

En este trabajo nos basamos en el uso de los filtros de Bloom [3] para codificar de manera eficiente los permisos del sistema. Esta idea está inspirada en los sistemas de micropago [16], [14], donde se utilizan cadenas hash para representar monedas. En estos sistemas el conocimiento de un valor de la cadena hash, permite a un usuario usar la moneda correspondiente.

Estos esquemas también se han utilizado para representar *tokens* de acceso o autorizaciones en sistemas de control de acceso o gestión de confianza (*trust management*) con soporte a la delegación [7]. El principal problema de estos esquemas es que no pueden trabajar con permisos estructurados como órdenes parciales, sino que están limitados a secuencias de permisos o *tokens*, que solo pueden formar una orden completa. En muchos escenarios no es difícil encontrar permisos estructurados como órdenes parciales o retículos.

De manera rápida definimos un filtro de Bloom como un conjunto de n elementos, representado por un vector de bits $B = B_1, \ldots, B_m$ inicializado a 0. Utiliza un familia de funciones hash unidireccionales f_i para $i = 1, \ldots, k$, con una imagen uniformemente distribuida en el intervalo $[1, m]$. Para añadir un elemento x en el filtro (conjunto), tenemos que calcular los valores $x_i = f_i(x)$ para $i = 1, \ldots, k$, fijar el conjunto de bits $B_{x_i} = 1$ en B. Para comprobar si un elemento y se encuentra en el filtro, simplemente tenemos que verificar si los bits B_{y_i} son igual a 1.

En general los filtros de Bloom presentan algunas características interesantes desde el punto de vista de la seguridad. Por ejemplo, dado un valor es fácil y rápido decidir si un elemento esta presente en el filtro, pero dado un filtro puede no ser sencillo decir que valores contiene.

II-A. Permisos

En general consideramos $(P, \geq)$ como un retículo de permisos P con un supremo denotado como $\top$ y un ínfimo denotado como $\bot$. Dados $x, y \in P$ la relación de orden $x \leq y$ se interpreta con el siguiente significado: si un usuario tiene el permiso y entonces también tiene de manera implícita el permiso x.

El filtro de Bloom $\mathcal{B}$ forma un retículo con relación de orden parcial $\sqsubseteq$ (subconjunto) que es isomorfo al retículo de permisos $(P, \leq)$ utilizando el mapeo $\mathcal{B}(a)$ para valores $a \in P$. Es decir, tenemos que para $a, b \in P$, :

$$a \leq b \Leftrightarrow \mathcal{B}(a) \sqsubseteq \mathcal{B}(b)$$

El retículo $(P, \leq)$ también es isomorfo al conjunto potencia con la relación de orden $\subseteq$ con el mapeo $\lceil a \rceil = \{x \in P \mid a \leq x\}$, es decir, tenemos que para $a, b \in P$ entonces [4]:

$$a \leq b \Leftrightarrow \lceil a \rceil \supseteq \lceil b \rceil$$

y por lo tanto,

$$a \leq b \Leftrightarrow \mathcal{B}(\lceil a \rceil) \sqsupseteq \mathcal{B}(\lceil b \rceil)$$

Denotamos como $(\mathbf{B}, \sqsupseteq)$ este retículo isomorfo, donde cada elemento en $\mathbf{B}$ es un filtro de Bloom que contiene el correspondiente conjunto del conjunto potencia de $(P, \leq)$. Un filtro de Bloom que pertenece a $\mathbf{B}$ lo llamamos *permiso de Bloom*.

II-B. Uso de permisos de Bloom

El administrador o autoridad administrativa del sistema y por tanto propietario de la política $(P, \leq)$ genera un conjunto de permisos de Bloom de la forma $\mathcal{B}(\lceil a \rceil)$ para cada $a \in P$.

El permiso supremo $\top \in P$ es secreto y solo lo conoce la autoridad administrativa de la política $(P, \leq)$; el resto de permisos $P \setminus \{\top\}$ se consideran públicos. Cabe remarcar que si $\top$ es secreto, $\mathcal{B}(\{\top\})$ también es secreto.

II-B1. Delegación de permisos: Si un participante posee el permiso de Bloom $\mathcal{B}(\lceil a \rceil)$, para poder delegar el permiso $x \in P$, éste participante debe calcular $\mathcal{B}(\lceil a \rceil) \sqcup \mathcal{B}(\lceil x \rceil \setminus \{\top\})$[1].

II-B2. Verificación de permisos: Un participante presenta un permiso de Bloom X que representa un permiso a la autoridad administrativa (o verificador) para realizar una acción que requiere el permiso b según la política del sistema. Esta petición se autoriza si y solo si:

$$\mathcal{B}(\lceil b \rceil) = X$$

Este test asegura que el participante que hace la petición posee el permiso adecuado y que éste incluye el secreto $\top$.

En el caso de que el participante posea otro permiso $a \in P$ tal que $b \leq a$, éste puede calcular fácilmente $\mathcal{B}(\lceil b \rceil)$ a partir de $\mathcal{B}(\lceil a \rceil)$, ya que $\mathcal{B}(\lceil b \rceil) = \mathcal{B}(\lceil a \rceil) \sqcup (\mathcal{B}(\lceil b \rceil) \setminus \mathcal{B}(\{\top\}))$.

II-C. Comentarios sobre la seguridad de los permisos de Bloom

Queda por ver que realmente estos permisos de Bloom proporcionan un mínimo grado de seguridad, es decir que no pueden ser falseados ni el sistema puede sufrir abusos con facilidad.

Por otra parte hay que tener en cuenta que los filtros de Bloom puede dar falsos positivos. Es importante tener en cuenta que este tipo de permisos no busca proporcionar una seguridad máxima, y esta orientado a aplicaciones donde la eficiencia puede ser más importante que la seguridad. Por ello, cuando decimos que es fácil y rápido tomar una decisión de autorización, nos referimos exclusivamente al aspecto computacional. Es decir es un esquema eficiente pero su fiabilidad (falta de errores) siempre estará acotada por la probabilidad de falsos positivos que presente el filtro de Bloom.

Actualmente estamos trabajando en proporcionar a los permisos de Bloom de una base teórica y empírica sobre su seguridad. Sin embargo se pueden observar algunas propiedades interesantes.

En general podemos observar que no es factible falsear $\mathcal{B}(\{\top\})$ sin el conocimiento de $\top$ ni del conjunto de permisos P y su respectiva codificación en filtros de Bloom.

En este caso un participante necesita adivinar la posición de los k bits correspondientes al elemento $\top$ en el vector del

[1] Se desconoce el secreto $\top$ por lo que solo se puede calcular $\lceil x \rceil \setminus \{\top\}$.

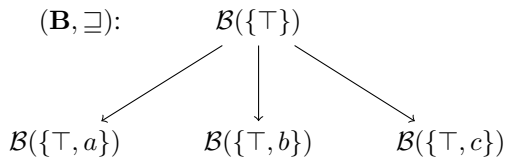

Figura 1. Ejemplo de permisos de Bloom.

filtro. Asumiendo que estos k bits se seleccionan aleatoriamente, hay $\binom{m+k-1}{k}$ combinaciones posibles (un bit puede ser seleccionado más de una vez por el mismo elemento). De esta manera la probabilidad de que un participante falsifique $\mathcal{B}(\{\top\})$ sin el conocimiento de más información del sistema es $P_0 = \binom{m+k-1}{k}^{-1}$.

Sin embargo el caso anteriormente mencionado es poco probable, ya que aunque un participante no conozca el secreto $\top$ sí que puede conocer más información sobre los permisos de Bloom que posee.

Un ejemplo más extremo sería el que muestra la Figura 1. En este caso podemos suponer que un usuario posee los permisos a y b, es decir conoce $\mathcal{B}(\{\top, a\})$ y $\mathcal{B}(\{\top, b\})$. Este usuario no debería poder conocer $\mathcal{B}(\{\top\})$ y consecuentemente $\mathcal{B}(\{\top, c\}$.

Esto se puede cumplir ya que la intersección de filtros de Bloom no es posible en el caso general. La intersección de dos filtros de Bloom se podría calcular haciendo un AND bit a bit entre los dos filtros. Sin embargo no es cierto que si $A \cap B = C \Rightarrow \mathcal{B}(A)$ AND $\mathcal{B}(B) = \mathcal{B}(C)$. Cabe remarcar que la unión de filtros de Bloom, al contrario que la intersección, sí que se puede calcular mediante una operación OR bit a bit.

En este artículo no entramos en detalle en describir la seguridad y parametrización necesaria de los permisos de Bloom, ya que nos centramos en una propuesta de su aplicación.

III. Escenario y aplicación

En catástrofes con un gran número de víctimas, la organización de los equipos médicos implicados en el salvamento de los heridos es crucial. Así como también lo es la presteza y prioridad con la que se tratan las víctimas.

El área de trabajo en una catástrofe de este tipo se divide en diferentes zonas dependiendo de su situación o función en la emergencia. Así, la **zona 0** es aquella donde ha ocurrido la emergencia. Es, entonces, donde están las víctimas y heridos de la catástrofe. La **zona 1** es donde se encuentran todos los efectivos médicos desplazados hasta la emergencia. También en esta zona se habilita el hospital de campaña y se prepara el personal de rescate. Finalmente también existe una **zona 2** que se compone de todos los efectivos médicos disponibles alrededor del globo para colaborar con los efectivos desplazados.

Desde el centro de control de emergencias (ECC) situado en la zona 1 se coordinan todos los efectivos para el correcto tratamiento y evacuación de las víctimas afectadas. Los miembros del equipo de clasificación recorren, aleatoriamente, el área afectada buscando heridos y etiquetándolos siguiendo un protocolo determinado, p.ej. START [17], pero no los tratan. Al mismo tiempo diversos equipos médicos especializados, equipados con material sanitario, buscan víctimas ya clasificadas y las tratan. Bien en el lugar donde se encuentran, o bien en una ambulancia, trasladándolas hasta el hospital de campaña, dependiendo de la gravedad de las heridas. En el caso de que el ECC tenga conocimiento de una zona con víctimas potencialmente graves puede ordenar el envió directo de un equipo hasta la zona afectada, agilizando su tratamiento y evacuación.

Para su clasificación se mide tanto la respuesta de la víctima a estímulos externos, p.ej. respuesta a órdenes simples, como la estabilidad de sus signos vitales, p.ej. pulso y respiración. Al finalizar la clasificación, cada víctima recibe una etiqueta identificando su estado de salud con un color, determinado por el tipo de clasificación utilizada. En START por ejemplo se usan 4 colores, verde, amarillo, rojo y negro, siendo verde heridas leves y negro fallecimiento.

Este método se mejoró en 2009 [11] donde el personal médico, tanto de clasificación como de tratamiento, se equipaba con unos dispositivos de mano. Con ellos se realizaba la clasificación con identificación por colores, pero se añadía la posición GPS de la víctima y se equipaba a la víctima, además de con la etiqueta física, con un RFID con la información de la clasificación.

Además, cada clasificación generaba un agente móvil en el dispositivo de mano. Éste, migrando de dispositivo de mano a dispositivo de mano utilizando como parámetro de decisión el tiempo que le faltaba para regresar al ECC (TTR - Time To Return), enviaba los datos de la víctima hasta el ECC. Desde allí se podía enviar un equipo médico directamente a una determinada víctima para agilizar su tratamiento, evitando así la búsqueda ciega de las víctimas más graves.

Finalmente, en [13] se añaden nodos sensores a la clasificación. Cada una de las víctimas tiene un nodo inalámbrico de sensores con el que se monitorizan las constantes de la víctima mediante un agente. Dependiendo de la proximidad de las víctimas, los nodos pueden crear, siempre siendo decisión del personal de clasificación, una red de sensores que será recorrida por un segundo agente, esta vez móvil. El recorrido que realiza el segundo agente es calculado *in-situ* por un dispositivo de mano transportado por el personal de clasificación [12]. Este segundo agente se encarga de recoger los resúmenes de los estados de las víctimas calculados por cada uno de los sensores y propagar estos resúmenes a cada uno de los nodos de la red, manteniendo así un estado general de toda la red en cada uno de los nodos (Figura 3). El dispositivo de mano del personal médico se adapta también para ser usado en la nueva aplicación, añadiéndole un nodo sensor que puede conectarse con los nodos de las víctimas (Figura 2).

En los nodos inalámbricos se utiliza una plataforma de agentes móviles adaptada a este tipo de dispositivos, Agilla [6]. Se usan agentes de esta plataforma para recorrer toda la WSN, leyendo los datos de los nodos y actualizando el estado de las víctimas en cada uno de los nodos. Esta tecnología además proporciona un método fácil de reprogramación de la WSN, permitiendo lanzar un nuevo agente, encargado de realizar otra tarea, a la red sin necesidad de reprogramar todos los nodos.

El agente encargado de recorrer la WSN se equipa con

Figura 2. Dispositivo de mano con nodo sensor del personal médico

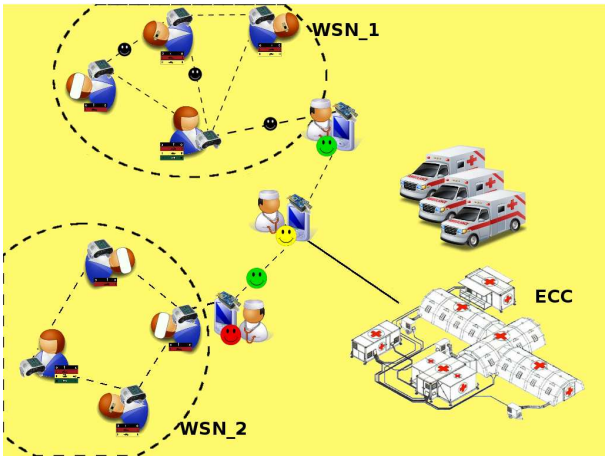

Figura 3. Aplicación utilizada en un escenario de emergencia

un itinerario precalculado [12] para realizar la tarea eficientemente en términos energéticos. Esto alarga el tiempo de vida de los sensores inalámbricos que los alojan. Se ha previsto también que este itinerario será fácilmente partido, ya sea por el movimiento de alguna de las víctimas o por el fallo de uno de los nodos. Así se ha equipado al agente con diferentes tipos de tolerancia a fallos que le permiten escoger un nodo alternativo en caso que el marcado en el itinerario no estuviese disponible.

Si durante el tiempo de funcionamiento de la WSN, uno de los sensores entra en contacto con algún dispositivo de mano del personal médico, el agente residente en ese nodo, que monitoriza la víctima, es el encargado de contactar con el dispositivo de mano y enviarle el resumen de la WSN. Se pueden encontrar más detalles en [12].

Como se hacía previamente, el estado de las víctimas es enviado, tras el clasificación, al ECC usando el propio personal médico como nodos [11]. En [12] además, se mantiene el estado disponible en el ECC actualizado. Para ello se utilizan los resúmenes de estado generados por los nodos sensores y propagados a toda la WSN. Cuando el dispositivo de mano de un miembro del personal médico establece contacto con un nodo sensor de las víctimas, este último envía toda su información, el resumen del estado de toda la red de sensores, al dispositivo de mano del personal médico. El resumen es finalmente enviado al ECC con un agente móvil de la misma manera que en MAETT [11]. De esta forma el ECC conoce los cambios de salud de las víctimas y puede planificar mejor su tratamiento.

III-A. Despliegue de permisos

En este escenario, proponemos el uso de los permisos descritos en la sección II-B. En nuestro caso contamos con una autoridad administrativa que es la máxima autoridad de coordinación en una situación de emergencia, el ECC, generalmente de carácter gubernamental.

Esta autoridad genera un retículo de permisos inicial con la previsión de dar cabida a gran numero de usuarios y situaciones

Como muestra la Figura 4, este esquema está organizado por unidades administrativas. El esquema de permisos queda dividido en los centros de coordinación de cada organización que participa (centro de bomberos, médico, etc.). Así mismo estos se siguen subdividiendo hasta donde sea necesario. Finalmente se encuentran los permisos asociados a los propios sensores (en este caso por tipo de sensor, y acción: *read*, *write*).

El retículo de la figura 4 se codifica como un retículo de permisos de Bloom tal como se explica en la sección II-A, y se hace uso de la delegación descrita en la sección II-B1 para evitar que la autoridad central tenga que asignar los permisos directamente a cada participante.

Este es solo un ejemplo ilustrativo y seguramente un poco irreal. La intención aquí es simplemente poder discutir el sistema en general y este esquema sirve a propósitos ilustrativos. La parte baja del retículo tiene los permisos finales que irán incluidos en los sensores. Estos sensores pueden realizar la operación de verificación de permisos descrita en la sección II-B2 para comprobar la peticiones de lectura o escritura que reciban. Dado que el esquema de permisos es un retículo, aunque pueden existir un gran numero de permisos (muchos cuerpos de emergencia diferentes con muchos departamentos, unidades, etc.), los permisos finales (parte baja) puede limitarse mucho. Esto hace que su implantación en los sensores sea bastante sencilla. Así mismo es importante tener en cuenta que estamos asumiendo que la aplicación es cerrada en el sentido en que es la misma entidad quien implanta y distribuye los sensores (directa o indirectamente).

El uso de este esquema de permisos proporciona dos importantes ventajas:

1. La administración de permisos queda distribuida mediante el uso de *delegación*, de manera similar a esquemas de *trust management* como SPKI/SDSI [5].
2. Las operaciones de verificación y delegación son muy eficientes. El cálculo necesario para verificar un permiso y realizar una delegación se limita a aplicar un número bajo y acotado de funciones hash, y hacer alguna operación como un OR bit a bit en el caso de la delegación de muy bajo coste computacional.

De la misma manera es importante tener en cuenta que estas ventajas comportan otras desventajas. Por una parte estamos sacrificado cierta seguridad en el sistema en pro

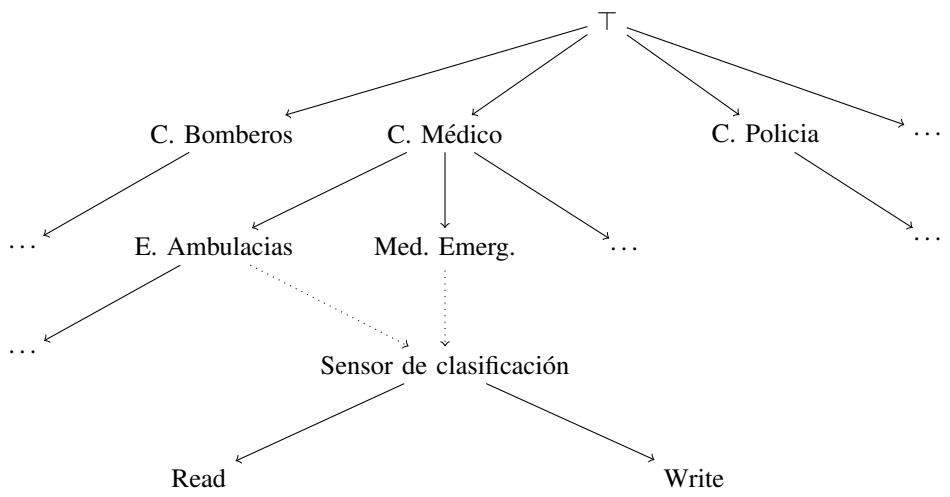

Figura 4. Ejemplo del esquema de permisos.

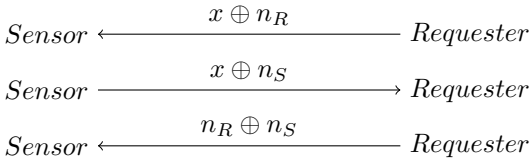

Figura 5. Ejemplo de protocolo para la petición de acceso.

de la eficiencia. Nuestro esquema no proporciona el mismo nivel de seguridad que proporcionaría un sistema basado en criptografía asimétrica, con el uso, por ejemplo, de certificados de autorización (o atributos). Por otra parte, y quizás más importante, estamos asumiendo que las comunicaciones del sistema son seguras. Es decir, un atacante no debe poder capturar el envío de una petición de acceso con un permiso de Bloom en ella, ya que podría realizar un ataque de repetición.

Respecto a este último punto existen diversas estrategias, entre ellas:

1. Si la aplicación es cerrada se asume que dichas comunicaciones pueden ser seguras (secretas y auténticas). Por ejemplo con el uso de criptografía simétrica con claves compartidas. Hay que tener en cuenta también que este tipo de aplicaciones tienen un tiempo de vida relativamente corto, por lo que se reduce la posibilidad de compromiso de claves. También asumimos aquí que los dispositivos son en cierta medida *tamper-proof*.
2. Evitar que en las peticiones de acceso realizadas sobre los sensores sea necesario incluir el permiso de Bloom. Si consideramos dicho permiso de Bloom, conocido tanto por el sensor como por quien realiza la petición de acceso ($requester$), como x, podemos utilizar un protocolo entre ambos como el que se muestra en la figura 5, donde n_R es un valor aleatorio generado por el *requester* y n_S un valor aleatorio generado por el sensor (los dos valores son secretos).

IV. Conclusiones

En este artículo hemos presentado la aplicación de unas estructuras de permisos basadas en funciones hash como son los filtros de Bloom, para dotar de un esquema de autorización en entornos donde el uso de criptografía asimétrica puede ser excesivamente costoso.

En nuestro caso estamos trabajando en la aplicación de dicho esquema sobre una aplicación para la coordinación en entornos de emergencia de diversos equipos. Dicha aplicación se esta desarrollando sobre TinyOS y Agilla en dispositivos TelosB.

Agradecimientos

Se reconoce el apoyo del gobierno Español (proyectos TSI2007-65406-C03-02, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, TIN2010-15764 y TIN2011-27076-C03-03) y de la Comisión Europea, proyecto (FP7/2007-2013) Data without Boundaries (grant agreement number 262608) .

Referencias

[1] Francesco Aiello, Giancarlo Fortino, Raffaele Gravina, and Antonio Guerrieri. A java-based agent platform for programming wireless sensor networks. *The Computer Journal*, 54(3):439–454, 2011.

[2] J. Ametller, S. Robles, and J. A. Ortega-Ruiz. An implementation of self-protected mobile agents. In *Eleventh IEEE International Conference and Workshop on the Engineering of Computer-Based Systems.*, pages 544–549, Brno, Czech Republic, May 2004. IEEE Computer Society Press.

[3] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7), 1970.

[4] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2 edition edition, 2002.

[5] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. Spki certificate theory. RFC 2693 Experimental, 1999.

[6] Ch-L. Fok, G-C. Roman, and Ch. Lu. Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. *ACM Trans. Auton. Adapt. Syst.*, 4(3):1–26, 2009.

[7] S. Foley. Using trust management to support transferable hash-based micropayments. In *Financial Cryptography*, pages 1–14, 2003.

[8] Dimitrios Georgoulas and Keith Blow. In-motes: an intelligent agent based middleware for wireless sensor networks. In *Proceedings of the 5th WSEAS international conference on Applications of electrical engineering*, AEE'06, pages 225–231, Stevens Point, Wisconsin, USA, 2006. World Scientific and Engineering Academy and Society (WSEAS).

[9] YoungMin Kwon, Sameer Sundresh, Kirill Mechitov, and Gul Agha. Actornet: an actor platform for wireless sensor networks. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06, pages 1297–1300, New York, NY, USA, 2006. ACM.

[10] Paolo Maggi and Riccardo Sisto. A configurable mobile agent data protection protocol. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, AAMAS '03, pages 851–858, New York, NY, USA, 2003. ACM.

[11] R. Martí, S. Robles, A. Martín-Campillo, and J. Cucurull. Providing early resource allocation during emergencies: The mobile triage tag. *Journal of Network and Computer Applications*, 32:1167–1182, 2009.

[12] E. Mercadal, S. Robles, R. Martí, C. Sreenan, and J. Borrell. Double multiagent architecture for dynamic triage of victims in emergency scenarios. *Progress in Artificial Intelligence*, 1:Special issue PAAMS 2011. To appear, available at https://senda.uab.cat/wiki/pai11, 2011.

[13] Estanislao Mercadal, Carlos Vidueira, Cormac J. Sreenan, and Joan Borrell. Improving the dynamism of mobile agent applications in wireless sensor networks through separate itineraries. *Computer Communications*, 2012, submitted.

[14] T. P. Pedersen. Electronic payments of small amounts. In *Proc. 4th International Security Protocols Conference*, pages 59–68, 1996.

[15] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IPSN '05 Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 364–369. ACM and IEEE, 2005.

[16] R. L. Rivest and A. Shamir. PayWord and MicroMint: Two simple micropayment schemes. In *Proc. 4th International Security Protocols Conference*, pages 69–87, 1996.

[17] G. Super, S. Groth, and R. Hook. START: simple triage and rapid treatment plan. *Hoag Memorial Hospital Presbyterian, Newport Beach (CA)*, 1994.

[18] L. Szumel, J. LeBrun, and J. D. Owens. Towards a mobile agent framework for sensor networks. In *Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*, pages 79–87, Washington, DC, USA, 2005. IEEE Computer Society.