Seguridad en redes de sensores inalámbricos

Catalina Aranzazu Suescún*

Fecha de recepción: 6-02-2009 Fecha de selección: 20-04-2009 Fecha de aceptación: 16-04-2009

ABSTRACT

This article presents a survey of the state of art of the mechanisms to provide security to the Wireless Sensor Networks. Over the principal schemes, are the key management and cryptography, the authentication protocols and the mechanisms of clone or intrusion detection.

INDEX TERMS

Attacker, MAC, key management, limited resources, security, sensor node.

RESUMEN

Este artículo presenta un estudio del estado del arte de los mecanismos para proveer seguridad en las redes de sensores inalámbricas. Dentro de los principales esquemas de seguridad se encuentran los manejos de claves y criptografía, los protocolos de autenticación y los mecanismos de detección de intrusos o clones.

PALABRAS CLAVE

Atacante, manejo de claves, MAC, nodos de sensores, recursos limitados, seguridad.

Clasificación Colciencias: Tipo 3

* Estudia Maestría en Telecomunicaciones en la Universidad Pontificia Bolivariana (UPB), Medellín (correo e.: catazazu@gmail.com)

I. INTRODUCCIÓN

Una red de sensores inalámbricos (WSN, *Wireless Sensor Network*), puede estar compuesta por una serie de miles, incluso millones de sensores, llamados nodos, los cuales poseen capacidad de almacenamiento, procesamiento y energía limitada.

Estas redes son utilizadas en aplicaciones militares, médicas, biológicas, entre otras. Generalmente se despliegan en ambientes hostiles para la recolección de diferentes tipos de datos, por lo cual se ven expuestas a severos ataques físicos y de software. El desarrollo de métodos que aumenten la seguridad se convierte entonces en un punto esencial en el estudio de las redes de sensores.

En este artículo se presenta un estudio de los principales problemas y las últimas soluciones planteadas en el tema de seguridad en redes de sensores inalámbricos. Se organiza de la siguiente manera: la Sección II, muestra los principales ataques del cual son objeto las WSN; la Sección III, se enfoca en las soluciones que utilizan el manejo de diferentes tipos de claves y cifrado; la Sección IV, describe protocolos de autenticación de sensores; la Sección V, presenta algoritmos y esquemas de detección de intrusos en la red y finalmente, se entregan algunas conclusiones sobre el tema.

II. TIPOS DE ATAQUES

Las redes de sensores están predispuestas a múltiples ataques debido a que su despliegue se realiza en áreas abiertas. Dentro de los principales ataques [1,2] se encuentran:

- Negación del servicio (DoS): Este tipo de ataque ocurre a nivel físico, un nodo malicioso envía indiscriminadamente mensajes que consumen el ancho de banda disponible de la red, consiguiendo la indisponibilidad temporal de un servicio o inclusive degenerando todo el sistema.
- Nodos comprometidos y suplantación de fuentes: En este caso el atacante introduce, ya sea física o por software, un nodo corrupto a la red para transmitir información corrupta.
- Recolección pasiva de información o *Eavesdropping*: El atacante "escucha" y recolecta la información de la red, sin realizar ningún daño al sistema.
- Ataques físicos: Es la sustracción física de los nodos de la red para sustraer la información y claves de criptografía.
- *SinkHole*: Se introduce un nodo malicioso cerca a la estación base para atraer información confidencial.
- Ataque *Sybil*: El atacante introduce múltiples nodos con identidades ilegítimas o con identidades hurtadas de la red.
- Ataque gusano (*Wormhole*): En este caso se genera un túnel entre dos nodos, por el cual el atacante recolecta la información y la reenvía con cierto retraso.

La Figura 1 presenta las principales amenazas en las redes de sensores.

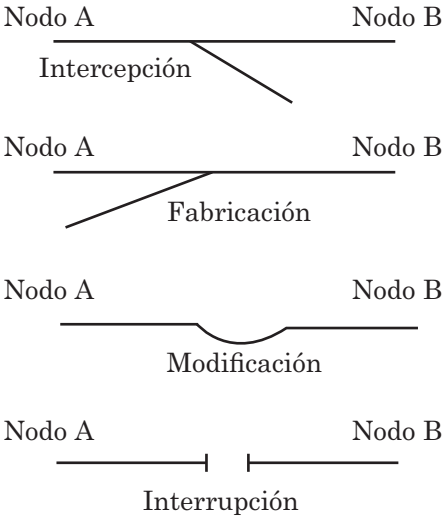

Figura 1. Amenazas de las redes WSN

III. MANEJO DE CLAVES Y CIFRADO

Gran parte de las implementaciones de seguridad en los WSN utilizan encripción y esquemas de manejo de claves. El principal problema a solucionar es lograr que el esquema de manejo sea suficientemente eficiente, de tal forma que si un intruso logra capturar un nodo, no le sea posible acceder a todas las claves de la red y por tanto a la información confidencial del sistema.

Los diferentes autores que investigan soluciones para la seguridad de las WSN, realizan suposiciones referentes a la capacidad de los diferentes nodos. Como se expuso anteriormente, los sensores poseen capacidades de procesamiento, almacenamiento y una fuente de energía limitada, sin embargo, es posible encontrar un pequeño grupo de nodos, cuyos recursos no sean tan limitados, denominados superiores o cabeza de *cluster* (celda). Para estos casos, la red es heterogénea. [3,4] En el caso en que todos los nodos posean las mismas características se les denomina "red homogénea". En esta sección se presentan los principales desarrollos en esquemas de manejo de claves con sistemas homogéneos y heterogéneos.

A. Redes homogéneas

• Piscina de claves

Un concepto ampliamente utilizado en los mecanismos de manejo de claves es el de piscina de claves. El primer trabajo que usa el concepto de piscina es el de [5].

A diferencia de [5] donde solo se utiliza una piscina de claves, en [6] se manejan dos piscinas, una de transmisión y otra de retorno. Antes del despliegue, a cada nodo se le entregan, aleatoriamente, dos anillos de $m/2$ claves de cada piscina (donde m es el número total de claves por nodo). Las piscinas se actualizan mediante una función *Hash* [7], luego de un período de tiempo o generación. Cada piscina posee $P/2$ claves, donde P es el número total de claves del sistema. Las claves de retorno se generan usando la cadena de *Lamport* basada en *Hash* [8], comenzando por las claves que corresponden a la última generación. Luego del despliegue, cada nodo inicia el reconocimiento de sus vecinos enviando un mensaje con su ID y el número de generación en el cual fue desplegado. Si un vecino encuentra claves comunes, envía un mensaje de respuesta con su ID y su generación. Se calculan entonces las generaciones en el ciclo de vida de cada nodo que se superponen, como se observa en la Figura 2, pues solo en estas generaciones se puede establecer comunicación.

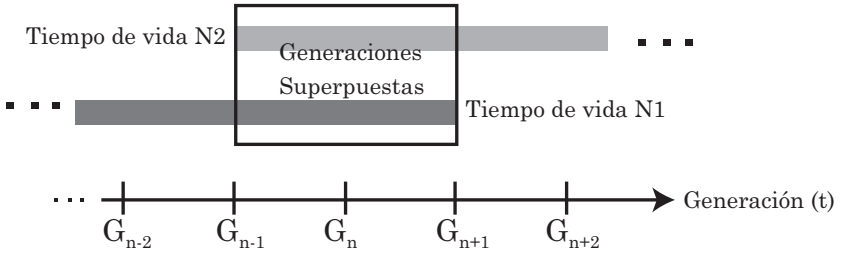

Figura 2. Ventana de generación

Continuando con el manejo de claves mediante piscinas, en [9] se presentan dos esquemas donde emplean diversas piscinas. El primer esquema se denomina ABAB, debido a la utilización de dos piscinas de claves, A y B, y un tercer conjunto (S) que está compuesto por las claves que comparten A y B. Se escogen *m* claves de A y se almacenan en un nodo y *m* claves de B y se almacenan en otro nodo. Se continúa el mismo proceso hasta completar todos los sensores de la red. Finalmente, se despliegan los nodos como se presenta en la Figura 3.

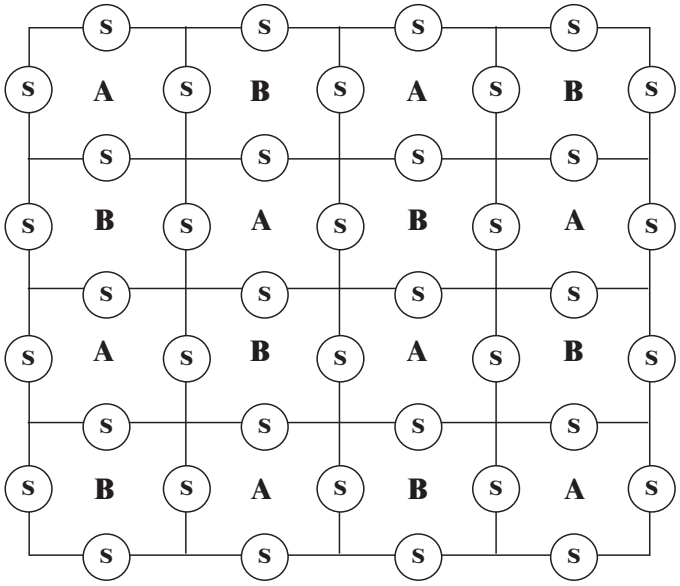

Figura 3. Esquema ABAB

El esquema ABCD, por su parte, usa *2r* piscinas, donde *r* es el número de filas de nodos a desplegar. En este caso, el conjunto S, comprende las claves que comparten las *2r* piscinas. El despliegue se muestra en la Figura 4. Una

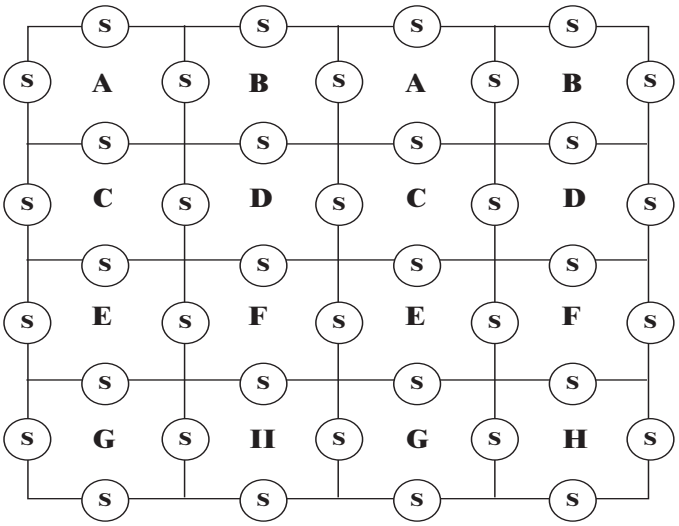

Figura 4. Esquema ABCD

vez efectuado el despliegue, se usan los protocolos de [5, 10] para establecer claves pares entre nodos.

En [11], se retoma la idea básica de [5], con una sola piscina de la cual se seleccionan P claves aleatoriamente. Se almacena un número m de claves en cada nodo antes de ser ubicado en el campo de sensado. Una vez se realiza el despliegue, los nodos buscan con cada vecino, la clave común para establecer una comunicación. Para esto, mandan múltiples mensajes cifrados, con cada una de las claves almacenadas, así, el vecino que logre comprender la clave descubre entonces lo que será la clave de sesión con el nodo transmisor. Pero si un par de nodos no encuentran una clave compartida, la generan. Para mejorar la seguridad, cada nodo borra aleatoriamente una serie de claves de su memoria.

- Nodos conscientes de su localización

Existen aplicaciones en las que, debido a la hostilidad del campo de sensado, es imposible acceder vía terrestre al lugar de interés y es necesario desplegar los nodos aleatoriamente desde el aire, lo que hace muy difícil determinar su ubicación exacta en el campo. Sin embargo, existen ocasiones donde los sensores están conscientes de su localización en la red.

En [12], los autores sugieren un nuevo esquema de generación de claves aleatorias cuando los nodos son conscientes de su localización. El campo de sensado se divide en pequeñas áreas y los N sensores desplegados son distribuidos uniformemente en el área formando celdas, como se observa en la Figura 5. Un servidor distribuye las claves en los nodos, que han sido generadas con una clave maestra y una función pseudoaleatoria. La distribución de las claves puede realizarse para sesiones en un grupo o entre grupos vecinos. En el primer caso, solo un nodo almacena la clave de sesión con un vecino, pues el otro nodo puede generar la clave

con su propia clave maestra. De esta forma se reduce a la mitad el número de claves almacenadas en cada nodo. Y en el segundo caso, con un grupo que tiene cuatro grupos vecinos, se determina para cada nodo de un grupo, varios nodos de cada grupo vecino con los cuales entablar sesiones y se le entregan las claves de la mitad de los vecinos ya que, como se mencionó anteriormente, puede generar las otras claves. Con este esquema se soportan redes grandes al no tener que almacenar una considerable cantidad de claves.

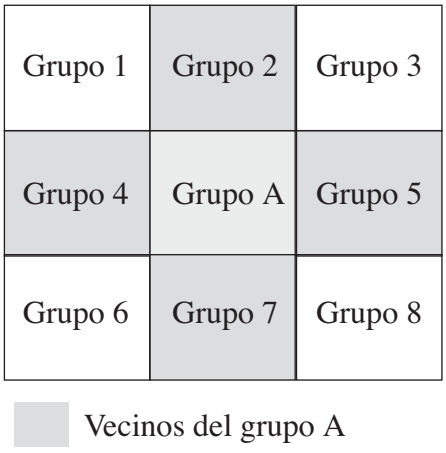

Figura 5. División del área de sensado

En [13] los nodos también conocen su localización en la red. La diferencia con el anterior esquema es la introducción de un *hardware* reprogramable, para la generación o actualización de claves, que utiliza DES (*Data Encryption Standard*) [14]. La BS examina las posibles rutas hacia cada nodo en el *cluster*, enviando de manera secuencial mensajes *Hello*. Es necesario entonces tener un previo conocimiento del número de nodos y su localización. Los saltos en las rutas de cada nodo no deben sobrepasar un umbral, para obtener un eficiente consumo de energía. Luego de almacenar las rutas, se asigna un sello de tiempo o periodo a cada nodo para que transmita sus mensajes, de manera similar a una multiplexación en el tiempo. Como entradas del hardware se encuentran la información del mensaje y la clave secreta compartida entre el nodo que trasmite y el próximo salto. Cuando es necesario realizar una actualización de claves, en el *hardware*, se ejecuta una XOR entre la clave antigua, el valor de la marca de tiempo, y el número de iteraciones que en el proceso se ha repetido. En el esquema, no existe una transmisión de claves, sino que cada nodo es previamente sincronizado para actualizar las claves.

• Otros esquemas

Algunos autores han decidido no basarse en trabajos anteriores y disminuir las consideraciones en la red de casos especiales o puntuales, como el conocimiento previo de la localización de cada nodo.

Para el cifrado, en [15] se supone que los nodos comparten una clave simétrica con la estación base y conocen la clave pública de la BS, además poseen claves pares simétricas con los vecinos. El proceso comienza con la generación de un número *n* (*nonce*) para la comunicación del nodo fuente con la estación base. El nodo calcula la MAC (código de autenticación de mensaje, *Messaje Authentication Code*) con la clave simétrica para sesiones con la BS, el *nonce* y el ID del nodo. Cifra la MAC, el *nonce* y el mensaje, con la clave que comparte con la BS, y obtiene el texto cifrado que va a transmitir. Cuando un nodo

de tránsito lo recibe, genera un número aleatorio para decidir si reenvía el mensaje sin modificación o si lo encapsula en un nuevo cifrado. La BS, descifra el mensaje, y compara la MAC que encuentra con la que anteriormente había calculado, y si es correcta envía el mensaje a su destino final. Si es necesario realizar un cambio de claves, el nodo fuente genera la clave nueva. Calcula un nuevo *nonce* y la MAC, con la clave nueva y una bandera que advierte a la BS de la presencia de una renovación. En la BS, se verifica que el *nonce* sea nuevo, y que la MAC corresponda a la del nodo fuente y si es correcta, se reemplaza la clave de sesión anterior con la recibida.

Otra consideración simple se encuentra en el esquema de [16], donde antes del despliegue se selecciona una clave maestra y se decide el tamaño máximo de cada celda. A cada sensor se le asigna una clave, cifrada en la clave maestra. La BS, entrega a los nodos una clave de grupo y calcula una función polinomial f(x) de un campo finito o de *Galois* GF [17], que se obtienen a través de *m+1* puntos (x,y) aleatorios (los puntos corresponden a localizaciones de nodos). Las funciones f(1), f(2)…f(m), junto al valor del tiempo actual *t*, cifrado en la clave de grupo y sin cifrar, son enviadas como *broadcast* a los nodos. Cada sensor, según el esquema *Shamir* [18], determina la clave con las funciones recibidas. Luego verifica si la clave encontrada es auténtica, descifrando el tiempo actual *t* y comparándolo con el recibido. Si es necesario enviar o renovar las claves, se cifran con la clave anterior. Si se encuentra un nodo corrupto, se vuelve a calcular la función polinomial y se excluyen las localizaciones de los nodos afectados.

B. Redes heterogéneas

• Enlaces solo con sus superiores

En las redes heterogéneas existe una jerarquización de los *clusters*, pues se tienen nodos denominados superiores. Con esta característica, algunos autores han decidido que la comunicación se establezca siempre entre nodos y sus superiores y de ser necesaria una sesión entre nodos que utilice como intermedio el jefe de *cluster*. Con esto se logra reducir el número de claves almacenadas o generadas por los nodos.

En [19], se enseñan dos protocolos donde solo trabajan con claves de grupo, claves públicas de superiores y claves de sesión con jefes de celda. Organizan la red en celdas mediante el algoritmo HEED (*Hybrid, Energy-Efficient, Distributed Clustering*) [20], de esta manera existe una jerarquización de los nodos. El primer protocolo busca la distribución de la clave para cada celda. Para esto, cada superior o cabeza de *cluster* genera una clave de grupo que envía a cada nodo de su celda. Esta clave de grupo se cifra con la clave que permite el enlace entre el superior y cada nodo, o mediante la clave pública del superior, las cuales son creadas en el despliegue. Cuando es necesario generar una nueva clave, el superior envía una petición de renovación de clave a la estación base (BS, *Base Station*) más cercana, luego de autenticarse mediante una firma digital, la cual contiene la clave privada y la identificación del superior. En el segundo protocolo, se genera la clave de sesión con la estación base, la cual se cifra mediante

la clave de grupo o *cluster* y se envía a cada miembro de la celda. Para brindar una mayor seguridad en la red, es necesario realizar el mismo procedimiento luego de que un nodo utilice las claves $22^{K/3}$ (donde K es el número de bits de la clave) y se adhiera o remueva un nodo de la red.

Por otro lado, los autores de [21], se enfocan en el tipo de encripción a utilizar. Está basada en LCG (generador lineal congruente, *Linear Congruential Generator*) [22, 23] y rotación ortogonal de matrices [24]. Para la primera encripción se genera una secuencia de números aleatorios *Sn*, de acuerdo con una semilla *So*. Estos números de secuencia se escogen de tal manera que el flujo de números provea un ciclo o periodo largo y tenga buenas propiedades estadísticas. Solo la semilla es secreta, los otros números y funciones con los que se genera la secuencia son públicos. Para la encripción final se usan matrices ortogonales, de modo que se escoja un ángulo de rotación conocido entre el superior y el nodo con el cual se establece la comunicación, para lograr descifrar correctamente.

• Enlaces entre nodos

Cuando es posible trabajar con sensores que posean una considerable capacidad de almacenamiento o procesamiento, se pueden establecer sesiones entre nodos, lo que permite funciones de agregación de datos [25], protección contra intrusos, entre otros.

Para los autores en [26], los nodos poseen un par de claves pública/privada, una clave de grupo que se actualiza constantemente y un certificado de autenticidad de la clave pública, pero solo los superiores o *Gateways* pueden usar la clave pública para calcular firmas digitales. Los nodos comparten la clave pública con el superior. Una autoridad de certificación firma las claves públicas de los *gateways*. Mediante la clave pública se genera una clave simétrica de sesión entre el nodo y su superior. Los nodos firman todos los datos transmitidos con la clave de sesión y los envían al superior. El *gateway* firma el mensaje con su clave privada antes de enviársela al servidor. Los sensores pueden utilizar su clave de sesión para generar claves pares para la comunicación entre nodos. La clave de grupo se utiliza para generar la MAC. Cuando se despliega un nodo nuevo, éste envía una petición de certificación al superior, cifrada en la clave de grupo. El *gateway* recibe la petición y le remite al nodo el certificado que contiene su clave pública. El sensor verifica la validez del mensaje con el certificado de autenticidad de clave pública precargado. Si es correcto, almacena la clave pública del superior. Luego, como respuesta, escoge un número aleatorio y junto a su ID, lo cifra con la clave pública del superior y lo envía. El *gateway* recibe el mensaje, lo descifra con su clave privada para almacenar el ID y el número aleatorio del nodo. Responde al sensor, transmitiendo el número del nodo, un número aleatorio, la clave de sesión y la clave de grupo, cifrada en la clave pública del nodo. El nodo colecta las claves y para verificar que posee las claves correctas, devuelve un mensaje al superior con el número aleatorio del *gateway* cifrado en la clave de sesión.

Una función pseudo aleatoria (PRF, *Pseudo-Random Function*) es usa-

da en [27] para crear una piscina de claves con las cuales establecer comunicación entre nodos vecinos. Suponen que los nodos son estáticos y conocen su localización en la red. Los superiores son denominados H y los nodos comunes son llamados L. El esquema utiliza tres pasos. Primero, se genera la piscina gracias a una clave maestra, y se precarga con *n* claves cada nodo L. Segundo, cuando ya están desplegados todos los sensores, los nodos L envían a su superior, las listas de claves de sesión, con los ID y localización de los nodos vecinos, para que las memoricen. Los superiores H, reenvían las listas completas a los nodos L para que las verifiquen. Tercero, se generan las claves entre superiores. Si la red es demasiado grande se utiliza un esquema distribuido donde cada nodo L trata de encontrar sus pares de claves para comunicarse con sus vecinos y solo pide ayuda al superior en caso tal de no encontrarlas.

Por otro lado en [28], se forman los *clusters* y los superiores envían un mensaje *Hello* que concatena su ID con su clave privada. Los nodos L recolectan los *Hello* y envían su propia concatenación para certificarse. Luego los nodos L generan una pseudo celda con los nodos vecinos según el diagrama de Voronoi [29], con el fin de crear sus propias claves de sesión.

En [30], se divide el campo de sensado en áreas iguales (basado en el trabajo de [31]) como se muestra en la Figura 5, los sensores conforman grupos uniformes y cada grupo selecciona un superior. Antes del despliegue se genera una clave para cada grupo de nodos y se almacena en cada integrante. Además se crean claves intergrupales para la comunicación entre los cinco grupos vecinos, las cuales son almacenadas en un número aleatorio de nodos. Cada nodo, llamado A, envía un *broadcast* con su ID, la identificación del grupo y número aleatorio *nonce*. Si un nodo, denominado B, que se encuentra en el mismo grupo recibe el mensaje, concatena su ID con una MAC que posee su ID, el *nonce* del nodo A y la clave de grupo y la envía al nodo A. El nodo A recibe y verifica la autenticidad del nodo B con la clave de grupo. Si el nodo B es legítimo, el nodo A genera la clave de sesión y la envía a B. Si un nodo necesita establecer una sesión con un nodo de otro grupo, envía una petición a su superior con las claves para enlazarse. El superior A cifra el mensaje con la clave de intergrupo y se lo transmite al superior B del grupo destino. El superior B lo descifra y envía las claves al nodo final. Luego de realizar toda la configuración de claves, se eliminan las claves de intergrupo.

Al igual que en la investigación de [13], el trabajo en [32] introduce un componente de hardware llamado LFSR (*Linear Feedback Shift Register*), el cual es el responsable de la generación de claves. Para la arquitectura, todos los nodos deben poseer LFSR y formar celdas con superiores o cabeza de *cluster*. La BS genera *n* secuencias de pulsos que sirven como entrada serial al LFSR para crear números aleatorios que funcionan como las claves. Estos números aleatorios o claves, se entregan a los nodos y se renuevan en cada sesión. El superior utiliza las mismas secuencias de pulsos de los nodos, pero invertida, de manera que ningún intruso com-

prenda el funcionamiento. Cuando un nodo ha utilizado toda la secuencia de pulsos disponible, intercambia su secuencia con otros nodos, utilizando como intermediario, su superior.

C. Análisis de los mecanismos de manejo de claves

Los mecanismos de manejo de claves que utilizan el concepto de piscina de claves, deben poseer nodos con grandes capacidades de almacenamiento, lo cual se imposibilita en la medida en que los sensores tengan limitaciones en la memoria. Sin embargo, cuando las claves se renuevan cada cierto periodo y los nodos que fallecen son reemplazados, el número de nodos comprometidos disminuye considerablemente, como ocurre en [6]. Si además, como en [9], se trabajan con más de una piscina, las cuales pueden ser reutilizadas en diferentes zonas, los anillos de claves de cada nodo puede aumentar consiguiéndose mayor seguridad y mejor conectividad.

Los mecanismos que necesitan que los nodos posean conocimiento previo de su localización, solo pueden ser usados en aplicaciones donde la red se despliegue de manera manual y premeditada. No obstante, poseen una ventaja frente a otros mecanismos cuando se usan claves maestras que el atacante no conoce y por tanto no se pueden generar claves de cifrado y descifrado [12].

El algoritmo [16], puede forzar la salida de un nodo comprometido, al aislarlo del sistema cuando genera daños en la red, pero su efectividad depende de los recursos del nodo.

Algunos esquemas, aunque son novedosos, poseen gran complejidad computacional [21] o comprometen el tamaño del chip y aumentan el consumo de potencia del nodo, disminuyendo su tiempo de vida [32].

Los mecanismos que poseen redes heterogéneas disfrutan la ventaja, frente a los sistemas homogéneos, de la posibilidad de descargar la seguridad a los nodos que tienen mayores recursos de procesamiento y almacenamiento, logrando aliviar el funcionamiento de los nodos comunes y en general de la red.

Dentro de los temas abiertos para trabajos futuros, se encuentran los siguientes aspectos:

- Mejorar la capacidad de adicionar y retirar nodos de la red sin comprometer la seguridad del sistema.
- Reducir el consumo de energía, disminuir el encabezado u *overead* de los mensajes que generan los diferentes algoritmos.
- Comprobar la efectividad de los mecanismos en diversas redes y topologías.
- Mejorar las políticas para renovar las claves, que permitan reducir el tiempo de operación.
- Concretar la arquitectura de hardware del sensor embebido [32].

IV. AUTENTICACIÓN

La autenticación es un punto fundamental en la seguridad de las redes de sensores. Esto se debe a que un atacante puede clonar un nodo o sustraer la información de las claves de la red y enviar información maliciosa a la red. Es necesario entonces generar mecanismos que permitan a los nodos reconocer que la información

recibida es auténtica, mediante la validación de la identidad del nodo transmisor. El mensaje de autenticación más utilizado es el MAC, el cual contiene diferente información que justifica la legitimidad de un nodo.

A. Autenticación de múltiples saltos

La autenticación por múltiples saltos se utiliza, generalmente en la verificación de las ID de los nodos nuevos que se despliegan en la red, para generar sesiones entre nodos o simplemente en las ocasiones en que no es necesario que toda la red, sino solo un grupo o un nodo, conozca la validez de un sensor.

Para el desarrollo de [33], se utiliza como base el protocolo de autenticación con *broadcast* μTESLA [34]. Suponen que solo la estación base puede realizar *broadcast*. Antes del despliegue, el nodo que transmite (A), crea dos secuencias de claves aleatorias y les aplica una función *Hash* repetidamente para generar otras claves. Así, se obtienen dos cadenas de claves, denominadas primaria y secundaria. La primera clave de la cadena primaria es llamada compromiso de la cadena y se utiliza para autenticar las otras claves. Y la primera clave de la cadena secundaria, es para actualizar la anterior cuando haya expirado. Cada nodo que transmite determina el tamaño del intervalo para la transmisión de paquetes. Para comenzar la autenticación, el nodo A envía las dos primeras claves de cada cadena, el intervalo y el tiempo de inicio del envío, además, asocia una clave de la cadena primaria a cada intervalo. En cada intervalo, el nodo A utiliza la clave asociada para generar la MAC y la envía. Si la cadena primaria expira, se actualiza con la secundaria y se genera otra nueva. El nodo que recibe (B), se asegura que el tiempo de tránsito del paquete no sobrepase el tamaño del intervalo más un retraso por divulgación, pues puede ser un paquete modificado. Si está correcto el tiempo, verifica la clave asociada al intervalo y si es correcta, es decir, no ha expirado, la almacena.

Por su parte, en [35], se despliegan aleatoriamente, sensores y programas de verificación de integridad, PIV (*Program-Interity Verification*). Algunos PIV se utilizan como servidores de autenticación. Luego del despliegue, se comienza el reconocimiento de vecinos, mandando *broadcast* con su ID, y los vecinos que quieren establecer una comunicación deben crear una clave de sesión. Para la autenticación, cada nodo escoge el PIV más cercano y calcula la clave de sesión entre ellos. El sensor envía un mensaje al PIV con un *nonce* y la MAC calculada con la clave de sesión. El PIV recibe el mensaje, calcula la clave de sesión y verifica el mensaje. Si está correcto, lo transmite a los PIV's de referencia que tiene almacenados en memoria, adicionándoles el *nonce* y ID del nodo, y la MAC calculada con la clave de sesión entre PIV's. Cada PIV de referencia que recibe el mensaje, verifica que sea legítimo y transmite un tiquete de autenticidad al PIV original, el cual comprueba que el tiquete sea de un PIV válido. De ser así, retransmite el tiquete al nodo, terminando el proceso.

En [36], los autores desarrollan un esquema de autenticación basados en el desarrollo SEF (*Statistical En-route Filtering Scheme*) de [37]. En el sistema se asume que la BS tiene

una piscina de claves global, que está dividida en sectores y cada nodo, antes de ser desplegado, almacena un cierto número de claves de una de las particiones de la piscina, seleccionada aleatoriamente, y además, los sensores conocen su posición en la red. Cuando se realiza el despliegue, se forman celdas en la red. Si ocurre un evento extraño, los nodos que lo detectan calculan su MAC con la información de la sección de la piscina a la cual pertenecen sus claves, y la envían al superior. El jefe de *cluster* escoge aleatoriamente una de las MAC que recibe para hacer un reporte de evento, cuando recoge una cantidad de MAC que supere un umbral establecido. Luego de determinar un enrutamiento por toda la red, manda un mensaje con la información sobre el ID, la localización y las secciones de piscina que posee el *cluster* al cual pertenece el nodo de la MAC que escogió anteriormente. La BS recibe la información y genera un mensaje de MAC asociado (MMA), el cual lo envía al superior del próximo *cluster*. Este superior almacena la información del *cluster* vecino y envía el mensaje a la próxima celda, continuando los saltos hasta autenticarse en toda la red

Para la autenticación, algunos autores en sus investigaciones utilizan árboles donde se jerarquicen las identificaciones de los integrantes de cada *cluster*. En [38], se asume que los nodos están pre-cargados con una clave y poseen una ID única. El servidor, antes del despliegue, predistribuye a cada nodo información secreta (para generar claves secretas), calcula una información de compromiso a partir de la información secreta, construye el árbol *Merkle* [39], donde el nivel más bajo contiene la función *Hash* de la ID de cada nodo concatenada con su información de compromiso, y entrega a cada superior su certificado de autenticidad y la etiqueta del administrador del árbol. Los datos de compromiso no son invertibles, es decir no es posible abstraer la información secreta de ellos. Luego del despliegue, cada jefe se valida con otros superiores presentando sus certificados y demostrando de manera única (no es posible entregar dos veces la misma demostración), que posee la información secreta. Luego, los superiores organizan la red, recolectando las ID e información secreta de los nodos. Calcula la información de compromiso con la cual construye el árbol *Merkle* local y envía el certificado de autenticidad y etiqueta del administrador a cada nodo. Luego, cada nodo puede verificar su autenticidad presentando el certificado y demostrando que posee la información secreta.

También, es posible utilizar otro tipo de árbol, cuya finalidad siempre es jerarquizar la red. En [40], inicialmente se asume que cada nodo hace parte de un grupo pequeño. Luego cada grupo recolecta información acerca del tamaño de los grupos vecinos. Con la lista, propone una fusión con un vecino de tamaño reducido. Si el grupo vecino acepta, se comienza una autenticación de los grupos. Se construye el árbol *Hash* de cada grupo, donde el nivel más bajo está conformado por las ID de los nodos, los niveles superiores son los ID de los administradores de grupo y el último o cúspide del árbol, es el identificador de la BS. El grupo de menor tamaño anuncia su identificación y tamaño al grupo que envía la petición de fusión. El grupo de mayor tamaño,

selecciona un nodo como retador. El nodo escoge un reto (mensaje cifrado) y lo envía a sus compañeros de grupo que verifican que el reto sea correcto. Luego se transmite al grupo vecino. El grupo de tamaño reducido, de acuerdo con el reto, selecciona un nodo como retado. Este nodo envía como respuesta al reto, su ID y la tabla de fusión del grupo, es decir, el árbol *Hash* del grupo. El grupo de mayor tamaño verifica los valores del árbol *Hash* y si concuerdan con el tamaño e identificaciones conocidas, se fusionan.

B. Autenticación por broadcast

La autenticación por *broadcast* es el esquema más utilizado en las redes de sensores inalámbricos, cuya herramienta básica es la función *Hash*.

Prueba clara de lo anterior se presenta en [41], donde utilizan funciones *Hash* y se asume que la autenticación *broadcast* basada en firmas o en μTESLA se puede utilizar en la red, es decir los nodos poseen suficiente capacidad de procesamiento. El nodo que envía la autenticación, genera una cadena de claves y envía la primera Co, a todos sus posibles receptores. La autenticación se basa en un rompecabezas [42], que posee la concatenación de un mensaje, el verificador de *broadcast* (secuencia que comprueba que no se ha utilizado el mensaje previamente) y una clave de la cadena previamente generada. Al rompecabezas se le adiciona la solución y el número de clave utilizada, se le aplica una función *Hash* y se envía en *broadcast*. El número l de bits en cero al comienzo del rompecabezas con la función *Hash* se conoce como la longitud del rompecabezas y determina los 2^l intentos que realiza el nodo transmisor para conseguir la solución del rompecabezas. El nodo que recibe el mensaje verifica la clave de la cadena mediante la clave Co y si es correcta comprueba la solución del rompecabezas.

Para mejorar la seguridad de la red de sensores, en [17], los autores desarrollan además del esquema de manejo de claves (sección III, Subsección A, Otros esquemas), la etapa de autenticación en la red. El sistema es homogéneo. La BS escoge dos números primos distintos, que posean entre 256–512 bits, y calcula la clave pública *N*, mediante la multiplicación de los números. La clave se almacena en cada nodo. La BS configura una firma *Rabin (RS, Rabin signature)* [43], cuyo resultado es un módulo de cuatro raíces cuadradas de baja magnitud. Luego, envía un *broadcast* con la firma y el valor del tiempo actual *t*. El sensor que recibe el mensaje, verifica que $RS^2 = h(M,t) \bmod(N)$, donde *h(.),* es la función *Hash* y *M* el mensaje. Si el sensor desea enviar un mensaje en broadcast, primero lo envía a la BS con el tiempo actual *t*, y la función *Hash* del mensaje, *t* y la clave secreta del sensor. La BS recibe el mensaje y verifica su autenticidad validando la clave secreta del nodo con la clave maestra. Si es correcto, la BS adiciona al mensaje la firma *Rabin,* el ID del nodo y envía el *broadcast* a la red.

Sin embargo, es posible encontrar desarrollos de esquemas de autenticación que utilizan diferentes funciones o herramientas. Por ejemplo en [44], emplean el protocolo de autenticación RC5 (*Rivest Cipher 5*) [45] para crear la MAC y árboles de claves para la organización de las claves inicial-

mente almacenadas. Los árboles se organizan de tal manera que el nivel más bajo, es decir, las hojas, son las claves de los nodos y el mayor nivel es la clave de la BS (Figura 6). Los nodos pueden utilizar las claves de cada nivel, que confluyan a la misma rama, por las cuales no cruzarían al realizar un trayecto desde la hoja a la BS. Por ejemplo, en la Figura 6, el nodo *N1*, puede utilizar las claves C2, C4, C8 y C9. Cada nodo manda un *broadcast* de mensajes *Echo,* que contiene su ID, un número aleatorio o *nonce* y la MAC, que se genera previamente mediante el uso de RC5 en las claves del árbol. Cuando los vecinos reciben los *Echo*, verifican la autenticidad del ID y luego calculan la MAC y si es correcto, el nodo receptor envía un mensaje *Echo Back* para él autenticarse, el cual posee la MAC de su ID y el *nonce* recibido. La clave de cifración se crea durante la comunicación entre nodos, lo que hace que para un atacante sea difícil capturarla. Para mejorar la seguridad se deben utilizar varios árboles de claves que no estén concatenados, pues un atacante puede conocer las claves que pertenecen a su nivel de descendencia.

C. Análisis de los algoritmos de autenticación

La base de los algoritmos de autenticación es el código de mensaje de autenticación MAC. Como adición o diferenciación, algunos utilizan dos anillos de claves aleatorios que teóricamente tienen un tiempo de vida infinito [33]. Otros manejan esquemas básicos de ID que de forma muy simple, con pocos encabezados y bajos recursos, logran que los atacantes no puedan capturar la información de la red de los datos entregados a los nodos [40].

Como beneficios agregados de la autenticación, en [36] se reducen los saltos de transmisión de los reportes, mejorando la eficiencia del filtrado de mensajes. En [38], se controlan los ataques *Sybil* utilizando solo criptografía simétrica, pero se debe limitar el número de nodos por grupo a 40 para tener un grado de seguridad aceptable. Y gracias a que no se necesita de un servidor de autenticación, en [35], se reduce el tráfico y consumo

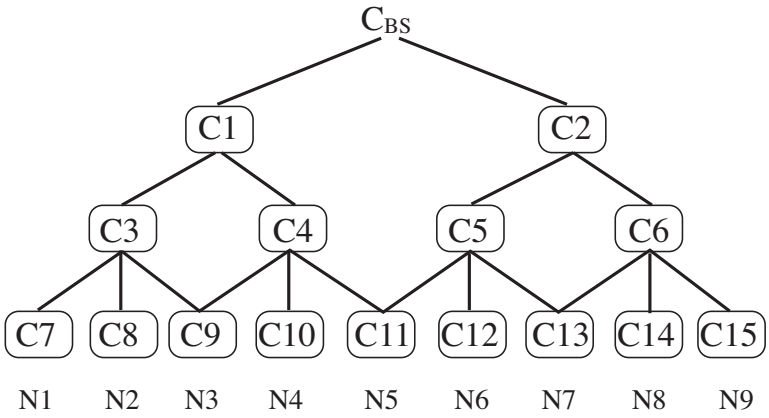

Figura 6. Árbol de claves

de energía, pero se requieren grandes recursos computacionales para los PIV.

Por otro lado, los atacantes pueden identificar la solución del rompecabezas mediante la fuerza bruta [41], pero al demorar largo tiempo su ejecución, es posible renovar el rompecabezas y aumentar el grado de seguridad de la red. Como limitación del algoritmo, se encuentra la longitud del rompecabezas, pues debe ser tal que sea posible su resolución en poco tiempo por parte del transmisor, pero lo suficientemente grande para obtener mayor seguridad.

Como temas abiertos para trabajos futuros, se encuentran los siguientes aspectos:

- Mayor investigación de los sistemas de detección de intrusos, IDS.
- Lograr esquemas que soporten grandes redes.
- Reducir los recursos necesarios para el manejo de PIV.
- Al igual que en la sección anterior, mejorar la capacidad de adicionar y retirar nodos.
- Reducir el costo computacional de la resolución del rompecabezas por parte del transmisor y el retraso generado [41].

V. DETECCIÓN DE INTRUSOS

Debido al despliegue de los sensores en áreas abiertas, donde es posible que un atacante capture un nodo y acceda a toda su información, fácilmente se presentan ataques por clonación, por intrusos o DoS. A los nodos comprometidos se les suele denominar *Moles* [46].

A. Monitoreo de nodos

Para el monitoreo de la red, varios autores postulan el concepto de que los mejores candidatos para proteger el sistema y por ende a los sensores, son los mismos nodos.

Basados en el anterior supuesto, los autores de [47], desarrollan dos algoritmos de autoprotección, donde una serie de nodos se activan para monitorear un área específica de la red. En el primero, denominado Activación Independiente Preprogramada, PIA (*Prescheduled independient activation*), cada sensor tiene una lista predefinida de activación, a la cual se acogen sin tener ningún conocimiento del comportamiento de otros sensores. De esta forma, cuando expira el reloj de un nodo, se activa el sensor con una probabilidad P y reinicia su reloj. Al activarse, busca nodos activos para enlazarse y compartir información de posibles eventos sospechosos, pero si no encuentra, vuelve a un estado de reposo o de sueño. Sin embargo, si ha conformado de manera premeditada, un par de comunicación con otro nodo, ambos se activan en el mismo instante. En el segundo algoritmo, llamado Cooperación de Vecinos, NC (*Neighborhood cooperative*), los nodos poseen una lista de activación distribuida, es decir los nodos se activan teniendo en cuenta el comportamiento de sus vecinos. El nodo pasa por cuatro estados, como se muestra en la Figura 7. Duerme durante un periodo establecido, al terminar el periodo pasa a estado de descubrir, en el cual envía mensajes a sus vecinos para tratar de establecer conexión. Si no encuentra enlace, pasa a un estado de espera, donde sigue enviando mensajes. Cuando

recibe un mensaje de aprobación pasa a estado activo y se conecta.

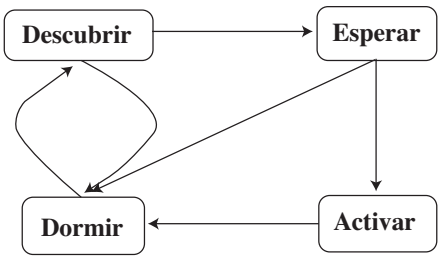

Figura 7. Diagrama de estados del sensor

En [48], presentan el protocolo Aleatorio, eficiente y distribuido RED (*Randomized, Efficient and Distributed*), para detección de ataques. Se fundamenta en el conocimiento que posee cada nodo sobre su localización en la red, y su objetivo es asignar, de manera aleatoria, el rol de testigo entre los nodos para que monitoreen la red. Este protocolo se ejecuta en dos períodos, en el primero se comparte un valor aleatorio a todos los nodos. Para la selección de los testigos se utiliza una función pseudoaleatoria que toma como entrada la ID de los nodos, el número de las localizaciones, es decir, testigos, que serán generados, y el valor aleatorio enviado en el primer paso del algoritmo. En el segundo período, cada nodo firma su proclamación con su ID y su localización y lo envía a un número aleatorio de vecinos, pues no es posible determinar exactamente cuál nodo es un testigo, pues se generan constantemente. Los testigos que reciben la proclamación verifican la firma y observan la identificación del nodo. Si es la primera vez que la reciben la acumulan, de otro modo, comparan la ID recibida y la almacenada, y si coinciden, continúan verificando otros nodos, de lo contrario proceden a la revocación del nodo.

Cada nodo es testigo de la presencia del otro en [49]. Para esto, cada nodo envía periódicamente un *broadcast* a la red que notifique su presencia. De esta forma, la ausencia de mensajes, indica la existencia de un nodo intruso. Se desarrollan dos esquemas de detección. En el esquema de Detección Simple Distribuida (SDD, *Simple Distributed Detection*), cada nodo posee la tarea de monitorear un conjunto de sensores. Si en el rango de cobertura de un sensor A, se introduce un nodo B, A activa el reloj interno de espera para que el nodo B notifique su presencia. Si expira el plazo de espera, A envía una alerta a la red de un nodo intruso. Sin embargo, es necesario que se presente un número de alarmas del mismo nodo para revocarlo, esto evita los falsos positivos. Por otro lado, en el esquema de Detección Cooperativa Distribuida (CDD, *Cooperative Distributed Detection*), dos nodos, A y B, intercambian información y los tiempos de espera para recibir la notificación de los sensores que ambos monitorean y se actualizan con el menor tiempo. Si el tiempo sobrepasa el umbral, proceden de igual manera que SDD.

También es posible realizar un monitoreo solidario entre vecinos. Por ejemplo en [50], se presenta el Protocolo Cooperativo de Seguridad (CSP, *Cooperative Security Protocol*). Cada nodo almacena la información de revocación, la cual es prerrequisito para pertenecer a la red. El protocolo posee cuatro fases. En la primera, se predistribuyen claves de encripción, las claves *Hash* y los árboles *Merkle*. En la segunda fase, se establecen las claves pares con los vecinos y se comparte la información de revoca-

ción para validarla. De esta manera forman el dominio distribuido de seguridad confiable (DTSD, *Dinamic Trusted Security Domain*). En la tercera fase, los vecinos se monitorean entre ellos. La cuarta se presenta cuando se produce un evento sospechoso. Los nodos miembros del DTSD sospechoso votan para definir si un sensor debe ser revocado de la red, luego, revelan e intercambian la información de revocación y los votos en contra del intruso. Entonces, un nodo de la DTSD sospechosa, calcula el voto de revocación definitivo y lo envía como *broadcast*. La BS valida el mensaje y aísla el nodo maligno.

Al igual que en anteriores protocolos, en [51], inicialmente se generan las celdas en la red y se seleccionan los superiores de cada una. Luego se ejecutan cinco pasos. Primero, se intercambian las listas de vecinos entre los nodos, para completar el inventario de los integrantes de cada celda. Segundo, se envían las listas completas de cada celda y se verifican entre los sensores vecinos. Tercero, se elige la lista correcta por unanimidad. Cuarto, se intercambia la última lista para que todos los nodos la actualicen. Quinto, si se encuentran inconsistencias en la última lista transmitida, se verifica de nuevo y si se observa un nodo extraño, se retira. Utilizan nodos que monitorean la red y los datos se analizan de acuerdo con las siguientes reglas: debe existir un periodo entre mensajes consecutivos de un nodo; se debe mandar información de todo evento; si es preciso retransmitir se realiza en un periodo específico; todo nodo solo puede retransmitir un número delimitado de ocasiones, el tamaño del *payload* debe ser constante y el número de colisiones de un mensaje menor que el de toda la red.

B. Sistema de detección de intrusos

Cuando los recursos de los nodos no son tan limitados, es posible utilizar un software especial para la detección de intrusos, denominado IDS (*Intrusion Detection System).*

Los autores de [52], utilizan el Sistema de Detección de Intrusos instalado en pequeños grupos de sensores. Estos nodos, monitorean la red buscando intrusos, manteniendo, durante un período, su área de cobertura en modo promiscuo, es decir, recolectando toda clase de información, para luego enviarla al superior de *cluster*. El modelo IDS desarrollado es distribuido, pues posee cuatro agentes diferentes en toda la red. El agente estático (SA, *Static Agent*), es instalado en cada superior de *cluster* y BS. Genera un evento cuando hay una actividad sospechosa y la envía al servidor de agentes móviles (MAS, *Mobile Agent Server*). El MAS, instalado en la BS, genera agentes Móviles (MA, *Mobile Agent*), para la tarea de detectar intrusos basados en los eventos transmitidos por los SA y los envía como apoyo a los nodos monitores. Los MA, almacenados en los superiores de *cluster* y en los nodos monitores, son los responsables de recolectar la evidencia. Los agentes nodales (NA, *Nodal Agent*), instalados en los nodos monitores, cuando encuentran un evento lo comunican al SA más cercano y tratan de crear acciones que contrarresten el ataque. Cabe señalar que un agente móvil es una entidad de software que funciona de manera autónoma en un ambiente determinado [53].

En [54], se desarrolla una mejora dinámica de los sistemas de detección de intrusos (DIDS *Dinamic IDS*), de manera que se identifiquen usos no autorizados de firmas mediante seis pasos. Primero se forman los *clusters*. Luego, se activa el IDS preinstalado en los superiores de *cluster* y nodos de borde. Tercero, si algún nodo consume el 30% de su energía utilizando el IDS, se reconfigura la red, usando el algoritmo de [55]. Cuarto, se activa el IDS en los nuevos nodos reconfigurados. Quinto, cuando el número de intrusos detectado en un instante es mayor que un umbral establecido, el algoritmo actualiza el núcleo de defensa y la de borde, es decir, los superiores y los nodos límites, para distribuir una defensa con mayor capacidad de detección. Además, activa el mecanismo de remoción de intrusos. Por último, se repiten los pasos desde el tercero.

C. Otros métodos

Diferentes autores desarrollan mecanismos donde se utilizan conceptos como votación, la huella dactilar del nodo, la huella social, árboles jerárquicos, entre otros, que permitan detectar e inclusive revocar nodos corruptos.

Para detectar las posibles *Moles* en la red, en [46], se propone un esquema de marcas anidadas probabilística, que consiste en dos técnicas. En la técnica básica, cada nodo envía su ID con un código de autenticación de mensaje (MAC), utilizando la clave de sesión entre el nodo y la estación base, al pasar al siguiente salto el mensaje es reencapsulado con el ID del nuevo nodo y la MAC. Cuando el mensaje llega a la BS, ésta verifica la concordancia entre el ID del nodo y su MAC. La *Mole* se encuentra entonces, donde se presenta la última MAC correcta de la ruta. En la técnica probabilística, cada nodo marca el paquete con una pequeña probabilidad *P*. Cuando el nodo es legítimo, usa una ID anónima que depende de la clave de sesión con la estación base. Una vez recibe el mensaje, la estación base construye una tabla con las ID anónimas y el nodo al que pertenecen. Luego reconstruye la ruta del mensaje y si encuentra un nodo que suplanta la fuente o que genera en la ruta un lazo, pues el intruso no puede reconocer a quién pertenece cada ID anónima, es una *Mole*.

La votación y el estimador del error cuadrático medio son los fundamentos de los mecanismos de detección de ataques, planteados en [56]. De esta manera, en el primer método, para efectuar la detección de intrusos se crea una lista de las localizaciones de referencia de cada nodo en la red. Luego se hace una comparación de la anterior localización con la estimación que cada nodo posee de su ubicación, mediante el error cuadrático medio. Si el valor entregado sobrepasa un umbral establecido, el nodo es maligno. La anterior verificación se puede realizar con tres métodos diferentes. En el algoritmo de fuerza bruta, cada nodo se compara con todas las localizaciones de referencia. En el algoritmo codicioso, se verifica que las localizaciones estimadas sean consistentes con las de referencia, si se encuentran problemas, se recoge un subconjunto de ubicaciones y las compara con menos localizaciones de referencia, reduce el umbral del error y vuelve a estimar. Repite el proceso hasta encontrar solo las ubicaciones consistentes. Y

en el algoritmo codicioso mejorado, para cada localización de referencia, se encuentran las ubicaciones estimadas que son consistentes con ella (conocido como grado de consistencia) y el nodo que posea menor grado de consistencia es maligno. El segundo esquema es conocido como estimación de localización basado en votación. El área se divide en cuadrados, y para cada localización de referencia, se generan anillos que determinan la posible localización del nodo. A cada cuadrado que se encuentre dentro del área del anillo se le contabiliza un voto. Cuando algunos anillos se traslapan, aumentan la votación de cada división, como se muestra en la Figura 8. La sección donde se encuentra la mayor votación es la localización del nodo. Para obtener una mayor precisión se puede volver a iterar en el sector o disminuir el tamaño de las divisiones.

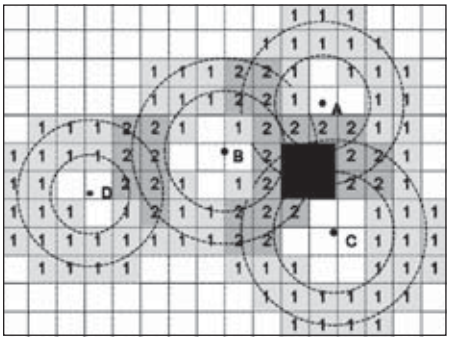

Figura 8. Estimación de localización basada en votaciones [54]

El concepto de huella dactilar es utilizado en [57], la cual contiene las propiedades análogas de las señales generadas por el nodo, como por ejemplo, las características del transiente de la señal. Si el sistema es centralizado, la infraestructura utilizada es un osciloscopio y una antena, ubicadas en varios puntos de la red. La antena captura las señales que comparten dos nodos y mediante el osciloscopio se verifica la huella del sensor que transmite. Si es correcto, verifica que la MAC sea válida. Cuando el sistema es distribuido, cada sensor puede verificar la huella de los nodos vecinos. El proceso de comprobación comienza con el nodo A, que envía un *broadcast* con un número aleatorio, *nonce,* y su ID que le permite descubrir vecinos. El nodo B que recibe el mensaje, confirma que la huella sea de A y si es correcta, transmite un nuevo *nonce* y el del nodo A en una MAC. A revisa la huella de B, y si corresponde, reenvía su *nonce* para que B lo vuelva a verificar.

Los códigos disjuntos superpuestos, que han sido desarrollados ampliamente en [58, 59], se utilizan en [60], para detectar intrusos. Antes del despliegue, se calcula un código disjunto superpuesto (matriz de MxN cuyas columnas son palabras claves y al menos una de ellas intercepta el subconjunto de las columnas que valen cero) y se almacena en cada nodo. Luego, cuando se han ubicado los nodos, cada uno envía un *broadcast* con su código y escucha los que sus vecinos transmiten. Los almacena y procesa un vector, denominado huella social del sensor, con la suma booleana de los códigos o huellas de sus vecinos. El vector debe poseer al menos un elemento con valor igual a cero. Cuando un nodo envía mensajes a sus vecinos, adjunta su huella social, y en caso de ser errónea, los vecinos alertan a la BS. La estación base, entonces, transmite una consulta a la vecindad y cada nodo le responde con su huella social. Compara las diferentes huellas con la lista que almacena

del ID y la correspondiente huella y al encontrar el nodo maligno, envía un mensaje de revocación para que los sensores aíslen al nodo corrupto.

En [61], se propone el esquema SET para la detección de clones en la red. Para escoger al superior de *cluster*, la BS envía una semilla a toda la red. Cada nodo que la recibe calcula la función *Hash* de su ID, y la semilla y el nodo cuya función sea la de mayor peso, se convierte en superior de *cluster*. El superior se anuncia enviando un *broadcast* con su ID, cada nodo que lo recibe pasa a un estado de "reglamentado" y transmite su ID para que el superior los almacene. Luego, cada nodo superior de *cluster* reporta su ID junto a la lista de las ID y MAC, con las claves de sesión, de sus miembros a los grupos vecinos. Los superiores seleccionan un grupo de nodos para realizar un reporte de autenticidad con su ID y la de los nodos y la envía a la BS, la cual, verifica que sean nodos legítimos y realmente pertenezcan al *cluster*. Además un jefe administrador, seleccionado aleatoriamente, comienza la construcción de árboles de los nodos para detectar clones localmente. Se construyen diferentes árboles de manera paralela que conformen los niveles del árbol principal. Para unirlos se comprueba que no haya intersección entre ellos, es decir, no existan clones. Una vez completo el árbol, se realizan reportes entre jefes de nivel. Para esto, un superior calcula una MAC para otro jefe de nivel y otra para el del nivel mayor que continúa la ruta hacia el administrador. Cuando existe un error en la MAC de un jefe, es el superior de nivel mayor el que lo detecta y envía el reporte a la BS.

De manera similar al desarrollo de [56], en [62], se divide el área pero esta vez en triángulos equiláteros de lado $\sqrt{(3r)}$, donde r es el radio de cobertura de un nodo. Los nodos conocen su posición. Poseen BS denominadas "colectoras", las cuales recogen la información del sistema que han capturado los nodos. Se mueven hacia el centro del hexágono que forma la unión de seis triángulos (Figura 9). También se tienen BS llamadas "conectoras", que se mueven a través de la red en línea recta, paralela a uno de los lados de los triángulos y con velocidad constante, recolectando la información de la posición de los nodos y de las BS "colectoras". Las BS "colectoras", se comunican entre ellas periódicamente, intercambiando datos de los nodos sospechosos y transmitiéndola a las "conectoras", las cuales combinan la información de rutas y localización de los nodos maliciosos, para determinar si efectivamente son corruptos. El resultado del análisis lo envían a las BS "colectoras", las cuales ponen en práctica las políticas de seguridad necesarias.

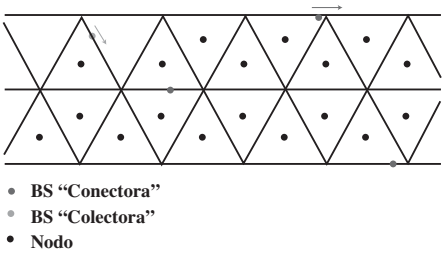

Figura 9. Ruta de las BS "colectoras" y "conectoras"

D. Mecanismos contra ataques DoS

Uno de los ataques más recurrentes es la negación de servicio, DoS, pues logra bloquear un servicio o inclu-

sive, la red completa. Esto debido a que cada nodo del sistema coopera en diferentes actividades y su mal funcionamiento genera una cadena de deterioro en el comportamiento de toda la WSN. Para evitar este tipo de ataque, varios mecanismos se han desarrollado, basados en criptografía, autenticación, rejuvenecimiento o reinicio del sistema, entre otros.

El concepto de rejuvenecimiento y reconfiguración de [63, 64], es utilizado en [65], para proponer cuatro estrategias contra los ataques DoS. La primera se denomina rejuvenecimiento, donde se reinicia el nodo para terminar con una aplicación maliciosa. La segunda es llamada reconfiguración, en la cual se puede cambiar el objetivo del sensado, la topología o el procesamiento de la información, o simplemente, se detiene o cambia la aplicación. La tercera hace una combinación de las dos primeras estrategias, realizando una reconfiguración y luego un rejuvenecimiento.

Por último, en la cuarta estrategia se realiza el proceso inverso de la anterior. La utilización de la tercera o cuarta estrategia logra mejorar el tiempo de vida de los nodos y aumenta el número de nodos que sobreviven a los ataques DoS.

El mecanismo propuesto en [66], consta de dos partes complementarias. En la primera, el nodo instigador, provee su validación que consta del ID, la clave pública y el certificado de la conexión entre los dos anteriores, expedido por la estación base. El nodo instigado verifica el mensaje y si es correcto, transmite un mensaje al nodo instigador para que éste se autovalide descifrando el mensaje con su clave privada. Si es correcto envía la verificación al nodo instigado y se comprueba que es un nodo legítimo. De lo contrario, el nodo es comprometido. Así se logra que el nodo instigador consuma más energía en la validación, y si trata de autenticarse repetidamente, el nodo es ignorado. En la segunda parte, el nodo instigado solo provee su validación cuando el instigador pasa la validación, realizando el mismo procedimiento anterior. Para autenticarse, el nodo instigado puede mandar una firma que contiene un valor conocido calculado, uno generado por un valor en la curva EGG [67] y uno aleatorio, o enviar un mensaje cifrado con una clave efímera, con el ID, el certificado, la clave pública y la clave privada del sensor. Este segundo segmento se puede realizar utilizando el método de la primera cuando los recursos de la red no son muy limitados.

Se utilizan filtros de preautenticación en [68] basados en grupos o en cadenas de claves. Cuando se utilizan los basados en grupos, cada nodo divide sus vecinos en celdas y generan su respectiva clave. Al enviar mensajes, se adicionan valores de compromiso que se validan con las claves de los grupos, así cuando uno de los grupos es sospechoso porque los valores no coinciden con las claves, se subdivide para aislar el nodo malicioso. Para utilizar los filtros basados en cadenas de clave, se generan las claves en un solo sentido, es decir, el receptor del mensaje solo puede leerlo, pero no puede crear nuevos mensajes cifrados en la misma clave. Solo se verifican las claves que son nuevas en la vecindad o que se hayan divulgado en un tiempo menor al establecido. Cuando se agotan las claves de la cadena, se

genera una nueva y se envía a los nodos.

Los autores de [69], presentan el esquema *Seluge*, el cual es una extensión del sistema de código de diseminación de fuente abierta, *Deluge* [70], incluido en *TinyOs* [71]. El mecanismo contra DoS, cuenta con tres pasos. En el primero, se parte el código a transmitir en páginas del mismo largo y cada página, se divide en paquetes de igual tamaño y se les aplica una función *Hash*. Se encapsula en la parte final del paquete de la página inmediatamente superior, como se muestra en la Figura 10. Crean un árbol *Merkle* para la página cero, concatenando los paquetes de la página uno. Cuando se alcanza el último nivel en el árbol, se adiciona la metadata del código [68], la firma del árbol y la del administrador. La BS envía el mensaje como *broadcast* a la red y cuando un nodo lo recibe, verifica el administrador del árbol con el cual se autentica cada nivel del árbol *Merkle*. En el segundo paso, para retransmitir o anunciar una página nueva, el nodo introduce un número único de secuencia y autentica el mensaje con la clave de *cluster*. En el tercer paso, para mitigar los ataques DoS, se utiliza un mecanismo de rompecabezas de mensaje específico [41]. En la configuración, la BS genera una cadena de claves *Hash*, y las predistribuye a los nodos. Dependiendo de la versión del código a transmitir, se utiliza una clave específica. Luego de firmar el mensaje, se adiciona la clave y la solución del rompecabezas que realiza la BS y se le aplica una función *Hash*. El nodo que recibe el mensaje, verifica que la clave del rompecabezas sea correcta y la compara con la solución de la BS.

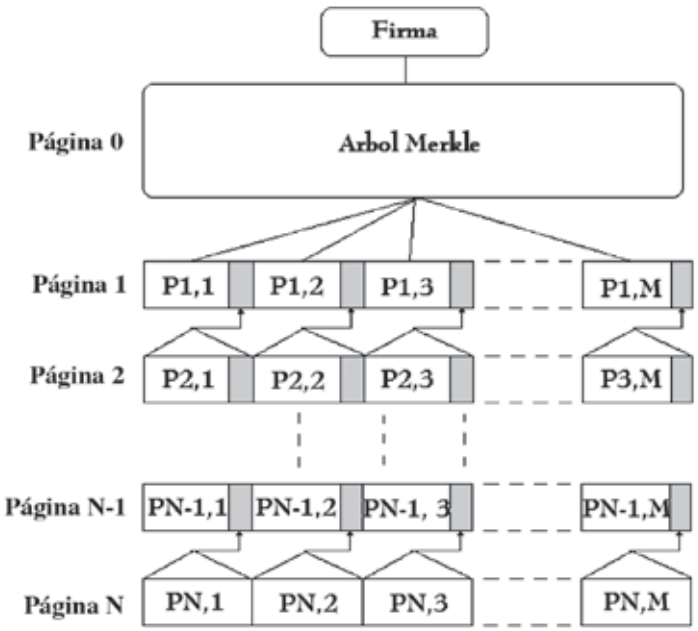

Figura 10. Encapsulamiento de paquetes

E. Análisis de los esquemas de detección de intrusos

En los mecanismos de detección de intrusos, se logra contrarrestar los ataques *Sybil*, gusano, clones, réplicas y DoS. Por ejemplo, en [57], se detectan los cambios en las características físicas de la señal, mas no el contenido y la huella de la señal que es única para cada nodo. El análisis de la huella es robusto, pues es posible separar las señales capturadas en el canal y el encargado es una entidad externa con suficientes capacidades de procesamiento. También se encuentra el mecanismo de votación de [56], pero su precisión, depende del tamaño de las áreas de división del campo, pero no pueden ser tan pequeñas que el procesamiento se torne imposible.

Algunos esquemas utilizan conceptos novedosos para detectar intrusos, entre ellos están la reconfiguración y el rejuvenecimiento [65]. Aumenta el número de nodos que sobreviven a ataques, llegando a estabilizarse cuando se realiza la reconfiguración. La colaboración entre nodos de [49], utiliza propiedades emergentes de los nodos móviles, y es generalizado, es decir, se puede usar en cualquier campo y con diferentes enrutamientos. En [62], se explota el conocimiento de localización de nodos, donde se trabaja como el cuerpo humano (linfocitos), teniendo una redundancia y por tanto mayor seguridad en el reconocimiento y revocación de intrusos. En [52, 54], se utiliza el concepto de movilidad de nodos. En [52], por ejemplo, se emplean agentes móviles que trasladan los recursos de la detección de intrusos entre los nodos con bajos recursos hasta los de más recursos. El sistema es descentralizado por lo que no se recarga la seguridad solo en la BS si no que se escogen varios nodos para monitorear la red. En [54], por su parte, se aumenta la flexibilidad de la capacidad de defensa. La desventaja es que consume mucha energía, pero se trata de contrarrestar, pues, cuando un nodo ha consumido el 30% de la energía comienza la reconfiguración. Los nodos con IDS monitorean de manera equitativa la red por lo que el sistema es muy estable. El concepto de *Mole* de [46], se ve contrarrestado gracias a que con pocas transmisiones, la BS recolecta las marcas de la red y por tanto reconoce los nodos legítimos, lo que se ve reflejado en el poco tráfico. El esquema es para la vecindad de una *Mole*, pero no para un nodo específico. La desventaja es que una *Mole* puede reutilizar un paquete que haya sido enviado anteriormente y sea legítimo.

En [47], existe una buena utilización de la energía. Todo el sistema es complejo y su optimización no puede ser universal. En PIA, los nodos no siempre consiguen sensor para comunicarse, por lo que se pierde cobertura. Contrario, en NC, se tienen menos nodos que no están protegidos y la seguridad mejora con el tiempo. El algoritmo puede ser usado en cualquier campo y topología de la red.

Para el esquema de [50] el máximo número de nodos comprometidos es C=q-1, donde q es mínimo número de nodos necesarios para revocar un intruso. El mínimo tamaño de la DTSD es Φ=q(2q-1). Puede utilizarse también para autenticación, transformar *clusters* en DTSD, para trabajo cooperativo. Se reserva el derecho de admisión de nodos a la red. La ope-

ración distribuida resiste más de *q* nodos comprometidos en la red, pues realizan el procedimiento de revocación de forma más apresurada. La revocación debe ser autenticada por muchos nodos para que sea válida. Las revocaciones se hacen temporalmente y se actualizan, por lo que no existen colisiones ni revocaciones falsas.

En [68], el esquema basado en grupo, no posee tantas validaciones, por lo que el *overhead* y retraso es menor que el basado en cadena. Pero el basado en cadena, no permite a un nodo vecino comprometido deshabilitar la verificación *broadcast*, aun si el transmisor es válido.

El algoritmo de [69] es resistente a los ataques externos DoS. La verificación de los paquetes es sencilla con las funciones *Hash*. Los atacantes internos no pueden forzar las claves de la red. El nodo debe tener memoria suficiente para almacenar mínimo *n* códigos imagen *Hash*. La computación extra se ve reflejada en las múltiples funciones *Hash* que deben realizarse. Posee cierto retraso debido a las funciones *Hash* y a la verificación de las firmas.

Algunos temas para trabajos futuros:

- Corregir el problema de los falsos positivos.
- Diferenciar el tráfico malicioso adicionado del legítimo.
- Realizar simulaciones en ambientes reales [51, 54].
- Realizar avances en la investigación de los nodos móviles.
- Reducir el consumo de energía y costo computacional cuando se utilizan IDS [51].
- Implementar la identificación de la huella dactilar en cada nodo [57].
- Buscar nuevas propiedades para encontrar la huella social y extenderlas a otras redes [60].
- Mejorar la tasa de uso de los mecanismos para optimizar el consumo de energía y costo del sensor [65].

VI. CONCLUSIONES

El artículo muestra el estado del arte de los principales mecanismos de seguridad de las redes de sensores inalámbricos. A continuación se presentan las más importantes conclusiones del tema de estudio.

- La seguridad es uno de los temas más importantes en las redes de sensores inalámbricos, debido a la imposibilidad de utilizar los mecanismos convencionales contra ataques, debido a que los sensores cuentan con recursos de procesamiento y almacenamiento limitados.
- No es posible formular un mecanismo infalible para proveer seguridad en las redes de sensores inalámbricos, es necesario realizar estudios, actualizaciones y evoluciones de los algoritmos que logren contrarrestar los continuos ataques emergentes.
- Para obtener un buen grado de seguridad en las WSN, es necesario ejecutar en conjunto los mecanismos de autenticación, manejo de claves y detección de intrusos en la red.

- Las investigaciones sobre esquemas de seguridad en redes de sensores, deben dirigir sus estudios a encontrar un equilibrio entre seguridad y consumo de energía, reducción de tráfico, disminución del encabezado de los mensajes y utilización proporcionada de los recursos limitados de los sensores.
- Aunque las simulaciones de los algoritmos de seguridad pueden entregar resultados satisfactorios, es en el campo real donde se observa la verdadera eficacia de los mecanismos, por esta razón, el paso a seguir de muchos investigadores es la prueba de sus esquemas en sistemas existentes.
- Los continuos avances en la utilización de las propiedades inherentes de los sensores y de la movilidad de los nodos, ha dado paso a escalar novedosos y más eficientes mecanismos de seguridad. Sin embargo, estos adelantos aún se encuentran en la primera etapa por lo que es un tema de estudio abierto para nuevas y prometedoras investigaciones.

BIBLIOGRAFÍA

[1] K. Papadopoulos, T. Zahariadis, N. Leligou y S. Voliotis. "Sensor Networks Security Issues In Augmented Home Environment". *IEEE International Symposium on Consumer Electronics, 2008*. ISCE 2008. Abril de 2008.

[2] T. Zia y A. Zomaya. "Security Issues in Wireless Sensor Networks". *International Conference on Systems and Networks Communications,* 2006. ICSNC Octubre de 2006.

[3] V. Mhatre, C. P. Rosenberg, y D. Kofman, *et al.* "A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint", *IEEE Transactions on Mobile Computing*. Enero de 2005, Vol. 4.

[4] M. Yarvis, N. Kushalnagar, y H. Singh, *et al.* "Exploiting Heterogeneity in Sensor Networks," *Proc. of IEEE INFOCOM 2005*.

[5] L. Eschenauer y V. D. Gligor. "A key-management scheme for distributed sensor networks". *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 41-47, Noviembre de 2002.

[6] C. Castelluccia y Angelo Spognardi. "RoK: A Robust Key Predistribution Protocol for Multi-Phase Wireless Sensor Networks". *Third International Conference on Security and Privacy in Communications Networks and the Workshops, 2007.* SecureComm 2007.

[7] I. Mironov. "*Hash* functions: Theory, attacks, and applications". Microsoft Research, Silicon Valley Campus. Noviembre de 2005. (En Línea): https://research.microsoft.com/users/mironov/papers/*Hash*_survey.pdf

[8] L. Lamport. "Password authentication with insecure communication". Commun. ACM, vol. 24, no. 11, 1981.

[9] S. Emre Tasçı, E. Bayramoglu y A. Levi. "Simple and Flexible Random Key Predistribution Schemes for Wireless Sensor Networks Using Deployment Knowledge". *2008 International Conference on Information*

Security and Assurance. IEEE 2008.

[10] W. Du, J. Deng, Y.S. Han, S. Chen, and P.K. Varshney. *A key management scheme for wireless sensor networks using deployment knowledge*. IEEE Infocom, 2004.

[11] Y. Ho Kim, H. Lee y D. Hoon Lee. "A secure and efficient key management scheme for wireless sensor networks". *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm* 2007. Third International Conference on Volume , Issue , 17-21 Septiembre de 2007.

[12] J. Young Chun, Y. Ho Kim, J. Lim y D. Hoon Lee. "Locationaware Random Pair-wise Keys Scheme for Wireless Sensor Networks". *Third International Workshop on Security Privacy and Trust in Pervasive and Ubiquitous Computing*. 2007.

[13] M. Meribout y A. Al-Zoubi. "A Recurrent Decentralized Key Management Architecture for Wireless Sensor Network". *Proceedings of the 2nd international workshop on Agent-oriented software engineering challenges for ubiquitous and pervasive computing*. 2008.

[14] U.S. Department of Commerce/National Institute of Standards and Technology. "Data Encryption Standard, (Des)". Federal Information Processing Standards Publication. Octubre de 1999. (En línea): http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[15] J. Luo, P. Papadimitratos y J-P. Hubaux. "GossiCrypt: Gíreles Sensor Network Data Confidentiality Against Parasitic Adversaries". *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks,* 2008. IEEE SECON '08.

[16] X. Yi, M. Faulkner y E. Okamoto. "Securing Wireless Sensor Networks". *The Third International Conference on Availability, Reliability and Security*. IEEE 2008.

[17] MIT open Course Aware. "Chapter 7: Introduction to finite fields". Ingeniería eléctrica y ciencias de la computación. (En línea): http://ocw.mit.edu/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-451Spring-2005/98871616-AC8D-4B31-9989-BB0235B27BDC/0/chap7.pdf

[18] A. Shamir. "How to share a secret", Communications of the ACM, vol. 22, no. 11, pp. 656-715. Nov 1979.

[19] J. Abraham y K. S. Ramanatha. "A Complete Set of Protocols for Distributed Key Management in Clustered Wireless Sensor Networks", *International Conference on Multimedia and Ubiquitous Engineering*. ACM, 2007.

[20] Ossama Younis and Sonia Fahmy. "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks," *IEEE Transactions on Mobile Computing*, volume 3. 2004.

[21] A. Hamid, M. Mahbub Alam y C. Seon Hong. "Developing a

Security Protocol based on LCG and Orthogonal Matrices for Wireless Sensor Networks". *The 9th International Conference on Advanced Communication Technology*. Febrero de 2007.

[22] D.E. Knuth. "Deciphering a Linear Congruential Encryption". *IEEE Transactions on Information Theory*, Vol. IT-X, no. 1, Enero de 1985, pp.49-52.

[23] D.E. Knuth. "The Art of Computer Programming, Vol 2: Seminumerical Algorithms". Ed. Addison-Wesley, 1969.

[24] I. Ingemarsson. "Commutative Group Codes for the Gaussian Channel". *IEEE Transactions on Information Theory*, vol. IT-19, no. 5, pp. 215-219, Marzo de 1973.

[25] K-W. Fan, S. Liu y P. Sinha. "Scalable Data Aggregation for Dynamic Events in Sensor Networks". *Conference On Embedded Networked Sensor Systems. Proceedings of the 4th internacional conference on Embedded networked sensor systems*. Noviembre de 2006.

[26] J. Mache, C-Y. Wan y M. Yarvis. "Exploiting Heterogeneity for Sensor Network Security". *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. SECON 2008.

[27] X. Du, H-H. Chen, Y. Xiao y M. Guizani. "A Pseudo-Random Function based Key Management Scheme for Heterogeneous Sensor Networks". IEEE GLOBECOM 2007 proceedings.

[28] J. Brown, X. Du, y K. Nygard. "An Efficient Public-Key-Based Heterogeneous Sensor Network Key Distribution Scheme". IEEE GLOBECOM 2007 proceedings.

[29] F. Aurenhammer y R. Klein. "Voronoi Diagrams". (En Línea): http://www.pi6.fernuni-hagen.de/publ/tr198.pdf

[30] Y. Sun, J. Zhang, H. Ji y T. Yang. "KMSGC: A Key Management Scheme for Clustered *Wireless Sensor Networks Based on Group-oriented Cryptography". IEEE International Conference on Networking, Sensing and Control, 2008*. ICNSC 2008. Abril de 2008.

[31] W. Du, J. Deng, Y. S. Han, S. Chen, y P. K. Varshney. "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge". INFOCOM 2004. *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*. Marzo de 2004.

[32] Kalpana Sharma, Vikash Varun y Rohit Kumar. "System on Chip for Sensor Network Security: A Proposed Architecture". *10th International Conference on Advanced Communication Technology*, 2008. ICACT 2008.

[33] J. Zhang, X. Liu Y H. Xu. "An Efficient Scheme for Broadcast Authentication in Wireless Sensor Networks". *ASIAN ACM Symposium on Information, Computer and Communications Security. Proceedings of the 2006 ACM Symposium on Informa-*

tion, computer and communications security. 2006.

[34] A. Perrig, R.Szewczyk, V.Wen, D.Culler y J.Tygar. "Spins: Security protocol for sensor networks". *Proceedings of Seventh Annual International Conference On Mobile Computing and Networks*. Julio de 2001.

[35] K. Chang y K. G. Shin. "Distributed Authentication of Program Integrity Verification in Wireless Sensor Networks". Securecomm and Workshops. IEEE, 2006.

[36] Byung Hee Kim y Tae Ho Cho. "Efficient Selection Method of Message Authentication Codes for Filtering Scheme in Sensor networks". *Conference on Ubiquitous Information Management And Communication. Proceedings of the 2nd international conference on Ubiquitous information management and communication*. ACM, 2008

[37] Ye, F., Luo, H., Lu, S. "Statistical En-Route Filtering Of Injected False Data In Sensor Networks", IEEE J. Sel. Area Communication. 23(4), pag 839-850, 2005.

[38] J. Yin y S. Kumar Madria. "Sybil Attack Detection in a Hierarchical Sensor Network". *Third International Conference on Security and Privacy in Communications Networks and the Workshops,* 2007. SecureComm 2007.

[39] R. C. Merkle. "Protocols for public key cryptosystems". *Proceedings of the 1980 IEEE Symposium on Security and Privacy*, 1980.

[40] N. Sultana y E-N. Huh. "An Efficient Scheme for Secure Group Communication in Mobile Wireless Sensor Networks". *Conference on Ubiquitous Information Management And Communication. Proceedings of the 2nd international conference on Ubiquitous information management and communication*. ACM, 2008.

[41] P. Ning, A. Liu y W. Du. "Mitigating DoS Attacks against Broadcast Authentication in Wireless Sensor Networks". *ACM Transactions on Sensor Networks*, Vol. 4, No. 1, Article 1. Enero de 2008

[42] A. Juels y J. Brainard. "Client puzzles: A cryptographic countermeasure against connection depletion attacks". *Proceedings of the 6th Network and Distributed Systems Security Symposium (NDSS'99)*.

[43] M. O. Rabin. "Digital signature and public key functions as intractable as factorization". MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212. Enero de 1979.

[44] A. Hamid, M. O. Rashid y C. Seon Hong. "Defense against Laptop Class Attacker in Wireless Sensor Network". *The 8th International Conference on Advanced Communication Technology*, 2006. ICACT 2006.

[45] A. Menezes, P. Oorschot, S. Vanstone. Handbook of Applied Cryptography, CRC Press, Boca Raton, 1996.

[46] F. Ye, H. Yang, y Z. Liu. "Catching "*Mole*s" in Sensor Networks". *27th International Confe-*

rence on Distributed Computing Systems. 2007.

[47] D. Wang, Q. Zhang y J. Liu. "The Self-Protection Problem in Wireless Sensor Networks". *ACM Transactions on Sensor Networks*, Vol. 3, No. 4, Article 20. Octubre de 2007.

[48] M. Conti, R. Di Pietro, L. V. Mancini y A. Mei. "A Randomized, Efficient, and Distributed Protocol for the Detection of Node Replication Attacks in Wireless Sensor Networks". *The ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2007.

[49] M. Conti, R. Di Pietro, L. V. Mancini, y A. Mei. "Emergent Properties: Detection of the Node-capture Attack in Mobile Gíreles Sensor Networks". *Conference On Wireless Network Security. Proceedings of the first ACM conference on Wireless network security*. 2008.

[50] O. Garcia Morchon, H. Baldus, T. Heer y K. Wehrle. "Cooperative Security in Distributed Sensor Networks". *International Conference on Collaborative Computing: Applications and Worksharing, 2007*. CollaborateCom 2007.

[51] I. Chatzigiannakis y A. Strikos. "A Decentralized Intrusión Detection System for Increasing Security of Networks" *Emerging Technologies and Factory Automation* 2007.

[52] M. Ketel. "Applying the Mobile Agent Paradigm to Distributed Intrusion Detection in Wireless Sensor networks". *Symposium on System Theory*. IEEE, Marzo de 2008.

[53] A. Inge Wang, C-F. Sorensen y E. Indal. Architecture for Heterogeneous Devices", *Proceedings of the 3rd IASTED International Conference on Wireless and Optical Communications* (WOC 2003), Banff, Canada. Julio de 2003.

[54] G. Huo y X. Wang. "DIDS: A Dynamic Model of Intrusión Detection System in Wireless Sensor Networks". *2008 IEEE International Conference on Information and Automation* Zhangjiajie, China. Junio de 2008.

[55] H. Su y X. Zhang. "Energy Efficient Clustering System Model and Reconfiguration Schemes for WSN". *40th Annual Conference on Information Sciences and Systems, 2006*. IEEE 200.

[56] D. Liu, P. Ning, A. Liu, C. Wang y W. Kevin Du. "Attack Resistant Location Estimation in Wireless Sensor Networks". *International Symposium on Information Processing in Sensor Networks, 2005*. IPSN 2005.

[57] K. Bonne Rasmussen y S. Capkun. **"**Implications of Radio Fingerprinting on the Security of Sensor Networks". *International Conference on Security and Privacy in Communications Networks and the Workshops, 2007*

[58] A. G. D'yachkov y V. V. Rykov. "Optimal superimposed codes and designs for renyi's search model". *Journal of Statistical Planning and Inference*. 2002.

[59] K. Xing, X. Cheng, L. Ma, y Q. Liang. "Superimposed code based channel assignment in

multi-radio multi-channel assignmen in multi-radio multi-channel gíreles mesh networks". *MobiCom '07*. 2007.

[60]. K. Xing, F. Liu, X. Chang y D. H.C. Du. "Real-time Detection of Clone Attacks in Gireles Sensor Networks". *The 28th Internacional Conference on Distributed Computing Systems*. IEEE 2008.

[61]. H. Choi, S. Zhu y T. F. La Porta. "SET: Detecting node clines in Sensor Networks". *Third Internacional Conference on Security and Privacy in Communications Netsorks and the Workshops, 2007*. SecureComm 2007.

[62]. P. Yali, D. Jiangang, X. Liqiong y Y. Min. "The Research of Movement Control and Security Detection on Tiered Mobile Network". *Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China*. Junio de 2008.

[63]. C. Hofmeister y J. Purtilo. "Dynamic Reconfiguration in Distributed Systems: Adapting Software Modules for Replacement", *Proceeding Ints Conf. On DCS., 1993.*

[64]. Y. Huang, C. Kintala, N. Lolettis y N. Fulton. "Software rejuvenation: analysis, module and application". *Proceedings of the Int. Sym. On Fault Toleratn Computing. Pasadena, CA., 1995.*

[65]. D. Seong Kim, C. Su Yang, y J. Sou Park. "Adaptation Mechanisms for Survivable Sensor Networks against Denial of Service Attack". *Second Internacional Conference on Availability, Reliability and Security*. ACM. 2007.

[66]. O. Arazil, H. Qi y D. Rose. "A Public Key Cryptographic Method for Denial of Service Mitigation in Gíreles Sensor Networks". IEEE SECON 2007 proceedings.

[67]. A. J. Menezes. Elliptic Curve Public Key Cryptosytems. Boston, MA: Kluwer Academia Publishers, 1993.

[68]. Q. Dong, D. Liu y P. Ning. "Pre-Authentucation Filtres: Providing DoS Resistance for Signatura-Based Broadcast Authentication in Sensor Networks". *Conference On Gireles Network Security. Proccedings of the first ACM conference on Gireles network security*. 2008.

[69]. S. Hyun, P. Ning, A. Liu y W. Du. "Seluge: Secure and DoS. Resistant Code Dissemination in Wireless Sensor Networks". 2008 International Conference on Information Processing in Sensor Networks.

[70]. J. W. Hui y D. Culler. "The dunamic behavior of a data dissemination protocol for network programming at scale". *Proccedings of the 2nd Internacional Conference on Embedded Networked Sensor Systems (SenSys'4), Noviembre de 2004.*

[71]. Tinyos: An open-source OS for the networked sensor regime. http://www.tinyos.net/.

CURRÍCULUM

Catalina Aranzazu Suescún (A'2009) nació en Medellín, Antioquia, Colombia, el 10 de febrero de 1984. Se graduó en el Colegio Suárez de la Presentación, estudió Ingeniería Electrónica en la Universidad de Antioquia sede Medellín, graduada en 2007, cursó la Especialización en Telecomunicaciones de la Universidad Pontificia Bolivariana sede Medellín en 2008, se encuentra en espera de la graduación como especialista y actualmente estudia la Maestría en Telecomunicaciones en la misma Universidad.

Laboró como monitora de Teoría Electromagnética para Bioingeniería en la Universidad de Antioquia. Su proyecto de pregrado se encuentra patrocinado por Colciencias para su completa elaboración como parte de un macro-proyecto de la Universidad de Antioquia. En el primer semestre del año 2008 se desempeñó como docente del área de ingeniería electrónica en la Universidad Católica de Oriente, en el primer semestre del año 2008. Su enfoque en las telecomunicaciones son los sistemas de transmisión de datos inalámbricos.