Control de Acceso para Mensajes Pro-activos en Redes DTN

Adrián Sánchez-Carmona, Carlos Borrego, Sergi Robles and Jordi Andújar
Department of Information and Communications Engineering (dEIC)
Universitat Autònoma de Barcelona
08193 Bellaterra, Spain
Email: adria.sanchez@deic.uab.cat

***Resumen*—Las redes tolerantes a retrasos e interrupciones (DTN) no admiten, por su naturaleza, los protocolos de la familia TCP/IP para el transporte de información. La mayoría de protocolos de encaminamiento que se han desarrollado para este tipo de redes no tienen en cuenta la estrecha relación entre la aplicación y la propia red. Nuestra propuesta al respecto, basada en mensajes que incluyen su propio código de encaminamiento, necesita que en cada nodo estos mensajes puedan consultar informaciones estrechamente relacionadas con la aplicación a la que pertenecen. En este artículo se exploran las necesidades del control de acceso a esta información y se presenta una propuesta basada en la identidad de los mensajes que garantiza la confidencialidad y la integridad.**

***Index Terms*—DTN, Seguridad, Control de acceso.**

I. Introducción

En el contexto de las redes tolerantes a interrupciones, la no contemporaneidad de las comunicaciones, así como la prevalencia de interrupciones, desconexiones, grandes retrasos e intermitencias, hace necesario el uso de algoritmos de encaminamiento de tipo *store-carry-forward*. Las propuestas del DTNRG [1] para estas redes parten de un enfoque continuista de TCP/IP: *Bundle Protocol* [19] proporciona una arquitectura adaptada a las necesidades de estas redes, mientras que *Licklider Transport Protocol* [18] se apoya en éste para garantizar la transmisión de la información más importante en las comunicaciones punto-a-punto.

En cuanto al encaminamiento, las diferentes propuestas adoptan diversos enfoques del problema. *PRoPHET* [17] o *MaxProp* [9], deciden la ruta a seguir en función de la probabilidad de que los nodos puedan entregar cada mensaje a su destino. Otras como *Spray and Wait* [20] optan por un encaminamiento de tipo epidémico [21] limitando el número de copias de cada mensaje que circula por la red.

En general, la mayoría de las propuestas tratan de establecer una estrategia lo suficientemente genérica como para que sea válida en cualquier situación o escenario. Sin embargo, no tienen en cuenta un aspecto que en este tipo de redes puede tener una gran importancia: la estrecha relación entre las aplicaciones y el uso que éstas hacen de la red.

En el otro extremo podemos encontrar propuestas como *Haggle* [11], un protocolo orientado al intercambio de archivos que utiliza las preferencias de los usuarios para decidir los archivos que se envían en cada contacto.

El uso del código móvil nos permite generar mensajes que contengan su propio código de encaminamiento, así podemos ofrecer a las aplicaciones la oportunidad de diferenciarse entre ellas y utilizar una misma red de formas diferentes. Para ello es necesario almacenar en los nodos información que pueda ser utilizada por las aplicaciones durante el encaminamiento de sus mensajes. En este punto se presenta un problema de seguridad importante. Estas informaciones destinadas a ser utilizadas por las diferentes aplicaciones durante el encaminamiento de sus mensajes pueden ser un objetivo muy apetecible para un hipotético atacante que quiera perturbar el correcto funcionamiento de la red. Por lo tanto, se hace necesario controlar el acceso a la información para evitar que ésta pueda ser consultada y/o modificada por un atacante no autorizado.

En este artículo analizamos este problema de seguridad y proponemos un sistema de control de acceso basado en identidad que garantiza la confidencialidad y la integridad de la información utilizada durante el encaminamiento. Para ello, implementamos el control de acceso durante el encaminamiento de forma que cuando un mensaje solicita acceso a una información, se aplican dos funciones de *hash* diferentes sobre el propio código que solicitó dicho acceso. Los valores de *hash* obtenidos son utilizados para recuperar la clave simétrica que permite descifrar la información.

El resto del artículo se organiza de la siguiente forma: En la Sección II se explican las redes DTN basadas en mensajes pro-activos. En la Sección III se estudia la mejor forma de identificar y autenticar a aquellos actores autorizados a acceder a determinadas informaciones. Posteriormente, en la Sección IV se describe formalmente nuestra propuesta de control de acceso basado en identidad para, en la Sección V, discutir los principales aspectos de seguridad relacionados con ésta. En la Sección VI se muestra una aplicación de nuestra propuesta en el encaminamiento en redes DTN. Finalmente, la Sección VII concluye el artículo.

II. Una necesidad a cubrir

En esta sección presentaremos el entorno de nuestra investigación, haciendo hincapié en el mensajes pro-activo como concepto que articula todo un paradigma de encaminamiento

Este trabajo ha sido financiado parcialmente por el Ministerio de Ciencia e Innovación a través del proyecto de referencia TIN2010-15764.

en redes DTN, veremos la utilidad de disponer de información ligada a las aplicaciones a la hora de encaminar y la necesidad de establecer un sistema de control de acceso para esta información.

II-A. DTN basada en mensajes pro-activos

Nuestra propuesta de solución para el encaminamiento en las redes de tipo DTN consiste en cambiar el enfoque tradicional y trasladar el algoritmo de encaminamiento de los nodos a los mensajes, utilizando código móvil tal y como se sugiere en [10], [8]. En este paradigma, los nodos proveen la infraestructura necesaria para que los mensajes puedan decidir el camino a seguir. El concepto clave es el de **mensaje pro-activo**.

Un mensaje pro-activo contiene cuatro campos (Figura: 1): el campo de origen; el campo de destino destino; el campo de datos, donde se almacena el contenido del mensaje; y el campo de encaminamiento, que contiene el código de encaminamiento que se debe ejecutar en cada nodo y que se encarga de decidir la ruta que seguirá el mensaje para llegar a su destino. De ahora en adelante, denotaremos c_i como el código de encaminamiento de un mensaje pro-activo i.

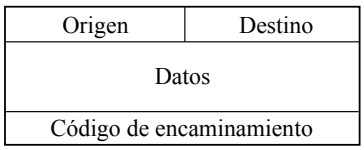

Figura 1. Esquema de los campos de un mensaje pro-activo.

II-B. Encaminamiento dependiente de la aplicación

La gran virtud de esta aproximación consiste en que se permite a las aplicaciones (que son las que mejor conocen sus propias necesidades) que utilicen la red de la forma que más les convenga, esto es, tomando aquellas decisiones de encaminamiento que les resulten óptimas.

Pero los mensajes pro-activos no pueden tomar estas decisiones si no disponen de información suficiente para ello. En una misma red pueden convivir aplicaciones muy diferentes, que necesiten informaciones de naturalezas muy diferentes para tomar estas decisiones. Por lo tanto es necesario que en cada nodo se ofrezca a estos mensajes la posibilidad de consultar o modificar un conjunto de información relacionada con la aplicación a la que pertenece el mensaje. A partir de ahora denotaremos I_j al conjunto de información referente a la aplicación j.

II-C. La necesidad del control de acceso

No podemos permitir que el código de encaminamiento de cualquier mensaje pro-activo c_i que llegue a un nodo pueda acceder y modificar cualquier información I_j. Sin un control de acceso apropiado, cualquier mensaje podría eliminar o modificar de forma maliciosa las informaciones almacenadas en los nodos. De esta forma un atacante podría perjudicar a las aplicaciones que utilizan esa información durante su encaminamiento.

Por lo tanto, es necesario desarrollar un sistema que garantice que solo el código c_i de aquellos mensajes autorizados por la aplicación j tengan acceso a la información I_j. Este problema hay que abordarlo teniendo en cuenta que:

- Un mensaje pro-activo i puede estar autorizado a acceder a la información I_j de una o más aplicaciones j diferentes.
- Uno o más mensajes i diferentes pueden estar autorizados a acceder a la información I_j de una aplicación j determinada.

III. Identidad del mensaje pro-activo

En esta sección veremos las necesidades concretas que se nos presentan a la hora de identificar aquellos mensajes pro-activos a los que hay que permitirles el acceso a una determinada información, discutiremos las diferentes formas posibles de autenticación para estos mensajes y veremos las ventajas que presenta nuestra solución con respecto a otras.

III-A. Identificación de los mensajes pro-activos

Supongamos que un mensaje i llega a una nodo y quiere acceder a la información de la aplicación j. Es necesario identificar y autenticar este mensaje para decidir si tiene autorización para realizar dicho acceso.

La cantidad de mensajes pro-activos i diferentes que puede generar una aplicación j es infinita, ya que el campo de datos puede contener cualquier valor. Sin embargo, durante el encaminamiento estos datos no tienen ninguna importancia, ya que serán entregados a su destinatario cuando corresponda, pero no se utilizarán ni jugarán ningún papel hasta que no lleguen a su destino.

Hay que garantizar que solo accedan a una información I_j los diversos c_i que estén autorizados para ello. No importan los datos que estos mensajes contengan, ya que un atacante que alterara un mensaje i modificando los datos que éste contiene no comprometería en ningún caso la información I_j almacenada en los nodos. En todo caso, podría comprometer la seguridad de la aplicación j, pero ese es un aspecto que queda fuera del alcance de este artículo. Sin embargo, un atacante que alterara un mensaje i modificando el código c_i de éste podría, desde este nuevo código c_i' acceder o modificar una información protegida I_j y comprometerla.

III-B. Autenticación de los mensajes pro-activos

En general, la autenticación se puede encarar desde cuatro perspectivas diferentes:

1. Saber algo que nadie más puede saber, como por ejemplo, una contraseña.
2. Tener algo que no tenga nadie más, como por ejemplo, una llave.
3. Saber hacer algo de una forma única, como por ejemplo, una firma.
4. Tener alguna característica única, como por ejemplo, una huella dactilar o una cadena de ADN.

Veamos ahora por qué las tres primeras aproximaciones no son válidas si trasladamos el problema de la autenticación a los mensajes pro-activos en redes de tipo DTN.

Los dos primeros puntos se pueden tratar conjuntamente, ya que un para una entidad software que viaja por la red es lo mismo *tener algo* y *saber algo*. Un mensaje no puede acarrear elementos físicos como una tarjeta, pero sí que puede llevarse consigo cualquier tipo de datos, como por ejemplo, una contraseña. Sin embargo la utilización de contraseñas presenta un problema importante, el robo de una contraseña deja al sistema comprometido de forma inmediata, ya que a partir de ese momento un atacante podría utilizar esta contraseña para hacer que otro mensaje pro-activo malicioso ganara acceso a una información sobre la que no debería tenerlo y desde su código c_i podría consultarla o modificarla.

La tercera aproximación nos hace pensar directamente en el uso de una técnica criptográfica ampliamente conocida y utilizada: la firma digital. Sin embargo, la firma digital se apoya en la *Public Key Infrastructure* (PKI), un esquema que no se puede aplicar en redes de tipo DTN, ya que ciertos elementos clave como son el acceso a las listas de revocación de certificados o la propia distribución de los certificados de confianza entre los diferentes nodos, son problemas no resueltos ([7], [12], [6]). Esto se debe a que dichos mecanismos parten de una serie de asunciones tales como una conectividad permanente punto a punto, o retrasos pequeños a nivel de capa de enlace que no son propias de las redes DTN.

La solución a nuestro problema la encontramos en el cuarto punto, cuando un código c_i intente acceder a una información I_j, analizando el propio c_i podremos decidir si se le permite o no el acceso a dicha información. En este caso estaríamos utilizando algo inherente a c_i, algo que no se puede robar o copiar y que forma parte de lo que c_i es, podríamos decir, realizando un símil con los sistemas de control de acceso basado en identidad convencionales, que utilizamos el ADN del c_i para identificarlo y autenticarlo.

III-C. Identificando el código en función de su ADN

El código c_i de cualquier mensaje puede tener una extensión significativa. A la hora de trabajar con el código es preferible que este tamaño sea manejable. Basta con utilizar una función *hash* sobre el propio c_i para obtener una secuencia binaria que lo identifica, a la vez que lo diferencia de cualquier otro.

Aunque la comunicación de un mensaje i fuera interceptada y su c_i fuera robado o copiado, la única forma que tendría un atacante de utilizar los datos obtenidos para acceder a la información pasaría por crear un mensaje i' con el c_i obtenido. Sin embargo, esta operación no comprometería en ningún caso la seguridad del sistema, ya que el código c_i del mensaje i' es exactamente el mismo del mensaje i original, está autorizado a trabajar con la información I_j y no realiza sobre ella ninguna acción que pueda comprometer la seguridad del sistema (de no ser así, no habría sido autorizado a utilizar I_j).

IV. Control de acceso basado en identidad

En esta sección analizaremos los requisitos que debe cumplir nuestro sistema de control de acceso y nos adentraremos su formulación. Para ello veremos primero los dos conjuntos (el de reglas y el de informaciones) necesarios y posteriormente el algoritmo encargado de construir dichos conjuntos. Finalmente veremos la forma de utilizar ambos conjuntos para realizar un control de acceso efectivo.

IV-A. Requisitos

El sistema que hemos diseñado utiliza un valor *hash* del c_i de los mensajes pro-activo para identificarlos y controlar el acceso por parte de éstos a información I_j protegida almacenada en los nodos. Conseguir esto no es trivial ya que es imprescindible cumplir con los siguientes requisitos:

- El sistema debe garantizar el acceso a la información I_j a los c_i autorizados.
- El sistema debe garantizar el secreto y la integridad de dicha información I_j, evitando que cualquier c_i no autorizado pueda acceder a ella o modificarla.
- El sistema debe tener un impacto mínimo en consumo de recursos y tiempo de ejecución para evitar entrar en conflicto con las ventanas de conectividad de las redes DTN, que pueden ser muy pequeñas.

IV-B. Un sistema basado en reglas

Nuestra propuesta se basa en la utilización de dos conjuntos, el conjunto de reglas de acceso y el conjunto de informaciones almacenadas, para determinar si un c_i tiene autorización para acceder a una I_j.

El conjunto de reglas de acceso: CR es el elemento central de nuestra propuesta, dicho conjunto contiene una colección de tripletas como la siguiente:

$$(j, h'(c_i), E_{h(c_i)}(k_j)) \tag{1}$$

Donde:

- j es el identificador de la información I_j.
- E es un algoritmo de cifrado con clave simétrica.
- h y h' son dos algoritmos de *hash* diferentes.
- c_i hace referencia al código de encaminamiento de un mensaje i.
- k_j es la clave simétrica necesaria para cifrar y descifrar la información I_j sobre la que se controla el acceso.

El conjunto de informaciones almacenadas: CI es una colección de duplas como la que sigue:

$$(j, E_{k_j}(I_j)) \tag{2}$$

En las que el primer elemento de cada dupla sirve para indexar las diferentes I_j y poder realizar una búsqueda rápida de una información concreta, y el segundo elemento es la I_j en cuestión, debidamente cifrada para que ningún c_i (u otro proceso) sin autorización pueda acceder a ella.

IV-C. Construcción del conjunto de reglas

A la hora de añadir a uno o más nodos una nueva información I_j sobre la que se ha de controlar el acceso, se utiliza el algoritmo 1 para añadir a los conjuntos CI y CR todas las tuplas y tripletas necesarias.

Algoritmo 1 *Almacenamiento de la información*

Entrada: I_j: Información a almacenar.
j: Identificador de la información.
M: Conjunto de mensajes i a con acceso a I_j.
Salida: $\emptyset$

1: Generar una aleatoriamente una clave k_j.
2: Cifrar $E_{k_j}(I_j)$.
3: Añadir a CI la tupla $(j, E_{k_j}(I_j))$.
4: **para cada** $i \in M$ **haz:**
5: Obtener el código de encaminamiento c_i.
6: Calcular $h(c_i)$.
7: Cifrar $E_{h(c_i)}(k_j)$.
8: Calcular $h'(c_i)$.
9: Añadir a CR la tripleta (j, $h'(c_i)$, $E_{h(c_i)}(k_j)$).
10: **final para cada**
11: **devolver:** $\emptyset$

IV-D. Utilizando el conjunto de reglas para controlar el acceso a la información

Cuando un mensaje i llega a un nodo y desde su código de encaminamiento c_i solicita acceso a la información identificada por j, se utiliza el algoritmo 2 para recuperar las claves necesarias para descifrar esta información y, en caso de que el proceso haya concluido satisfactoriamente, permitir a c_i que acceda a I_j.

Algoritmo 2 *Control de acceso*

Entrada: i: Mensaje que solicita el acceso a la información.
j: Identificador de la información a la que se accede.
Salida: I_j: Información identificada por j.

1: Obtener c_i del mensaje i.
2: Calcular $h'(c_i)$.
3: Buscar en CR una tripleta que coincida con (j, $h'(c_i)$, *$clave_cifrada*).
4: Guardar en la variable *$clave_cifrada* el valor correspondiente de dicha tripleta.
5: Calcular $h(c_i)$.
6: Descifrar $D_{h(c_i)}(E_{h(c_i)}(\$clave\_cifrada)) = k_j$.
7: Buscar en CI una dupla que coincida con (j, *$información_cifrada*).
8: Guardar en la variable *$información_cifrada* el valor correspondiente de dicha dupla.
9: Descifrar $D_{k_j}(E_{k_j}(\$información\_cifrada)) = I_j$.
10: **si** I_j ha sido descifrado correctamente **entonces:**
11: **devolver:** I_j
12: **sino**
13: **devolver:** $\emptyset$
14: **final si**

IV-E. Gestionando la modificación de una información

Cuando un mensaje i al que se le ha permitido el acceso a I_j realiza una modificación sobre esta información y decide almacenar la nueva versión I'_j en lugar de I_j, se utiliza el algoritmo 3 para ello.

Algoritmo 3 *Modificación de la información*

Entrada: i: Mensaje que solicita la modificación a la información.
j: Identificador de la información que se modifica.
I'_j: Información modificada.
Salida: verdadero o **falso**.

1: Ejecutar las 9 primeras sentencias del algoritmo 2.
2: **si** I_j ha sido descifrado correctamente **entonces:**
3: Cifrar $E_{k_j}(I'_j)$.
4: Eliminar de CI la tupla $(j, E_{k_j}(I_j))$.
5: Añadir a CI la tupla $(j, E_{k_j}(I'_j))$.
6: **devolver: verdadero**
7: **sino**
8: **devolver: falso**
9: **final si**

Conviene notar que los algoritmos 2 y 3 son esencialmente iguales (las 9 primeras sentencias son las mismas). Para cualquier combinación de valores de entrada, el algoritmo 2 siempre se ejecuta antes que el algoritmo 3. Esto permite que el algoritmo 3, pese a ser el más costoso de los tres que componen nuestro sistema se pueda optimizar y ejecutar más rápido si se mantienen en memoria durante un tiempo los resultados de los cálculos intermedios calculados por el algoritmo 2, evitando así la repetición innecesaria de esas operaciones.

V. Discusión de la seguridad en el control de acceso

En esta sección discutiremos la seguridad ofrecida por nuestro sistema de control de acceso basado en identidad. Para ello analizaremos dos posibles escenarios. En el primero un atacante trata de crear un mensaje pro-activo malicioso i' de forma que su c'_i pueda ganar acceso a una I_j a la que no esté autorizado a acceder. En el segundo, un atacante consigue comprometer el sistema encargado del control de acceso e intenta, teniendo toda la información de los conjuntos CR y CI a su disposición, comprometer una I_j concreta.

V-A. Seguridad ante un ataque remoto

Un atacante remoto que quiera comprometer la información I_j necesitaría forzosamente crear un mensaje pro-activo malicioso i' con un código de encaminamiento malicioso c'_i que se haga pasar por un c_i autorizado ya que, como ya hemos visto en la sección III, al atacante no le sirve de nada que $c'_i = c_i$.

Si el atacante ha sido capaz de interceptar un mensaje i con un c_i autorizado a acceder a I_j, puede utilizar las funciones h y h' conocidas para calcular $h(c_i)$ y $h'(c_i)$. En este punto, lo único que necesita para que el ataque tenga éxito es encontrar un c'_i tal que $h(c'_i) = h(c_i)$ y $h'(c'_i) = h'(c_i)$ con $c'_i \neq c_i$.

El atacante necesita, por lo tanto, realizar un doble ataque de pre-imagen ([14], [15]) a dos algoritmos de *hash* diferentes.

La complejidad de un doble ataque de pre-imagen es de 2^{n+m} si elegimos unos algoritmos h y h' cuya salida sea de

n bits y m bits, respectivamente, y que cumplan las siguientes consideraciones:

- Son seguros ante este ataques de pre-imagen (esto es: la complejidad de un ataque de este tipo es del orden de 2^n donde n es la longitud en bits de la salida de la función).
- Son totalmente independientes entre ellos.

Por lo tanto, si elegimos dos funciones h y h' independientes cuyas longitudes n y m sean consideradas seguras y contra las que no existan algoritmos conocidos que reduzcan la complejidad del ataque de preimagen, podemos concluir que el sistema es seguro ante este tipo de ataques.

V-B. *Seguridad ante un ataque local*

En el peor escenario posible un atacante ha comprometido la infraestructura de un nodo y quiere comprometer una o más informaciones I_j concretas. Se pueden presentar dos situaciones diferentes en función de si a ese nodo llegan o no, mensajes con c_i autorizado. Cabe destacar que ésta es una situación muy improbable ante la que la seguridad de cualquier sistema de control de acceso resultaría comprometida.

Ataque en solitario: Denominamos así al intento realizado por un atacante por comprometer la información I_j utilizando únicamente la información contenida en los conjuntos CR y CI, a los que previamente ha conseguido ganar acceso.

El atacante debe conocer la j mediante la que se identifica I_j, por lo que puede ignorar el conjunto CR y centrarse únicamente en CI, que, como se ha visto en la sección IV contiene duplas con los valores $(j, E_{k_j}(I_j))$. Una vez el atacante localice una dupla cuyo valor del primer campo sea j, la complejidad del ataque se reduce a romper el cifrado proporcionado por el algoritmo E.

Si el algoritmo E elegido es seguro, la complejidad de dicho ataque es de 2^n donde n es el tamaño en número de bits de la clave k_j utilizada, por lo que podemos concluir que el sistema es seguro ante este tipo de ataques.

Ataque en presencia de mensajes autorizados: Por otro lado, si el atacante ha conseguido acceder a los conjuntos CR y CI e intercepta un mensaje i con código de encaminamiento c_i autorizado a acceder a I_j. Entonces sí que puede comprometer dicha información, ya que puede usar c_i para calcular $h(c_i)$ y $h'(c_i)$ y utilizarlos, siguiendo el algoritmo explicado en la sección IV, para realizar un acceso no autorizado a I_j.

VI. Aplicación: mensajes pro-activos en MobileC

Para comprobar la viabilidad de nuestra propuesta y evaluar su funcionamiento en el marco de una aplicación concreta, hemos modificado la plataforma de agentes móviles MobileC [2] con el objetivo de utilizarla como elemento central de una DTN basada en mensajes pro-activos. Estas modificaciones se sustentan en el uso de las librerías OpenSSL [3], Libxml2 [5] y Raptor [4]. Entre las más importantes se cuentan:

- La adición de la posibilidad de definir el código de encaminamiento de los mensajes.
- La adición de un módulo que realiza el descubrimiento de vecinos. Este módulo es necesario para utilizar esta plataforma en entornos DTN.
- La inclusión del sistema de control de acceso presentado en este artículo, implementado a través de los algoritmos especificados en la Sección IV.

Respecto a los algoritmos de *hash* y cifrado necesarios, hemos escogido los siguientes: h = SHA2-256, h' = SHA1, E = AES-256 CBC. Para realizar esta elección hemos tenido en cuenta que h y h' sean algoritmos independientes entre ellos y que el tamaño de la clave de E y de la salida de h y h' permita trabajar con ellas sin tener que truncar o extender estos valores.

Asimismo, se ha implementado una serie de mensajes pro-activos que acceden a las informaciones de nuestra colección. La colección de informaciones utilizada consiste en un conjunto de sentencias RDF [16] en las que cada información puede estar guardada en claro, y por lo tanto ser una información de acceso público, o puede estar protegida, esto es, cifrada y marcada con una etiqueta que indica que hay que controlar el acceso a dicha información. A continuación se muestra un ejemplo de una información de cada tipo.

```
<rdf:Description rdf:about="http://prueba.com/publ">
  <campo:frase>Dato numero 1</campo:frase>
  <campo:cifrado>0</campo:cifrado>
</rdf:Description>
<rdf:Description rdf:about="http://prueba.com/prot">
  <campo:frase>h+Iy3+RwDfn/qcPEF2Y5oA=</campo:frase>
  <campo:cifrado>1</campo:cifrado>
</rdf:Description>
```

El entorno de prueba ha estado constituido por una máquina con procesador Intel Core2 Quad Q8400 de 64 bits con 6GB de RAM y sistema operativo GNU/Linux con kernel 3.3.2.

Con la finalidad de obtener resultados concluyentes hemos medido el tiempo utilizado para el encaminamiento de 1400 mensajes pro-activos. La mitad de estos mensajes ha realizado su encaminamiento accediendo a informaciones públicas y la otra mitad a informaciones protegidas. El tamaño de estas informaciones es de 2KB, 100KB, 200KB, 500KB, 1MB, 2MB, 3MB, 4MB, 5MB, 6MB, 7MB, 8MB, 9MB y 10MB. Las pruebas realizadas nos han permitido obtener los valores promedios de tiempo de encaminamiento, así como sus correspondientes desviaciones estándar.

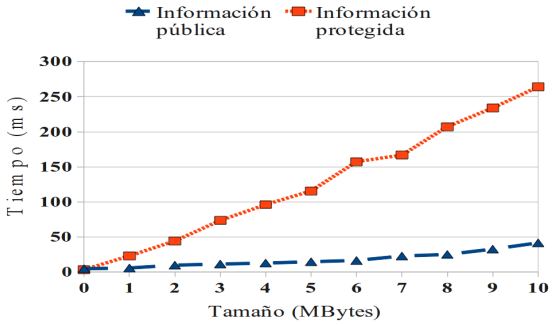

Figura 2. Tiempo de encaminamiento con acceso a información

Para obtener el gráfico mostrado en la Figura 2 hemos agrupado las muestras en función del tipo y el tamaño de la información accedida. En el gráfico podemos observar como la

penalización que añade nuestro sistema de control de acceso crece linealmente con respecto al tamaño de la información protegida. En promedio, el tiempo necesario para el encaminamiento cuando se accede a una información protegida es 6,66 veces mayor del que se necesita si el acceso se realiza a una información pública del mismo tamaño.

Información	Media aritmética (μs)	Desv. estándar (μs)
Pública (2KB)	3.435,31	136,14
Protegida (2KB)	4.809,11	242,59
Pública (10MB)	32.711,84	2.204,36
Protegida (10MB)	263.975,88	46.824,52

Tabla I
TIEMPO UTILIZADO EN EL ENCAMINAMIENTO

En la Tabla I se muestran los resultados obtenidos al acceder a informaciones de 2KB y 10MB. En la aplicación en la que hemos implementado el control de acceso, PROSES [13], las informaciones utilizadas durante el encaminamiento tienen un tamaño máximo de 2KB, el tamaño medio de la cola de mensajes es 10 y las ventanas de conectividad son de 10-30s, por lo que el tiempo necesario para el encaminamiento aumenta de unos 32-36ms a unos 46-50ms cuando se aplica el control de acceso. En el caso extremo (en nuestra aplicación) de utilizar informaciones de 10MB, el tiempo necesario para el encaminamiento se mantiene por debajo de los 3s. En ambos casos, la penalización introducida es totalmente asumible.

VII. CONCLUSIONES Y TRABAJO FUTURO

La mayoría de algoritmos de encaminamiento para redes DTN no tienen en cuenta la estrecha relación entre las aplicaciones y el modo en que estas utilizan la red. Las redes DTN basadas en mensajes pro-activos solucionan este problema permitiendo a las aplicaciones definir el tipo de informaciones que les son útiles a la hora de encaminar y dotando a los mensajes de un código de encaminamiento que utiliza esta información orientada a aplicación para decidir la mejor ruta hacia su destino.

En este artículo se ha presentado un sistema de control de acceso a información de encaminamiento para redes DTN basadas en mensajes pro-activos. La aportación fundamental de esta propuesta radica en el uso de la propia identidad del código que trata de acceder a una información para decidir si está autorizado a hacerlo. Este sistema utiliza dos funciones de *hash* diferentes, una para identificar al código de encaminamiento que trata de acceder a la información y la otra para, de forma conjunta con un algoritmo de cifrado con clave simétrica, proteger esta información y garantizar su confidencialidad e integridad.

Esta propuesta ayudará a mejorar la seguridad de las redes DTN basadas en mensajes pro-activos, evitando que un atacante pueda generar mensajes maliciosos desde cuyo código de encaminamiento se realicen accesos o modificaciones no autorizadas a los datos guardados en los diferentes nodos. El sistema es seguro ante cualquier ataque remoto, e incluso ante uno de los peores escenarios posibles: una situación en la que un atacante haya logrado comprometer la infraestructura de un nodo de la red, ya que el atacante necesitará también interceptar un mensaje autorizado a acceder a la información atacada si quiere comprometerla.

La aplicación de nuestra propuesta en una contexto determinado, PROSES, nos ha permitido evaluar su viabilidad y su rendimiento, que ha demostrado ser más que suficiente incluso utilizando informaciones con un tamaño 5.000 veces mayor a las que se suelen utilizar en este contexto.

Como trabajo futuro, quedaría realizar una generalización del sistema para utilizarlo fuera del ámbito de las redes DTN. Además, siguiendo los mismos principios con los que se controla el acceso a información por parte del código móvil de los mensajes pro-activos, se podría controlar el acceso de los agentes móviles a diferentes recursos locales.

REFERENCIAS

[1] Delay Tolerant Networking Research Group. http://www.dtnrg.org.
[2] MobileC. http://www.mobilec.org/.
[3] OpenSSL Project. http://www.openssl.org/.
[4] Raptor RDF Syntax Library. http://librdf.org/raptor/.
[5] The XML C parser and toolkit of Gnome. http://xmlsoft.org/.
[6] K. Aniket, Z. Gregory M., and H. Urs. Anonymity and Security in Delay Tolerant Networks. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 504–513, 2007.
[7] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo. Towards Securing Disruption-Tolerant Networking. *Nokia Research Center, Tech. Rep. NRC-TR-2007-007*, 2007.
[8] C. Borrego and S. Robles. Seguridad en la planificaciónn de agentes móbiles en redes dtn. In *Actas de la XI Reunión Española de Criptología y Seguridad de la Información*, Tarragona, September 2010.
[9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, 2006.
[10] S. Castillo, S. Robles, M. de Toro, and J. Borrell. Seguridad en protocolos de encaminamiento para redes dtn. In *Actas de la XI Reunión Española de Criptología y Seguridad de la Información*, Tarragona, September 2010.
[11] C. Diot and *et al.* Haggle project. http://www.haggleproject.org/.
[12] S. Farrell and V. Cahill. *Delay- and Disruption-Tolerant Networking*. Artech House, Inc., Norwood, MA, USA, 2006.
[13] N. Giuditta, S. Robles, A. Viguria, S. Castillo, M. Cordero, and L. Fernández. Proses - network communications for the future european atm system. In *In Proceedings of the International Conference on Application and Theory of Automation in Command and Control Systems*, May.
[14] P. Hoffman and B. Schneier. Attacks on Cryptographic Hashes in Internet Protocols. RFC 4270, November 2005.
[15] J. Kelsey and B. Schneier. Second preimages on n-bit hash functions for much less than 2^n work. Cryptology ePrint Archive, Report 2004/304, 2004. http://eprint.iacr.org/.
[16] G. Klyne and J.J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, February 2004.
[17] A. Lindgren, A. Doria, and O. Schelén. Probabilistic Routing in Intermittently Connected Networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.
[18] M. Ramadas, S. Burleigh, and S. Farrell. Licklider Transmission Protocol - Specification. RFC 5326 (Experimental), September 2008.
[19] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.
[20] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and Wait: an Efficient Routing Scheme for Intermittently Connected Mobile Networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, page 259. ACM, 2005.
[21] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected ad hoc Networks. Technical report, Duke University, 2000.