Evaluación del coste energético de la seguridad en entornos extremo a extremo de sensores IPv6

Jasone Astorga
Dpto. Ingeniería de Comunicaciones
Euskal Herriko Unibertsitatea,UPV/EHU
Email: jasone.astorga@ehu.es

Eduardo Jacob
Dpto. Ingeniería de Comunicaciones
Euskal Herriko Unibertsitatea,UPV/EHU
Email: eduardo.jacob@ehu.es

Marivi Higuero
Dpto. Ingeniería de Comunicaciones
Euskal Herriko Unibertsitatea,UPV/EHU
Email: marivi.higuero@ehu.es

***Resumen*—La integración de sensores y dispositivos de capacidades reducidas en el mundo IP es una de las bases de la Internet de las Cosas (IoT), dando lugar al desarrollo y despliegue de innumerables aplicaciones en las que cualquier entidad IP puede establecer comunicaciones extremo a extremo con los sensores; pero también a nuevas amenazas de seguridad, ya que cualquier atacante en Internet puede tratar de obtener la información recogida por los sensores o modificar su funcionamiento. En este escenario, es imprescindible disponer de información fiable y actualizada a cerca de la legitimidad de cada intento de interacción con un sensor, distinguiendo diferentes niveles de acceso, como por ejemplo, acceso a los datos recogidos por el sensor, capacidad de configurar ciertos parámetros, etc. Nuestra propuesta para hacer frente a esta problemática se denomina Ladon, y es un protocolo de seguridad específicamente adaptado a dispositivos de capacidades reducidas. Ladon está basado en el ampliamente conocido protocolo Kerberos, con dos grandes modificaciones para adecuarlo a las características de los escenarios considerados: soporte de funcionalidades de autorización e independencia de la sincronización de relojes. Hemos evaluado el coste en términos de consumo de energía de la protección de la seguridad de las comunicaciones mediante el protocolo Ladon. Los resultados obtenidos mediante simulación muestran que el consumo es comparable al de otros protocolos con menos funcionalidades y que este consumo se mantiene estable incluso en situaciones de tasas de pérdidas de paquetes elevadas.**

I. Introducción

Hoy en día, gracias a estándares como 6LoWPAN [1], la integración de forma nativa de dispositivos con capacidades tan reducidas como los sensores en el mundo IP es ya una realidad. El hecho de que estos dispositivos sean globalmente direccionables en Internet da lugar a un nuevo concepto de redes de sensores, abriendo la puerta al desarrollo y despliegue de un sinfín de nuevas aplicaciones en las que cualquier entidad IP puede establecer comunicaciones extremo a extremo con un sensor. Además, permite evitar las arquitecturas tradicionales en las que la información recogida por los sensores se almacena en sistemas intermedios, lo cual puede ser problemático en el caso de información crítica o con consideraciones de privacidad especiales. Como resultado, los sensores se convierten en diminutos servidores de información o aplicaciones.

Un ejemplo de las aplicaciones consideradas podría ser un sistema de control de vehículos avanzado. En este caso un coche podría estar equipado con diferentes tipos de sensores, como sensores de presión en las ruedas, a los que podrían acceder diferentes entidades con diferentes niveles de acceso. Por ejemplo, el conductor podría tener acceso a los datos recogidos por los sensores, pero no tendría que poder modificarlos. El personal mecánico encargado del mantenimiento y reparación del vehículo, en cambio, tendría que estar autorizado tanto a acceder a los datos recogidos por los sensores como a modificarlos: resetear ciertos contadores o configurar algún parámetro. Por otra parte, el distribuidor o fabricante del sensor no tendría que tener acceso a ninguno de los datos recogidos por el sensor ni a modificar su configuración, pero podría ser autorizado, de forma puntual, a actualizar el firmware del mismo.

En un escenario de estas características, es esencial garantizar la autenticidad y legitimidad de todos los intentos de interacción con los sensores. Esto conlleva la necesidad de disponer de mecanismos de seguridad que permitan implementar funcionalidades de autenticación y autorización de usuarios remotos, así como garantizar la integridad y confidencialidad de la información transmitida. Sin embargo, los sistemas de seguridad tradicionales, orientados a máquinas de altas prestaciones, no son directamente aplicables a los entornos de sensores, debido a las grandes limitaciones de estos últimos. Concretamente, limitaciones en cuanto a memoria, capacidad de almacenamiento, capacidad de cómputo y sobre todo, suministro de energía, ya que típicamente funcionan gracias a baterías de duración finita y reducida.

En este trabajo presentamos Ladon, un protocolo de seguridad que permite a los dispositivos de capacidades reducidas implementar funcionalidades de autenticación, autorización y establecimiento de claves compartidas extremo a extremo a nivel de aplicación. En cuanto a los mecanismos de autenticación, Ladon se basa en el ampliamente conocido protocolo Kerberos (RFC 4120). Sin embargo, para adecuarlo al ámbito de los sensores, proponemos dos grandes modificaciones con respecto a Kerberos: (1) soporte de funcionalidades de autorización y (2) independencia de la sincronización de relojes. Por último, incluimos un estudio del coste de la implementación de Ladon en los entornos considerados, en términos de consumo de energía.

II. Posicionamiento respecto al estado del arte

La seguridad en redes de sensores ha recibido una gran atención en los últimos años, lo que ha llevado al desarrollo

de numerosos protocolos de seguridad con fines ligeramente distintos. Protocolos como [2]–[6] están orientados a proporcionar integridad y confidencialidad a los datos transmitidos entre vecinos de una red de sensores, así como a garantizar que nodos maliciosos no puedan acceder a la misma. Otros protocolos están orientados al establecimiento de claves criptográficas simétricas entre pares de nodos. Entre ellos se pueden distinguir los que se basan exclusivamente en criptografía simétrica [7], [8] y los que hacen uso de criptografía de clave pública [9], [10]. Con respecto a estos últimos, aunque en general la criptografía de clave pública no se considera adecuada para el caso de las redes de sensores debido a los elevados requerimientos de computo y almacenamiento que implica, en los últimos años han surgido propuestas que hacen uso de funcionalidades reducidas de este tipo de criptografía, y principalmente de la criptografía de curva elíptica (ECC). Por último, en cuanto a la especificación ZigBee-2007 [11], ésta no define un protocolo concreto, sino una arquitectura de capas de comunicaciones y una serie de servicios y mecanismos de seguridad que pueden ser implementados a diferentes niveles de la arquitectura de protocolos.

Sin embargo, todos estos protocolos desarrollados específicamente para redes de sensores se centran en proteger las comunicaciones entre los sensores de una red y la estación base, o en algunos casos, entre dos nodos sensores dentro de la misma red, pero no consideran la posibilidad de que entidades externas a la red de sensores puedan acceder directamente a la información proporcionada por los nodos que la componen.

Respecto a los mecanismos de seguridad tradicionales, orientados a máquinas de altas prestaciones, la implementación de infraestructuras de clave pública presenta grandes retos debido a la complejidad derivada de la adquisición de claves y certificados, la verificación de las listas de revocación, etc. En cuanto a los mecanismos que hacen uso exclusivo de criptografía simétrica, destaca el protocolo Kerberos, un protocolo ampliamente probado y utilizado que aun no habiendo sido diseñado específicamente para entornos de capacidades reducidas, se adapta muy bien a las necesidades que estos plantean, entre otras cosas por su gestión centralizada de las credenciales de autenticación.

No obstante, Kerberos no es directamente aplicable a las necesidades de seguridad planteadas en este trabajo, principalmente por dos razones. Primero, Kerberos utiliza sellos de tiempo para garantizar la frescura de los mensajes, por lo que requiere que los relojes de todas las entidades participantes estén sincronizados entre sí. En un entorno distribuido como el planteado, y que además incluye sensores, es difícil garantizar que los relojes de todas las entidades participantes en la comunicación estén permanentemente sincronizados, lo que puede derivar en la denegación de accesos legítimos. Segundo, Kerberos no soporta funcionalidades de autorización. A pesar de que durante años se han planteado múltiples alternativas para hacer frente a esta necesidad [12], [13], todas ellas han sido diseñadas para máquinas de altas prestaciones y por lo tanto, no son adecuadas para los escenarios aquí considerados. Por ello, basándonos en la arquitectura Kerberos, proponemos un protocolo robusto y sin reloj que permite implementar funcionalidades de autenticación y autorización de manera eficiente.

III. Descripción del protocolo Ladon

Al ser Ladon un protocolo fuertemente basado en Kerberos, es interesante recordar algunos conceptos de este último que se utilizarán más adelante a lo largo del artículo. Cada cliente o servidor se denomina *principal* en Kerberos, y cada *principal* se caracteriza por compartir una clave secreta con el centro de distribución de claves (KDC). El mecanismo de autenticación de Kerberos se basa en la utilización de *tickets*, los cuales son credenciales distribuidas por el KDC que contienen una prueba de la identidad del principal que lo solicitó. Los tickets están cifrados de forma que únicamente las entidades para las cuales están dirigidos son capaces de descifrarlos. Por lo tanto, un cliente que quiera autenticarse frente a un servidor utilizando Kerberos tendrá que presentarle un ticket expedido por el KDC para dicho servicio.

El diseño de Ladon implica la modificación del KDC de Kerberos para incluir dos nuevos repositorios de información: (1) una *Base de Información de Conexiones Activas*, utilizada para determinar la frescura de los tickets y mensajes del protocolo; y (2) una *Base de Información de Autorización*, utilizada para almacenar las políticas de autorización. Además, se han añadido tres nuevos mensajes (LDN_AP_IND, LDN_AP_IND_REQ y LDN_AP_IND_REP) y se ha modificado el significado de algunos campos de los mensajes definidos en Kerberos. Principalmente, se han introducido unos *nonces* (cadenas impredecibles de bits) especiales para evitar la sincronización de relojes. Aunque la sustitución de sellos de tiempo por nonces es una técnica clásica, la dificultad radica en hacerlo sin aumentar el número de mensajes intercambiados por el protocolo. En Ladon esto se consigue mediante la utilización de cadenas de claves de un sólo sentido.

La Figura 1 muestra las principales interacciones de Ladon, mientras que la Tabla I detalla el contenido de los mensajes intercambiados. En [14] se proporciona una descripción detallada del protocolo junto con su validación desde el punto de vista de seguridad y rendimiento en cuanto a la sobrecarga de las comunicaciones y necesidades de almacenamiento y cómputo.

III-A. Fase de autenticación

La fase de autenticación está compuesta por los dos primeros mensajes del protocolo y permite al cliente obtener un Ticket Generador de Tickets o TGT ($Ticket_{TGS}$) y una clave de sesión compartida con el servidor TGS ($K_{C,TGS}$). Para ello, el cliente envía un mensaje especificando su identidad (ID_C), la identidad del TGS (ID_{TGS}), un tiempo de vida ($Lifetime_1$) y un valor nonce utilizado para asociar la petición con su correspondiente respuesta. En Kerberos el tiempo de vida contiene la fecha y hora de expiración solicitada por el cliente; sin embargo, como nuestro protocolo no requiere sincronización de relojes, hemos modificado el significado de

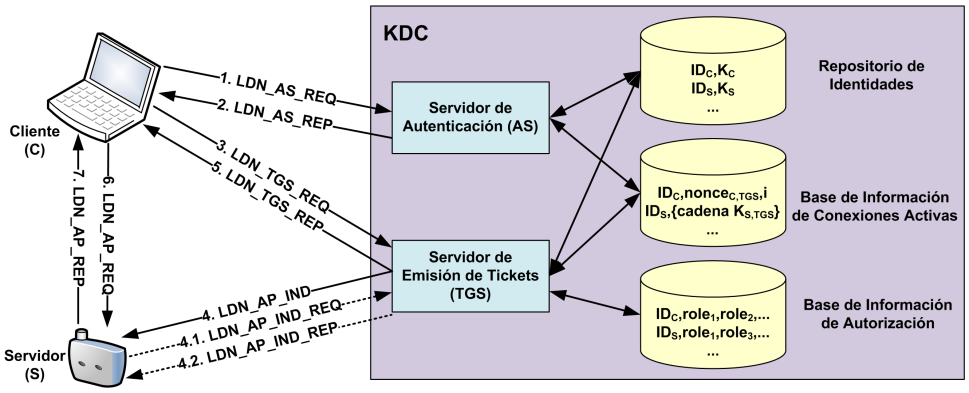


Figura 1: Arquitectura e intercambio de mensajes de Ladon

Cuadro I: Detalle del contenido de los mensajes Ladon

LDN_AS_REQ	$C \rightarrow AS:$	$ID_C \| \| ID_{TGS} \| \| Lifetime_1 \| \| Nonce_1$	
LDN_AS_REP	$AS \rightarrow C:$	$ID_C \| \| Ticket_{TGS} \| \| \{K_{C,TGS} \| \| Nonce_{C,TGS} \| \| Nonce_1 \| \| ID_{TGS}\} K_C$ $Ticket_{TGS} = \{K_{C,TGS} \| \| ID_C \| \| ID_{TGS} \| \| Nonce_{C,TGS}\} K_{TGS}$	*siendo,*
LDN_TGS_REQ	$C \rightarrow TGS:$	$ID_S \| \| Lifetime_2 \| \| Nonce_2 \| \| Ticket_{TGS} \| \| AuthN_{TGS}$ $Ticket_{TGS} = \{K_{C,TGS} \| \| ID_C \| \| ID_{TGS} \| \| Nonce_{C,TGS}\} K_{TGS}$ $AuthN_{TGS} = \{ID_C \| \| Nonce_{C,TGS} + i\} K_{C,TGS}$	*siendo,*
LDN_AP_IND	$TGS \rightarrow S:$	$ID_S \| \| ID_C \| \| Nonce_{C,S} \| \| Lifetime_2 \| \| K^i_{S,TGS} \| \| MAC(K_S, ID_C \| \| Nonce_{C,S} \| \| K^i_{S,TGS} \| \| Lifetime_2)$	
LDN_AP_IND_REQ	$S \rightarrow TGS:$	$ID_S \| \| Nonce_3 \| \| MAC(K_S, ID_S \| \| Nonce_3)$	
LDN_AP_IND_REP	$TGS \rightarrow S:$	$ID_S \| \| K^{i+1}_{S,TGS} \| \| MAC(K_S, ID_S \| \| Nonce_3 \| \| K^{i+1}_{S,TGS})$	
LDN_TGS_REP	$TGS \rightarrow C:$	$ID_C \| \| Ticket_S \| \| \{K_{C,S} \| \| Nonce_{C,S} \| \| Nonce_2 \| \| ID_S\} K_{C,TGS}$ $Ticket_S = \{K_{C,S} \| \| ID_C \| \| ID_S \| \| Nonce_{C,S} \| \| AuthZ\} K_S$ $AuthZ = \{RoleID\} K_S$	*siendo,*
LDN_AP_REQ	$C \rightarrow S:$	$Ticket_S \| \| AuthN_S \| \| Nonce_4$ $Ticket_S = \{K_{C,S} \| \| ID_C \| \| ID_S \| \| Nonce_{C,S} \| \| AuthZ\} K_S$ $AuthZ = \{RoleID\} K_S$ $AuthN_S = \{ID_C \| \| Nonce_{C,S} \| \| Subkey\} K_{C,S}$	*siendo,*
LDN_AP_REP	$S \rightarrow C:$	$\{Nonce_{C,S} \| \| Subkey \| \| Nonce_4\} K_{C,S}$	

este campo para que represente el número de segundos que el ticket será válido después de su creación.

El Servidor de Autenticación (AS) responde con dos instancias de la clave compartida entre el cliente y el TGS ($K_{C,TGS}$): una cifrada con la clave secreta del TGS, y la otra cifrada con la clave secreta del cliente. La respuesta incluye también un valor nonce ($Nonce_{C,TGS}$) que se almacena en la *Base de Información de Conexiones Activas*, junto con la identidad del cliente y un tiempo de vida inicializado con el valor $Lifetime_1$. Cuando este contador expira, la entrada se elimina de la *Base de Información de Conexiones Activas*, lo que evita la reutilización de TGTs antiguos, ya que el TGS sólo acepta TGTs cuyo valor $Nonce_{C,TGS}$ coincida con el almacenado en la *Base de Información de Conexiones Activas*.

III-B. Fase de autorización

La fase de autorización se compone de los mensajes 3-5. El cliente envía una petición LDN_TGS_REQ, en la que incluye el TGT obtenido en la fase anterior ($Ticket_{TGS}$) y un nuevo *Autenticador*, utilizado para evitar que un atacante pueda reutilizar mensajes LDN_TGS_REQ legítimos antiguos. En Ladon, al no existir sincronización de relojes, los autenticadores se basan en un contador mantenido de forma independiente por ambos extremos de la comunicación ($Nonce_{C,TGS} + i$). Cada vez que un cliente necesita generar un nuevo autenticador incrementa en uno el contador. El TGS por su parte, rechaza mensajes con un valor del contador menor del esperado.

Una vez validado el mensaje LDN_TGS_REQ, el TGS pasa a verificar si el cliente está autorizado a acceder al servicio solicitado. Este punto constituye una de las principales diferencias con respecto a Kerberos, donde los Tickets de Servicio se crean para todos los clientes autenticados, sin importar que dichos clientes estén o no acreditados para acceder al servicio deseado. Por lo tanto, mientras que los Tickets de Servicio Kerberos consisten en credenciales de autenticación, los Tickets de Servicio Ladon conllevan tanto autenticación como autorización.

El modelo de autorización propuesto se basa en un diseño combinado de RBAC (Role-Based Access Control) y ABAC (Attribute-Based Access Control). Nuestra propuesta consiste en una arquitectura basada en atributos donde un "rol" no representa un conjunto de permisos, sino el nombre de un

atributo. Cada principal servidor se pre-configura con el valor del rol o roles a los cuales ha de permitir el acceso y únicamente acepta Tickets de Servicio que contengan alguno de estos valores embebido. Además, esta información se cifra de forma individual con la clave secreta del principal servidor para evitar que terceras partes puedan generar su propia información de autorización y enviársela al TGS como información de autorización cifrada a incluir en los tickets, tal y como se define en la RFC de Kerberos v5 (RFC 4120). Por otra parte, nuestro modelo permite también la asignación dinámica de roles a usuarios en función de atributos contextuales, como la hora del día. Las políticas de asignación de roles a usuarios se almacenan en la *Base de Información de Autorización*.

Cada vez que el TGS crea un nuevo Ticket de Servicio, envía un mensaje LDN_AP_IND al principal servidor objetivo, incluyendo toda la información que éste necesitará para validar el mensaje LDN_AP_REQ que recibirá del cliente a continuación. Este mensaje no incluye ningún valor secreto, y por lo tanto, no se cifra. Sin embargo, ha de ser autenticado, para lo que se incluye un código MAC. Una vez validado el mensaje LDN_AP_IND, el principal servidor almacena los valores ID_C y $Nonce_{C,S}$ incluidos en el mismo. No obstante, para evitar desbordar la capacidad de almacenamiento del sensor, si después de un tiempo $Lifetime_2$ no se ha recibido el correspondiente mensaje LDN_AP_REQ, el principal servidor descarta los valores almacenados.

Un aspecto importante del diseño de nuestro protocolo es cómo garantizar que los mensajes LDN_AP_IND no son objeto de ataques de repetición, sin incurrir en un aumento en el número de mensajes intercambiados. Para ello, hemos diseñado un mecanismo basado en cadenas de claves de un solo sentido. Recordar que dada una función de un solo sentido F, es relativamente sencillo calcular hacia adelante (obtener K^{L-1} dado K^L), pero es computacionalmente inviable calcular hacia atrás (obtener K^{L+1} dado K^L). Antes de transmitir el primer mensaje LDN_AP_IND a un cierto principal servidor (S), el TGS genera una cadena de claves de un solo sentido de longitud L. Después, cada vez que el TGS envía un mensaje LDN_AP_IND a S, embebe en dicho mensaje el siguiente valor de la cadena de claves, siguiendo el orden $K^0_{S,TGS}$, $K^1_{S,TGS}$, ..., $K^{L-1}_{S,TGS}$. Por lo tanto, una vez que S cuenta con un valor $K^{i-1}_{S,TGS}$ es suficiente comprobar que $F(K^i_{S,TGS}) = K^{i-1}_{S,TGS}$ para aseverar la frescura del mensaje que transporta dicho valor $K^i_{S,TGS}$. El problema radica en cómo proporcionar a cada principal servidor, de forma autenticada, el valor $K^0_{S,TGS}$ inicial necesario para validar el resto de la cadena de claves. Con este fin, hemos diseñado un mecanismo de inicialización compuesto por dos mensajes (LDN_AP_IND_REQ/_REP) mediante los cuales el principal servidor consulta directamente al TGS a cerca del siguiente valor de la cadena de claves.

Finalmente, el TGS responde al cliente con un mensaje LDN_TGS_REP incluyendo el Ticket de Servicio ($Ticket_S$) y la clave de sesión ($K_{C,S}$) necesarios para acceder al servicio deseado.

III-C. Fase de acceso al servicio

Una vez obtenido el Ticket de Servicio, el cliente establece una conexión segura con el principal servidor deseado mediante el envío de un mensaje LDN_AP_REQ. La validez de los mensajes LDN_AP_REQ se determina en función del valor $Nonce_{C,S}$ y $role$ que incluyen. Básicamente el principal servidor compara el valor $Nonce_{C,S}$ incluido en el Ticket de Servicio y en el autenticador con el valor correspondiente que tiene almacenado, y únicamente acepta el mensaje si los valores coinciden. Con el fin de evitar que los Tickets de Servicio puedan ser reutilizados, una vez que el principal servidor acepta un mensaje LDN_AP_REQ con unos valores $ID_C/Nonce_{C,S}$ determinados, elimina dichos valores de su repositorio local.

Por último, el principal servidor responde al cliente con un mensaje LDN_AP_REP en el que acepta la utilización de la clave propuesta en el mensaje de petición ($subkey$) o le propone una nueva. Esta clave puede utilizarse a continuación para derivar claves de cifrado y de integridad que sirvan para proteger las subsiguientes comunicaciones entre dicho par cliente/servidor. El uso concreto de esta clave dependerá de la aplicación específica que se esté ejecutando en el sensor.

III-D. Mecanismos de recuperación

Para garantizar la frescura de los mensajes y evitar así ataques de repetición, Ladon se basa principalmente en información compartida entre los dos extremos de una comunicación. Sin embargo, este mecanismo presenta problemas cuando algún mensaje se pierde o corrompe durante su transmisión. Para hacer frente a este problema, cada vez que un cliente envía un mensaje LDN_AS_REQ, LDN_TGS_REQ o LDN_AP_REQ, inicia un contador y si éste expira antes de que se haya recibido la respuesta correspondiente, el cliente envía una petición nueva. Esta operación se repite hasta un máximo número de reintentos dado, después del cual, si la respuesta sigue sin llegar, se genera un error.

En el caso de los mensajes LDN_AP_IND el mecanismo anterior no es válido, ya que estos mensajes no cuentan con una respuesta asociada. De hecho, la pérdida de algún mensaje de este tipo la detecta el principal servidor cuando recibe un LDN_AP_IND con un valor $K^{i+1}_{S,TGS}$ que no cumple la condición $F(K^{i+1}_{S,TGS}) = K^i_{S,TGS}$. En tal caso, el servidor sigue aplicando la función de un solo sentido sobre el valor recibido hasta un número máximo de intentos, y si cualquiera de los valores obtenidos coincide con el valor $K^i_{S,TGS}$ almacenado, acepta el mensaje y actualiza el valor $K^i_{S,TGS}$ almacenado con el valor $K^{i+1}_{S,TGS}$ recibido. Es decir, supongamos que el principal servidor tiene un valor $K^i_{S,TGS}$ cuando recibe un mensaje LDN_AP_IND con el valor $K^{i+2}_{S,TGS}$, ya que el mensaje que contenía el valor $K^{i+1}_{S,TGS}$ se ha perdido durante la transmisión. En este caso, $F(K^{i+2}_{S,TGS})$ no se corresponderá con el valor almacenado ($K^i_{S,TGS}$). Sin embargo, el principal servidor validará el mensaje recibido comprobando que $F(F(K^{i+2}_{S,TGS})) = K^i_{S,TGS}$. Este mecanismo de recuperación es más eficiente que cualquier otro

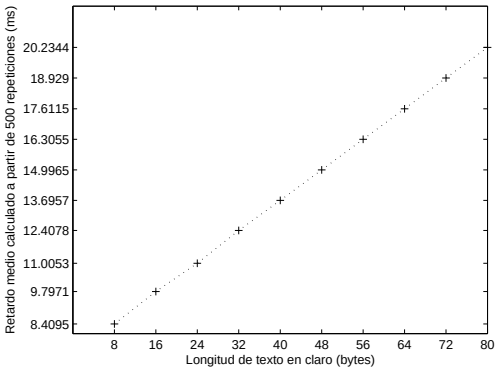


Figura 2: Tiempo medio de cifrado en un sensor TelosB

procedimiento que implique retransmisiones de mensajes.

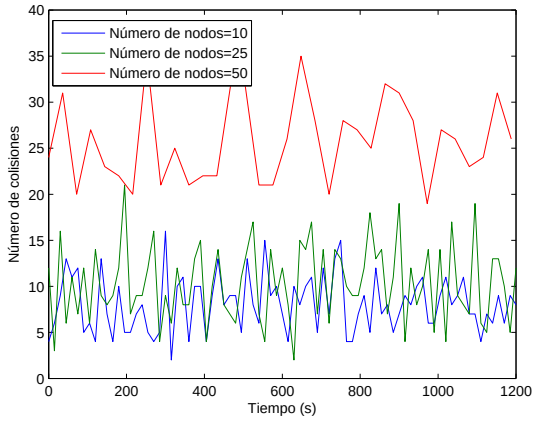
No obstante, en caso de tasas de pérdidas de paquetes altas, puede ocurrir que aún después de aplicar la función de un solo sentido sobre el valor $K^j_{S,TGS}$ recibido hasta el número de intentos máximo definido, el principal servidor no pueda validar el mensaje recibido. En este caso, será necesario ejecutar un mecanismo de recuperación más costoso que consiste en el intercambio de los mensajes LDN_AP_IND_REQ/_REP.

IV. Evaluación de rendimiento

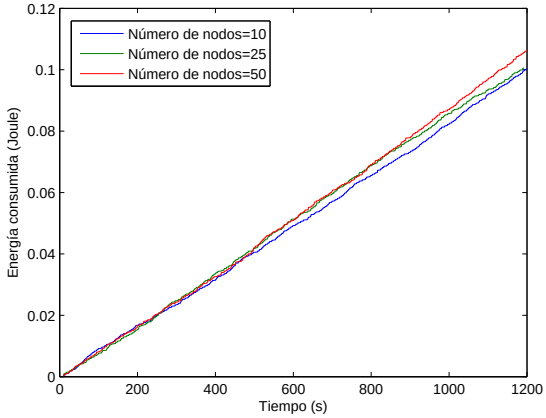
Se ha llevado a cabo un estudio cuyo objetivo es cuantificar el impacto de la implementación de comunicaciones autenticadas y autorizadas en entornos de sensores mediante el protocolo Ladon desde el punto de vista del consumo de energía. En [14] ya se probó que el impacto en cuanto a sobrecarga de las comunicaciones y necesidades de almacenamiento y cómputo es reducido, y se obtuvo un valor adecuado para el número máximo de intentos de validación de un mensaje LDN_AP_IND mediante la aplicación de la función de un solo sentido, antes de iniciar el intercambio LDN_AP_IND_REQ/_REP (5 intentos).

Para llevar a cabo la evaluación de rendimiento hemos utilizado el simulador OPNET [15], un potente simulador de redes basado en eventos discretos. Con respecto a la implementación de nuestro protocolo, debido a las limitaciones de los dispositivos considerados, hemos optado por una implementación binaria, en lugar de seguir la codificación ASN.1. Esto permite reducir el tamaño de los mensajes enviados y recibidos por los sensores en un 90 % aproximadamente.

La energía consumida se ha calculado en base al tiempo que el sensor se encuentra en cada uno de los estados inactivo, realizando operaciones criptográficas, enviando o recibiendo; y al consumo del sensor en cada uno de dichos estados. Para simplificar el cálculo se ha considerado que la tasa de cifrado y descifrado es la misma en un mismo dispositivo. Además se distinguen únicamente dos tipos de dispositivos: dispositivos de altas prestaciones (cliente y KDC) y dispositivos de bajas prestaciones (sensor). El coste asociado a operaciones como concatenación de datos o comparación de patrones se ha considerado insignificante y no se ha incluido en el cálculo.



(a) Número de colisiones



(b) Energía consumida

Figura 3: Energía consumida y colisiones a lo largo del tiempo

En el caso de los dispositivos de altas prestaciones, se ha considerado una tasa de realización de operaciones criptográficas de 1Mbps, lo cual es un valor más que aceptable para cualquier PC actual [16]. En cambio, para calcular la tasa de cifrado de un sensor, hemos medido de forma práctica el tiempo empleado por un sensor TelosB en cifrar textos de diferentes tamaños mediante RC5 CBC y claves de 128 bits. Los resultados obtenidos se muestran en la Figura 2. En cuanto al consumo en los modos de transmisión, recepción, ejecutando cálculos e inactivo, se han tomado como referencia los valores indicados en la hoja de especificaciones del TelosB.

La Figura 3b muestra el consumo de energía a lo largo del tiempo de un sensor que implementa Ladon y al cual se están enviando peticiones de servicio con una distribución de Poisson de media 5s. Esta evaluación se ha llevado a cabo en 3 escenarios independientes respresentando redes de 10, 25 y 50 nodos respectivamente, todos ellos recibiendo peticiones de servicio con la misma tasa (Poisson de media 5s). Como puede observarse en la Figura 3a, al aumentar el número de dispositivos aumenta también el número de colisiones, con lo que cabría esperar un incremento en el consumo

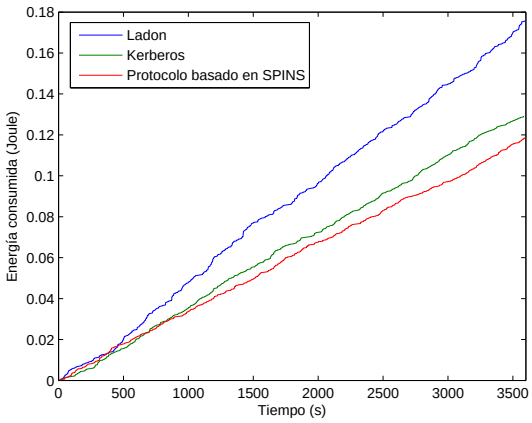


Figura 4: Comparación de consumo de diferentes protocolos

de energía debido a una mayor tasa de envío de mensajes LDN_AP_IND_REQ/_REP para hacer frente a la perdida de mensajes LDN_AP_IND consecutivos. Sin embargo, el consumo es prácticamente idéntico en los 3 escenarios, lo cual demuestra que el mecanismo diseñado para hacer frente a la pérdida de mensajes es eficiente y adecuado para redes con tasas de pérdidas altas, como las redes de sensores comunes.

Con el objetivo de comparar Ladon con otras propuestas, hemos modelado también Kerberos y el protocolo propuesto en [5] para el establecimiento de claves simétricas en redes de sensores. El escenario de simulación utilizado en este caso consiste en un cliente enviando peticiones a 10 servidores en base a una distribución de Poisson con media 1s. Los resultados recogidos en la Figura 4 muestran cómo aunque el consumo de Ladon es mayor que el de los otros dos protocolos, sigue siendo reducido y aceptable para los entornos considerados. Más aun teniendo en cuenta que Ladon proporciona la funcionalidad adicional de autorización con respecto a los otros dos protocolos con los que se compara.

En el caso de las otras dos alternativas consideradas sería necesario llevar a cabo comprobaciones adicionales para alcanzar el mismo nivel de seguridad que el proporcionado por Ladon, por ejemplo, mediante la consulta de listas de control de acceso (ACLs) o bases de datos externas. La implementación de ACLs implica varias desventajas, como el consumo de parte de la escasa memoria ROM disponible y el hecho de que cada vez que se quiera llevar a cabo alguna modificación en la lista es necesario interactuar con el sensor. La consulta de bases de datos externas es también desaconsejable, ya que supone un aumento en el número de mensajes enviados/recibidos por el sensor, lo cual redunda en un aumento de la energía consumida. Además estas consultas deberían ser protegidas también mediante algún mecanismo de seguridad adecuado, lo cual implicaría una mayor sobrecarga de las comunicaciones.

V. Conclusiones

En este trabajo se han expuesto las necesidades de seguridad que es necesario atender antes de que dispositivos de capacidades reducidas como los sensores puedan integrarse de forma masiva en el mundo IP. Con el objetivo de hacer frente a esta necesidad proponemos el protocolo Ladon, el cual está basado en Kerberos pero implica dos modificaciones importantes: (1) evita la necesidad de mantener relojes sincronizados y (2) proporciona funcionalidades de control de acceso.

El principal objetivo de este protocolo es liberar a los dispositivos de capacidades reducidas de las tareas relativas a la implementación de mecanismos de seguridad en la mayor medida posible. Para ello se basa en una arquitectura centralizada, de forma que es posible llevar a cabo tareas como la revocación de credenciales en tiempo real, la definición de políticas de acceso multi-nivel complejas y la modificación de las mismas, sin necesidad de interactuar con los servicios afectados.

Gracias a los resultados de simulación obtenidos, hemos demostrado que el protocolo cumple con los objetivos planteados manteniendo un consumo de energía adecuado incluso en entornos con tasas de pérdida de paquetes elevadas.

Agradecimientos

El trabajo descrito en esta publicación ha sido generado en la Unidad de Formación e Investigación UFI11/16 financiada por la UPV/EHU.

Referencias

[1] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, RFC 4944, 2007.
[2] C. Karlof, N. Sastry, D. Wagner, “TinySec: A Link Layer Security Architecture for Wireless Sensor Networks”, in: Proc. 2nd Int. Conf. Embedded Networked Sensor Systems (SenSys2004), pp.162–175, 2004.
[3] L. E. Lighfoot, J. Ren, T. Li, “An energy efficient link-layer security protocol for wireless sensor networks”, in: Proc. IEEE Int. Conf. Electro/Information Technology, pp.233–238, 2007.
[4] M. Luk, G. Mezzour, A. Perrig, V. Gligor, “MiniSec: A secure sensor network communication architecture”, in: Proc. 6th Int. Conf. Information Processing in Sensor Networks (IPSN’07), pp.479–488, 2007.
[5] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, D. E. Culler, “SPINS: security protocolos for sensor networks”, ACM Wireless Networks, vol. 8, no. 5, pp.521–534, 2002.
[6] K. Ren, W. Lou, Y. Zhang, “LEDS: Providing location-aware end-to-end data security in wireless sensor networks”, IEEE Transactions on Mobile Computing, vol. 7, no. 5, pp.585–598, 2008.
[7] S. Zhu, S. Setia, S. Jajodia, “LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks”, in: Proc. 10th ACM Conf. Computer and Communications Security (CCS’03), pp.62–72, 2003.
[8] T. Park, K. G. Shin, “LiSP: A lightweight security protocol for wireless sensor networks”, ACM Transactions on Embedded Computing Systems, vol. 3, no. 3, pp.634–660, 2004.
[9] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, P. Kruus, “TinyPK: Securing sensor networks with public key technology”, in: Proc. 2nd ACM Workshop Security of Ad Hoc & Sensor Networks, pp.59–64, 2004.
[10] R. A. Shaikh, S. Lee, M. A. U. Khan, Y. J. Song, “LSec: Lightweight security protocol for distributed wireless sensor network”, in: Proc. 11th IFIP Int. Conf. Personal Wireless Communications (PWC’06), pp.367–377, 2006.
[11] ZigBee Alliance, “ZigBee-2007 specification”, 2008.
[12] P. Kaijser, T. Parker, D. Pinkas, “SESAME: the solution to security for open distributed systems”, Computer Communications, vol. 17, no. 7, pp.501–518, 1994.
[13] G. H. Wettstein, J. Grosen, “IDfusion, an open-architecture for Kerberos based authorization”, in: Proc. AFS and Kerberos Best Practices Workshop, 2006.
[14] J. Astorga, E. Jacob, M. Huarte and M. Higuero, “Ladon: end-to-end authorization support for resource-deprived environments”, accepted for publication in IET Information Security Journal
[15] OPNET, http://www.opnet.com/
[16] http://www.cryptopp.com/benchmarks-amd64.html