Protocolo de No-repudio para Redes DTN Basado en Intercambio Justo de Firmas

Sergi Martínez-Bea, Sergio Castillo-Pérez, Sergi Robles and Marcel Gozalbo-Baró
Department of Information and Communications Engineering (dEIC),
Universitat Autònoma de Barcelona,
08193 Bellaterra, Spain
Email: {smartinez, scastillo, srobles}@deic.uab.es

***Resumen*—En este artículo se presenta un protocolo de no-repudio para el intercambio de agentes móviles en redes DTN. El protocolo se basa en el uso combinado de un esquema de intercambio justo de firmas, de la criptografía basada en la identidad, y del propio agente a transmitir como elemento clave de revelación tardía en el intercambio de firmas. El esquema presentado permite determinar, mediante el rastreo de recibos, el nodo que no reenvía agentes pese haber aceptado su custodia y compromiso de reenvío. El artículo incluye detalles sobre la implementación realizada, los resultados empíricos obtenidos, y una discusión sobre su utilización en entornos DTN de baja conectividad. Los resultados pueden ser fácilmente adaptables a otros escenarios similares.**

***Index Terms*—Redes oportunísticas, DTN, Seguridad, Protocolos de No-repudio.**

I. Introducción

En las redes DTN (*Delay and Disruption Tolerant Networking*) [1], el encaminamiento es una función básica. En estas redes, los nodos se ponen en contacto solo de forma esporádica, aprovechando estas ocasiones para intercambiar mensajes entre ellos. El encaminamiento, por tanto, no puede seguir las mismas estrategias basadas en la dirección de destino y en tablas de encaminamiento locales, que se usan en otras redes como Internet. Por otro lado, este encaminamiento es un elemento crítico para que los mensajes lleguen a su destino, y que lo hagan de la mejor manera. En [2] se presenta un innovador enfoque para realizar un encaminamiento basado en el contexto y proporcionado por la propia aplicación. En este esquema, cada mensaje toma un papel activo en su propio encaminamiento, pudiendo existir en una misma red y al mismo tiempo diversos mensajes siendo encaminados bajo políticas totalmente diferentes e independientes entre ellas, proporcionadas por las aplicaciones a las que pertenecen y sin realizar ningún despliegue previo. El sustrato de esta nueva manera de afrontar los retos de los escenarios DTN es el código móvil, y en particular los agentes móviles software.

La protección de este encaminamiento es un punto especialmente sensible en el campo de la seguridad en redes DTN. Por un lado, tenemos que la privacidad de los mensajes debe ser protegida de observadores externos ajenos a los elementos comunicantes; y por otro lado debe tenerse la certeza que el nodo que acepta la custodia y retransmisión de un mensaje pertenece a la red y no pretende secuestrarlo. En los escenarios de conectividad muy limitada, el acceso solo a los vecinos inmediatos presenta serios problemas para las soluciones de seguridad que requieren de un acceso a una cierta infraestructura de seguridad, como por ejemplo a un servicio de comprobación de certificados revocados.

En este artículo se presenta un protocolo de no-repudio para el sistema de migración de agentes móviles entre nodos de una DTN. El protocolo de no-repudio está basado en un esquema de intercambio justo de firmas en el que se introduce el propio objeto de la migración, el agente móvil, como elemento criptográfico de intercambio, que no es revelado hasta el final del proceso. En el protocolo se introducen también algunos elementos que utilizan criptografía basada en la identidad (IBC), que en los entornos DTN de baja conectividad ofrecen una solución aceptable como alternativa a la infraestructura de seguridad siempre accesible [3]. La utilización conjunta de estos tres elementos –migración de agente, intercambio justo de firmas, e IBC– no está totalmente acoplada, de modo que es posible reemplazar alguna de estas partes en caso necesario sin afectar al diseño general, siempre claro está que se conserven sus funcionalidades básicas.

El protocolo presentado aquí permite obtener una serie de recibos que pueden presentarse a terceros para demostrar las transmisiones de agentes entre nodos de la red DTN, de manera que en caso de desaparición de un agente móvil sea posible el seguimiento del rastro de recibos hasta dar con el último nodo responsable de su custodia. Este mecanismo de seguridad, por tanto, es útil para disuadir a los nodos que rehúsen el reenvío de mensajes una vez aceptados y promover su comportamiento honesto.

Para analizar la viabilidad técnica de la propuesta y determinar la penalización temporal que debe pagarse por su uso respecto al envío repudiable, hemos realizado una implementación en Mobile-C usando diversas bibliotecas criptográficas en un entorno de aplicación de DTN basada en agentes.

La sección II presenta un estado del arte sobre el tema, y la sección III describe en detalle la propuesta. En la sección IV se realiza una discusión del protocolo y de su aplicabilidad en los entornos de DTN de baja conectividad basados en agentes. La implementación, estudio de viabilidad técnica y resultados se encuentran en la sección V. Y finalmente, la sección VI presenta las conclusiones finales del artículo.

II. Estado del arte

El no-repudio es estrictamente necesario en cualquier red que se necesite tener comprobantes no falseables de envios y recepciones de datos. La mayoría de propuestas hasta ahora, se basan en una tercera parte de confianza (TTP, *Trusted Third Party*) que actúa como intermediario o moderador de las comunicaciones. En [4] presentan una recopilación de protocolos de no-repudio en el que se encuentran propuestas tanto con TTP como sin TTP. En el entorno DTN, por su naturaleza, las propuestas con TTP no son viables y, por lo tanto, conviene centrarse en los protocolos sin TTP. En [5] se presenta una propuesta sin TTP, que se basa en partir el mensaje en n partes y enviarlas una a una, recibiendo confirmación del receptor por cada una esas partes enviadas. En caso que el receptor no confirme alguna de las partes, el emisor aborta la transmisión y el receptor no tiene el mensaje completo. Este protocolo no es $100\,\%$ fiable ya que existe la probabilidad $1/n$ de que el receptor pueda adivinar cual es la última transmisión y lograr así el repudio. El enfoque probabilístico que proponen tiene la limitación que en el entorno DTN, la ventana de tiempo disponible para la transmisión es variable y podría ser tan pequeña que no diera tiempo a completar los n pasos.

En criptografía existe el problema del intercambio justo de firmas, dónde las firmas de emisor y receptor quedan vinculadas al mismo tiempo. De esta forma ni emisor ni receptor se pueden retractar de la comunicación, proporcionando así no-repudio. En [6] se introduce el concepto de las firmas concurrentes, en que emisor y receptor pueden producir firmas ambiguas que no son vinculantes hasta que uno de los dos libera una pieza extra de información. En casos concretos, la ambigüedad no se cumple y las firmas podrían ser perfectamente vinculadas antes de la liberación. En [7] solucionan este problema presentando las firmas concurrentes perfectas, que proporcionan una ambigüedad completa de las firmas antes de ser vinculadas.

En el entorno DTN con agentes móviles, la gestión de las claves requeridas en la firma concurrente perfecta no es viable por la naturaleza de las redes, puesto que no permite disponer de una PKI. Una posible solución es usar criptografía basada en la identidad (IBC, *Identity Based Cryptography*) ya que permite que las claves públicas sean un rasgo identificativo y, por tanto, conocido por todos. En esta línea, en [8] proponen una forma de construir genéricamente firmas concurrentes perfectas basadas en la identidad. El principal problema de estas propuestas es su complejidad, por lo que no es viable su aplicación directa en las redes. Por este motivo, en [9] presentan un esquema eficiente para el intercambio justo de firmas, que sigue manteniendo las propiedades interesantes de la ambigüedad y la no vinculación hasta que se libera una pieza concreta de información.

Dentro del area de las redes DTN, también se han hecho propuestas para dotarlas de seguridad. En [2] se hace un recorrido por los distintos protocolos de encaminamiento, des de los más clásicos a las tendencias más actuales, como lo es el encaminamiento basado en agentes móviles, todo desde la perspectiva de la seguridad. Hasta el momento no se ha presentado ninguna propuesta para añadir no-repudio a las comunicaciones en redes DTN basadas en agentes móviles en las que, por su naturaleza, no se pueden aplicar los métodos clásicos.

III. Propuesta

En este artículo proponemos un protocolo de no-repudio simétrico basado en el esquema eficiente de intercambio justo de firmas propuesto en [9], combinándolo con criptografía basada en la identidad para solventar el problema de la gestión de claves. Nuestra propuesta está orientada a entornos DTN basados en agentes móviles. A continuación se presentan unas definiciones previas y la especificación del protocolo en cuestión.

Notación	Descripción
sk_i	Clave privada del usuario i
pk_i	Clave pública del usuario i
$H(m)$	Función hash sobre el mensaje m
$Salt$	Número aleatorio
ipk	Clave pública IBC del receptor (Identidad)
isk	Clave privada IBC del emisor
$E_{ipk}(m)$	Cifrado IBC de m con la clave isk
$D_{isk}(m')$	Descifrado IBC de m' con la clave isk
$S_{isk}(m)$	Firma IBC de m con la clave ipk
$V_{ipk}(m,s)$	Verificación de la firma IBC s asociada al mensaje m con la clave ipk

Cuadro I
Notación utilizada en el protocolo propuesto

III-A. Definiciones previas

Para comprender bien el funcionamiento del protocolo que proponemos, primero se deben definir los algoritmos que se usarán durante el funcionamiento del mismo. Dichos algoritmos son *SystemSetup*, *FSign*, *SVerify* y *KVerify*, y se encuentran definidos a continuación. Para describir la propuesta basada en dichos algoritmos, y con la finalidad de facilitar la comprensión al lector, en la tabla I se ha recogido la notación básica que se empleará a lo largo del artículo.

Algorithm 1 *SystemSetup*

Input: $\emptyset$
Output: $\emptyset$

1: Escoger dos primos grandes p y q tales que $q \mid p-1$
2: Escoger un g de orden q tal que $g \in \mathbb{Z}_p^*$
3: **for** i *in* $\langle Alice, Bob \rangle$ **do**
4: Generar el par de claves (sk_i, pk_i) tal que $pk_i = g^{-sk_i} \ mod \ p$, donde sk_i es la clave privada y pk_i la clave pública.
5: **end for**
6: **return**

Algorithm 2 *FSign*

Input: pk: Clave pública del emisor.
isk: Clave IBC privada del emisor.
k: $H(Agent \mid\mid Salt)$.
Output: La tripleta $\langle m, k, s\rangle$, que corresponde a la firma σ

1: Escoger un w tal que $w \in \mathbb{Z}_p^*$
2: Calcular $m = \langle E_{ipk}(pk), S_{isk}(H(pk))\rangle$
3: Calcular $r = g^w \ mod \ p$
4: Calcular $e = H(m, k, r)$ donde H es una función hash.
5: Calcular $c = w + esk \ mod \ q$, donde sk es la clave privada de quien firma.
6: **return** $\sigma = \langle m, k, s\rangle$ donde $s = \langle r, e, c\rangle$

Algorithm 3 *SVerify*

Input: σ: Firma a verificar, donde $\sigma = \langle m, k, s\rangle$, $s = \langle r, e, c\rangle$ y $m = \langle E_{ipk}(pk), S_{isk}(H(pk))\rangle$
Output: True o False

1: Extraer $pk = D_{isk}(E_{ipk}(pk))$
2: Calcular $r_v = g^c pk^e \ mod \ p$
3: **if** $e == H(m, k, r_v)$ **AND** $V_{ipk}(H(pk), S_{isk}(H(pk))) == True$ **then**
4: **return** True
5: **else**
6: **return** False
7: **end if**

Algorithm 4 *KVerify*

Input: σ: Firma recibida donde $\sigma = \langle m, k, s\rangle$
ks: $\langle Agent, Salt\rangle$.
Output: True or False

1: **if** $SVerify(\sigma) == True$ **AND** $k == H(keystone)$ **then**
2: **return** True
3: **else**
4: **return** False
5: **end if**

III-B. Especificación del protocolo

Una vez ya se han definido los algoritmos que se usan en el protocolo, se puede proceder a la especificación del mismo. Imaginemos un nodo emisor y un nodo receptor Alice y Bob, respectivamente, en el que Alice quiere quiere realizar una migración de un agente hacia Bob. En la Figura 1 se puede apreciar el esquema del protocolo, y que detallamos a continuación:

Fase 1: Antes de establecer las primeras comunicaciones entre nodos, hace falta generar los parámetros del protocolo. Para ello, se deben generar los valores p, q y g que deben ser conocidos por todos los nodos del sistema y cada nodo creará su par de claves $\langle sk, pk\rangle$, tal y como queda especificado en el Algoritmo 1.

Fase 2: Una vez están generados los parámetros del sistema y cada nodo dispone ya de su par de claves ya se puede iniciar transmisiones. Para ello, Alice, el nodo emisor, genera su firma $\sigma_A = \langle m_A, k, s_A\rangle$ tal y como se muestra en el Algoritmo 2, y se lo manda a Bob, el nodo receptor. De momento, la firma que Bob recibirá no tiene ningún tipo de validez delante de una tercera parte, ya que no vinculará a Alice en ningún momento.

Fase 3: Bob recibe la firma de Alice. Aunque la firma que ha recibido no se vincule a Alice, Bob ha de verificar que se trata de una firma válida. Para ello, seguirá los pasos descritos en el Algoritmo 3. En caso que la firma sea válida el algoritmo devolverá *True* y Bob podrá proceder a la Fase 4 del protocolo. En caso contrario, el algoritmo habría devuelto *False* y, por lo tanto, se abortaría la transmisión.

Fase 4: Bob ha recibido una firma válida de Alice y debe proceder a responder. Para ello, Bob genera su firma $\sigma_B = \langle m_b, k, s_B\rangle$, tal y como se indica en el Algoritmo 2, y se la envia a Alice.

Fase 5: Cuando Alice recibe la firma de Bob, procede también a verificar que se trate de una firma válida. Para ello, se debe seguir los pasos indicados en el Algoritmo 3. En caso que el valor de retorno sea *True* se procederá a la Fase 6, donde Alice, el emisor, libera la pieza clave de información que hará que las firmas σ_A y σ_B se puedan verificar y vincular a sus respectivos dueños. En caso que el valor de retorno del Algoritmo 3 sea *False*, la firma no sería válida y, por lo tanto, se abortaría la transmisión.

Fase 6: Una vez Alice y Bob han intercambiado sus firmas, hace falta la liberación de la pieza clave que vinculará cada firma a su dueño. Ésta pieza clave, se compone del Agente en si, y de un *Salt* (número aleatorio) que será distinto para cada transmisión. De esta forma, la pieza clave es única para cada migración y, por lo tanto, no se podrá falsear. En caso que la pieza clave no se liberara, ni Alice ni Bob podrían demostrar delante de una tercera entidad que Alice ha enviado el agente o que Bob ha recibido el agente, quedando así como si no se hubiera establecido ninguna comunicación. Si la liberación se ha llevado a cabo con éxito, se procede a la Fase 7, la fase de verificación y vinculación de las firmas.

Fase 7: La última fase del protocolo es la que verifica y vincula las firmas tanto del emisor como del receptor, proporcionando así el no repudio simétrico. En este paso, se utiliza el Algoritmo 4 que comprueba si la pieza clave es válida y si la verificación de la firma es correcta y, por lo tanto, la vincula inequívocamente a su creador. Este paso lo puede realizar cualquier nodo que disponga de las firmas y de la pieza clave.

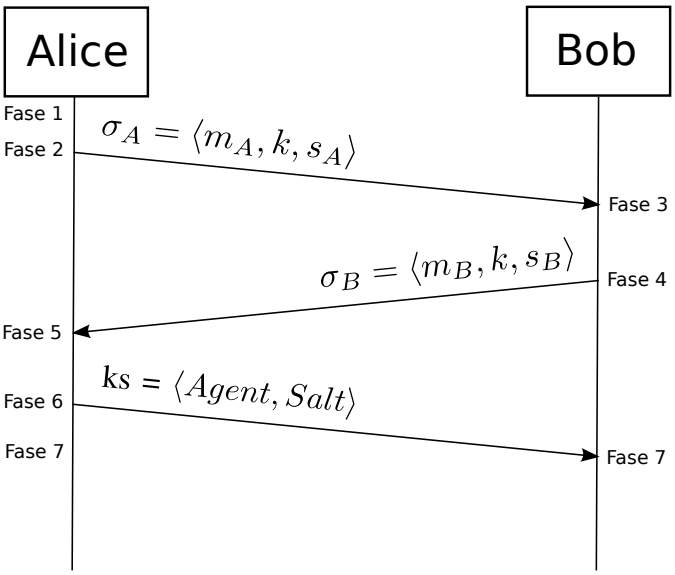


Figura 1. Esquema del protocolo de no repudio.

III-C. Escenario de apliación

Tal y como se ha mencionado anteriormente, nuestra propuesta está orientada a un entorno específico, las redes DTN basadas en agentes móviles, donde la conectividad y la capacidad de los nodos es limitada. En el apartado anterior hemos definido cómo se deben comunicar dos nodos cuando se quieren intercambiar mensajes, pero no hemos especificado cómo se comportan estos nodos dentro de la red. Cada nodo, cuando realiza intercambio de mensajes, va guardando los recibos de todas las comunicaciones con no-repudio que ha realizado, para poder justificarse en caso de ser necesario. Resulta evidente pensar que los nodos, aún sin saber su capacidad, podrían encontrarse sin recursos de almacenamiento y no poder guardar más recibos, lo que supondría un problema.

El hecho de utilizar criptografía basada en identidad, conlleva suponer que se dispone de un elemento que genera las claves privadas para los nodos, llamado PKG (*Private Key Generator*), al que los nodos deberán contactar periódicamente para renovar sus claves. Aprovechando este punto de encuentro, los nodos descargarían todos sus recibos en el PKG, el cual los administraría y cancelaría. Dada una serie de recibos que provienen de comunicaciones relacionadas entre sí, el PKG podrá cancelar esos recibos de migraciones de agentes que han quedado debidamente justificadas por parte de emisor y receptor. Por ejemplo, un nodo A inicia la migración de un agente a un nodo B. Se ejecuta el protocolo de no-repudio que se ha descrito en el apartado anterior y, por lo tanto, A puede demostrar que le ha enviado el agente a B, y B puede demostrar que A se lo ha enviado. En el momento en que B realice una migración del agente hacia un nodo C, y esta migración concluya con éxito, los recibos de la comunicación entre A y B ya no son relevantes puesto que la responsabilidad de un posible incidente, recaerá en la comunicación entre B y C únicamente. De esta forma, ya no hay peligro de que los nodos se queden sin espacio para almacenar los recibos y, además, se guardaría solamente la información relevante.

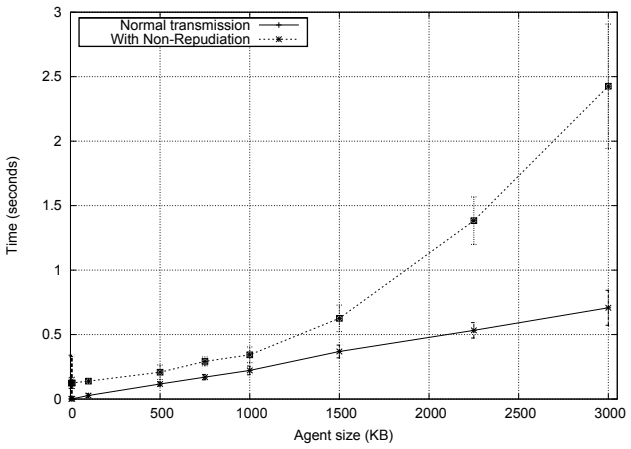
IV. Discusión

En esta sección analizaremos nuestra propuesta des del punto de vista de la seguridad. Para cualquier protocolo de no-repudio es necesario comprobar su corrección y su resistencia a la falsificación. Dado que nuestra propuesta se basa en el esquema de intercambio justo de firmas que se presenta en [9], las demostraciones que ofrecen son totalmente válidas y aplicables a nuestra propuesta.

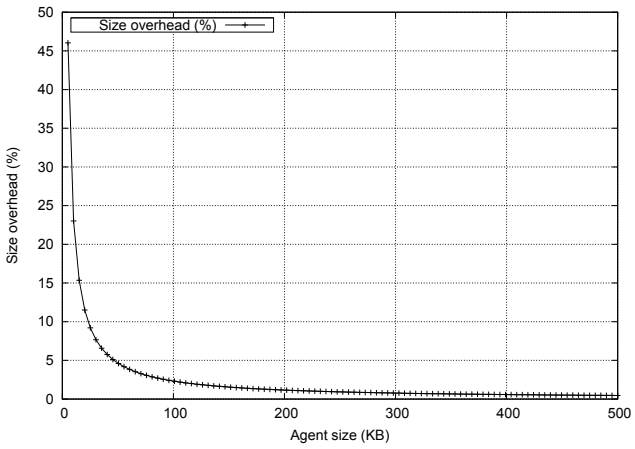
Cómo hemos visto en la sección anterior, donde se especifica el protocolo, éste consta de siete fases. Las fases de verificación que hay entre transmisiones nos aseguran que los mensajes que llegan son totalmente válidos. Cuando Bob recibe la firma de Alice (Figura 1, Fase 3) la verificación que hace es para asegurarse que lo que le ha llegado es realmente una firma correspondiente al protocolo y, además, asegurarse de que realmente es Alice quién lo manda y no un tercer nodo malicioso. La propuesta en la que se basa nuestro protocolo [9], ofrece las herramientas para realizar el primer tipo de verificación pero no para validar la autenticidad del mensaje. Aquí es donde entra en juego el uso de la criptografía basada en identidad. Dado que Bob va a necesitar la clave pública de Alice (la que se genera en la Fase 1), ésta se la manda cifrada y firmada con IBC dentro de la firma σ_A. Así pues, Bob solamente ha de verificar la firma IBC de Alice para saber que realmente proviene de ella. Alice realiza el mismo proceso cuando recibe la firma σ_B de Bob. Ambas firmas no se pueden vincular ni a Alice ni a Bob de forma inequívoca, hasta que la pieza clave ks no es liberada. Además, en nuestro esquema, la pieza clave ks contiene el agente a ser migrado, con lo que si no se completa todo el protocolo el agente no llega a realizar la migración. Cabe destacar que las firmas σ_A y σ_B, están vinculadas con la pieza clave, puesto que llevan dentro el hash del agente concatenado con la *Salt*. De esta manera, se utiliza una característica distintiva del agente a migrar en el propio proceso de su transmisión, es decir, se utiliza el propio agente como un elemento criptográfico del protocolo en vez de un dato descartable. Una vez liberada la pieza clave, cualquier nodo puede verificar y validar las firmas, por lo tanto es cuando se hace efectivo el no-repudio. Cabe destacar que en nuestra propuesta IBC no está encastada en el propio protocolo, al contrario de lo que ocurre en otras aproximaciones, como por ejemplo en [8]. Este hecho nos proporciona una mayor flexibilidad, puesto que permitiría remplazar IBC por otra alternativa en caso necesario.

V. Implementación y evaluación

Con el objetivo de evaluar la penalización que podría introducir nuestra propuesta con respecto a una transmisión sin no-repudio, se ha implementado una librería escrita en lenguaje C y denominada `DTN-NonRep`. Dicha librería, especialmente diseñada para su utilización en contextos DTN, incluye los algoritmos necesarios de acuerdo a la especificación del protocolo recogido en la Sección III. Su implementación se



(a) Time to Send vs. Agent Size



(b) Size overhead vs. Agent Size

Figura 2. Resultados experimentales

sustenta en el uso de las librerías OpenSSL v.1.0.0h [10] y la PBC Library v.0.5.12 [11] de la Universidad de Stanford. Respecto a los esquemas de criptografía basados en identidad para el cifrado y la firma digital, se han empleado Boneh-Franklin [12], e ISO/IEC 14888-3 IBS-1 basado en el esquema de Hess [13], respectivamente.

Dicha librería se ha integrado dentro de la plataforma Mobile-C [14]. Mobile-C es catalogada como una plataforma multi-agente para el soporte de agentes móviles escritos en C/C++, especialmente diseñada para entornos de redes inteligentes basadas en ingeniería mecatrónica y sistemas embebidos. Asimismo, dicho entorno sigue el estándar IEEE FIPA, lo que garantiza la interoperabilidad con otras plataformas que sigan el mismo estándar. Sus características lo hacen de un sistema ideal para la utilización de agentes móviles en el ámbito de las redes DTN.

En esta integración se han estimado los tiempos necesarios en el envío de diversas muestras de agentes móviles con el protocolo de no repudio[1]. El entorno de prueba ha estado constituido por dos máquinas cuyas características han sido las siguientes. Un sistema denominado `alice`, basado en un procesador Intel Core2 Quad Q8400 de 64 bits con 6GB de RAM a 2.66GHz, y sistema operativo GNU/Linux con kernel 3.3.2 y librería glibc 2.13. Un sistema denominado `bob`, basado en un procesador Intel Core2 Duo E8400 a 3.00GHz de 32 bits con 1GB de RAM y sistema operativo GNU/Linux con kernel 2.6.32 y librería glibc 2.11. En ambas plataformas se ha empleado el compilador gcc en su versión 4.5.3 y con los *CFLAGS* `-O3 -march=native` con el objetivo de optimizar el código máquina resultante.

Las muestras utilizadas como banco de pruebas se ha basado en nueve agentes móviles de tamaños de 5KB, 10KB, 100KB, 500KB, 750KB, 1MB, 1.5MB, 2.25MB y 3MB. Con la finalidad de obtener resultados concluyentes, las aplicaciones han realizado el envío de cada muestra 100 veces, midiendo los tiempos tanto con la incorporación del protocolo de no repudio, como sin éste. Las pruebas realizadas nos han permitido obtener los valores promedios del tiempo de transmisión así como sus correspondientes desviaciones estándar. Dichos resultados han sido recogidos en la gráfica 2(a). De forma análoga, el *overhead* introducido por el protocolo de no repudio ha sido representado de forma porcentual en la gráfica 2(b).

Como podemos observar en las gráficas correspondientes, el porcentaje del *overhead* respecto al tamaño del agente va tomando menos importancia conforme se incrementa el tamaño del mismo, mientras que el tiempo de transmisión es cada vez mayor conforme se aumenta el tamaño del agente. A pesar de los altos tiempos de transmisión para tamaños grandes, el coste resulta asumible teniendo en cuenta los tamaños que toman los agentes en entornos de aplicación, como es el caso de [15], que oscilan entre 1KB y 2KB.

VI. Conclusiones y trabajo futuro

En este artículo hemos presentado una propuesta de protocolo de no-repudio para la migración de agentes en redes DTN que utilizan código móvil para el encaminamiento dinámico. Nuestro esquema se basa en una combinación de intercambio justo de firmas y criptografía basada en la identidad (IBC), que resuelve el problema de gestión de claves en este tipo de redes.

Además, se ha pensado la aplicabilidad del protocolo en el contexto de este tipo de redes, teniendo en consideración los detalles relativos a la descarga de los recibos en los PKGs, y la cancelación de los recibos innecesarios.

La implementación realizada y los experimentos ejecutados, han demostrado que la utilización del protocolo dentro del ámbito de las redes DTN, a pesar de las limitaciones de recursos que tienen los nodos, es viable debido a que la penalización añadida es asumible.

Como trabajo futuro, quedaría realizar una generalización del protocolo para poder ser usado fuera del ámbito de las

[1]Nótese aquí que el tiempo necesario en el envío del las claves privadas por parte del PKG ha sido despreciado, considerando exclusivamente el tiempo derivado de la transmisión de los agentes móviles

redes DTN. Además, de la misma forma que los agentes móviles tienen su código de encaminamiento, se podría dotar a los mismos de código de no-repudio. De esta forma, la implementación del no-repudio sería delegada al programador de las aplicaciones que correrían en la red DTN, y no restringiría al programador a la infraestructura.

AGRADECIMIENTOS

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

REFERENCIAS

[1] S. Farrell and V. Cahill, *Delay- and Disruption-Tolerant Networking*. Norwood, MA, USA: Artech House, Inc., 2006.

[2] S. Castillo-Pérez, S. Robles, M. C. de Toro, and J. Borrell, "Seguridad en protocolos de encaminamiento para redes DTN," in *XI Reunión Española sobre Criptología y Seguridad de la Información (RECSI), Tarragona, Spain*, Sep. 2010.

[3] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo, "Applicability of Identity-Based Cryptography for Disruption-Tolerant Networking," in *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*. New York, NY, USA: ACM, 2007, pp. 52–56.

[4] S. Kremer, O. Markowitch, and J. Zhou, "An Intensive Survey of Fair Non-Repudiation Protocols," *Computer Communications*, vol. 25, no. 17, pp. 1606 – 1621, 2002.

[5] O. Markowitch and Y. Roggeman, "Probabilistic Non-Repudiation without Trusted Third Party," in *2nd Conference on Security in Communication Networks*, Amalfi, Italy, Sep. 1999.

[6] L. Chen, C. Kudla, and K. Paterson, "Concurrent Signatures," in *Advances in Cryptology - EUROCRYPT 2004*, ser. Lecture Notes in Computer Science, C. Cachin and J. Camenisch, Eds. Springer Berlin / Heidelberg, 2004, vol. 3027, pp. 287–305.

[7] W. Susilo, Y. Mu, and F. Zhang, "Perfect Concurrent Signature Schemes," in *Information and Communications Security*, ser. Lecture Notes in Computer Science, J. Lopez, S. Qing, and E. Okamoto, Eds. Springer Berlin / Heidelberg, 2004, vol. 3269, pp. 14–26.

[8] S. Chow and W. Susilo, "Generic Construction of (Identity-Based) Perfect Concurrent Signatures," in *Information and Communications Security*, ser. Lecture Notes in Computer Science, S. Qing, W. Mao, J. López, and G. Wang, Eds. Springer Berlin / Heidelberg, 2005, vol. 3783, pp. 194–206.

[9] J. Liu, R. Sun, W. Ma, Y. Li, and X. Wang, "Fair exchange signature schemes," in *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, Mar. 2008, pp. 422 –427.

[10] "OpenSSL Project," Apr. 2012. [Online]. Available: http://www.openssl.org/

[11] "Pairing-Based Cryptography Library," Apr. 2012. [Online]. Available: http://crypto.stanford.edu/pbc/

[12] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, Mar. 2003.

[13] F. Hess, "Efficient Identity Based Signature Schemes Based on Pairings," in *Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography*, ser. SAC '02. London, UK, UK: Springer-Verlag, 2003, pp. 310–324.

[14] "Mobile-C: a Multi-Agent Platform for Mobile C/C++ Agents," Apr. 2012. [Online]. Available: http://www.mobilec.org/

[15] 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. 2011.