Uso de rutas cacheadas en el encaminamiento seguro basado en DSR

José Luis Tornos
Centro Politécnico Superior
Universidad de Zaragoza
Email: jltornos@unizar.es

José Luis Salazar
Centro Politécnico Superior
Universidad de Zaragoza
Email: jsalazar@unizar.es

Joan Josep Piles
Centro Politécnico Superior
Universidad de Zaragoza
Email: jpiles@unizar.es

***Resumen*—DSR is a simple and efficient routing protocol in ad hoc networks. This paper is based on a previous variation of the DSR protocol where security is added using aggregate signatures. Our proposal uses an additional Route Discovery Feature of the original protocol, to reduce the number of messages needed around the network to answer a Route Request. We describe how the cached routes of the intermediate nodes can do this work without losing the security level.**

I. INTRODUCCÍON

Una red ad-hoc es aquella en la que no existe ninguna infraestructura de comunicación definida. Normalmente se define sobre dispositivos móviles y a la falta de infraestructura se añade la movilidad de los dispositivos que la conforman, por lo que el sistema es dinámico y admite variaciones.

Debido a la falta de infraestructura predefinida y al dinamismo del sistema, para la comunicación entre nodos sin conexión directa se requiere la cooperación de los nodos intermedios. Existen múltiples protocolos específicos para este tipo de redes ([1], [2], [3], [4], [5], [6]), en los que los métodos de encaminamiento tradicionales no resultan eficientes debido al dinamismo de los dispositivos. Estos protocolos son vulnerables al no tener en cuenta la seguridad de los mismos. Para entornos en los que la seguridad es un requisito importante se han definido protocolos de encaminamiento seguro ([7], [8], [9], [10], [11]), en los que existe un compromiso entre nivel de seguridad ofrecido, ancho de banda, necesidad de procesamiento y potencia necesaria.

Este artículo se fundamenta en el protocolo DSR [12] y profundiza en el planteamiento de un encaminamiento seguro basado en este protocolo [11]. En este esquema se emplean las firmas agregadas como primitiva criptográfica que permiten que M usuarios firmen M mensajes diferentes y las M firmas resultantes sean compactadas en una sola. Gracias a esta característica se consigue reducir el tamaño del campo de firmas, y por tanto, el tamaño del mensaje que se transmitirá. Como contrapartida tendremos que la verificación deberá ser realizada sobre el conjunto de las firmas, no pudiendo verificarse cada una de ellas de modo individual. Es decir, si la firma final es correcta, se validan todos los mensajes que la componen, pero si es incorrecta, no seremos capaces de saber qué mensaje, o mensajes, han sido los que la han invalidado.

El aporte que realiza este artículo, es el uso de las rutas cacheadas para responder a los paquetes de solicitud de ruta, *Route Request (RR)*, opción recogida en el protocolo DSR, manteniendo las características de seguridad preestablecidas. Se desarrolla un método para emplear las rutas cacheadas de los nodos intermedios para ofrecer un encaminamiento seguro que reduciría los mensajes que circulan por la red. También se conseguirá reducir el tiempo necesario para la obtención de una ruta, en el caso de que un nodo intermedio disponga de una ruta cacheada hacia el nodo destino requerido en el paquete *RR*.

En la sección II se presenta brevemente el protocolo DSR y las firmas agregadas. En la sección III desarrollamos nuestra propuesta. Y en la sección IV recogemos las conclusiones y las líneas futuras de trabajo.

II. BACKGROUND

A continuación vamos a recordar los dos elementos fundamentales sobre los que se basa nuestra propuesta: el protocolo de encaminamiento DSR seguro y la primitiva criptográfica de firmas agregadas.

II-A. DSR seguro

Aunque no existe ninguna especificación estándar del protocolo DSR seguro, sí que hay alguna propuesta sobre la mesa basado en DSR ([9], [10], [11]). Empezaremos recordando el DSR básico, para luego explicar la versión segura de [11].

II-A1. DSR: DSR es un protocolo de encaminamiento en el que el nodo origen establece la ruta a seguir en la red hasta alcanzar el nodo destino. Por tanto, en el mensaje de datos enviado están listadas las direcciones de los nodos intermedios que debe atravesar el paquete. Si un nodo tiene que comunicarse con otro para el que no conoce una ruta, envía un paquete *RR*, que será retransmitido por los nodos de la red hasta alcanzar su objetivo. Cuando el paquete *RR* llega al nodo destino, éste responderá con un paquete de respuesta, *Route Reply (REP)*, en el que aparecerán los nodos intermedios que deben procesar los paquetes para comunicar los nodos extremos.

Cada nodo mantendrá una tabla con las rutas conocidas, a las que se añade una marca temporal. Si una ruta no es utilizada dentro del margen temporal será borrada de la tabla de rutas conocidas.

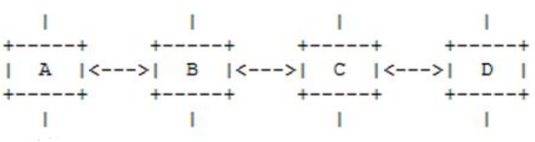

Figura 1: Distribución de nodos en la red ejemplo.

II-A2. Resumen del protocolo DSR seguro: El esquema descrito en [11] para el descubrimiento de rutas de modo seguro, establece que el nodo que quiere obtener una ruta hacia otro nodo, genera un paquete *RR* que firma y propaga en modo multidifusión (broadcast). Los nodos que reciben el mensaje comprueban si son el nodo destino y en caso de no serlo añaden su dirección a la ruta, firman y agregan su firma a la anterior y propagan el mensaje en modo multidifusión.

Cuando el mensaje llega al nodo destino, éste comprueba la firma agregada. Si es correcta, crea un paquete *REP* con la ruta que especificaba el paquete *RR* y lo envía por el camino inverso al que se ha recibido. Cada uno de los nodos intermedios recibe el paquete *REP*, agrega su firma al mensaje y lo envía al siguiente nodo de la lista. Cuando el mensaje llega al nodo que originó el paquete de descubrimiento de ruta, comprueba la validez de la firma agregada y, en caso de ser correcta, añade la ruta recibida a su tabla de rutas.

II-A3. Firmas agregadas: Las firmas agregadas son un concepto criptográfico relacionado con las multifirmas [13]. En el caso de las firmas agregadas, el escenario es un conjunto de usuarios U, cada uno de ellos con su par de claves pública y privada (K_{u+}, K_{u-}). Entonces, dado un subconjunto U'$\subseteq$U, cada usuario u$\in$U', produce una firma σ_u de un mensaje M_u, distinto para cada usuario. Las firmas obtenidas podrán agregarse formando una única firma. Para verificar la firma agregada será necesario tener el acceso a las claves públicas de los usuarios que han firmado cada uno de los mensajes, así como a sus correspondientes mensajes.

Se han desarrollado firmas agregadas en paralelo [14] y secuenciales [15]. Las firmas agregadas en paralelo permiten su verificación sin tener en cuenta el orden en el que se realizó la agregación. Mientras que las firmas agregadas secuenciales deben ser verificadas en el mismo orden en el que se realizó la agregación. Existe otro tipo de esquema de firmas agregadas basadas en identidad [16]. Este esquema elimina la necesidad de certificados pero requiere de una entidad confiable maestra. Otra característica de las firmas agregadas es la longitud de la firma final, que no se mantiene constante para todas las implementaciones. La propuesta elegida es la descrita por Boneh [14], basada en aplicaciones bilineales, y que sí mantiene constante la longitud de la firma agregada e implementa un esquema de firma en paralelo.

III. Empleo de rutas cacheadas en DSR seguro

Partimos del protocolo ya descrito [11] y de sus mecanismos para realizar el descubrimiento de rutas. Al emplear criptografía de clave pública, cada nodo firmará los paquetes, antes de proceder a su retransmisión, con su clave privada. Supondremos la existencia de una autoridad de certificación confiable AC, cuya clave pública es conocida por todos los nodos. La gestión de los certificados y sus revocaciones queda también fuera del ámbito de este trabajo, existiendo diversos esquemas ([17], [18], [19]) que pueden ser utilizados.

Con estas premisas, se va a explicar un método con el que se aprovechan las rutas cacheadas de los nodos intermedios para elaborar paquetes de respuesta de descubrimiento de ruta. Con ello, se consigue una respuesta más rápida y con una necesidad menor de recursos que los requeridos sin esta opción y manteniendo el nivel de seguridad previo.

III-A. Estructura de la tabla de rutas cacheadas

Para poder utilizar las rutas cacheadas para responder a las solicitudes de ruta de otros nodos, deberemos almacenar el paquete con el que se haya verificado esa ruta. También será necesario almacenar las direcciones de los nodos origen y destino que originen el paquete así como el número de identificación de ruta asociado al mismo. La necesidad de almacenar estos valores radica en que son necesarios para elaborar la firma de los paquetes, como se explica en el siguiente apartado.

Además los nodos no añadirán rutas a su tabla solamente cuando ellos son los que originan un paquete de *RR*, sino que cuando reciban paquetes *RR* y *REP*, también serán capaces de actualizar su tabla de rutas, siempre y cuando la verificación de las firmas sea correcta.

Usaremos como ejemplo la distribución de nodos de la Figura 1 y que el nodo A inicia un descubrimiento de ruta hacia el nodo D. Cuando el nodo C reciba un paquete *RR* firmado por los nodos A y B, tendrá un paquete con el que será capaz de validar una ruta hasta el nodo A a través de B. De este modo, si no la conociese de antemano, añadiría a su tabla de rutas el camino para comunicarse con los nodos A y B.

Esto que ocurre con los paquetes de *RR* para los nodos intermedios, también puede usarse con los paquetes *REP*. Por ejemplo, el nodo B recibe el paquete *REP*, que ha enviado el nodo D, y que ha pasado por el nodo C. Recibe el paquete en el que indica que la ruta para llegar del nodo A al nodo D es a través de B, el mismo, y después los nodos C y D. Al estar el paquete firmado por los nodos C y D, será capaz de verificar la validez de esa ruta para esos nodos. Así que será capaz de actualizar su tabla de rutas hacia los nodos C y D gracias al paquete *REP*.

III-B. Datos firmados de cada paquete

Independientemente del tipo de paquete, *RR* o *REP*, se firmarán siempre los siguientes datos y en este orden:

- Número de identificación de descubrimiento de ruta
- IP origen del nodo que ha iniciado el descubrimiento de ruta
- IP destino del nodo objetivo del descubrimiento de ruta
- IP de los nodos intermedios

K_{A-}	Clave privada del nodo A
K_{A+}	Clave pública del nodo A
$\{d\}K_{A-}$	Datos d firmados por el nodo A
$\{d\}K^{M}{}_{ABC-}$	Datos firmados por los nodos A,B y C y compactados en una firma agregada
$\{d\{d'\{d''\}\}\}K^{M}{}_{ABC-}$	Datos firmados por los nodos A,B y C y compactados en una firma agregada. El nodo A habrá formado d, el nodo B habrá firmado dd' y así sucesivamente

Tabla I: Lista de abreviaciones

Con este formato, se unifica la estructura de los datos sobre los que se calcula la firma independientemente de si el paquete es *RR* o *REP*.

Dependiendo de si el paquete que se está enviando es un *RR* o un *REP*, tendremos que los nodos intermedios componen una ruta completa entre los nodos origen y destino, para los paquetes *REP*, o una ruta parcial que comunica el nodo origen con el último de los nodos intermedios que componen la lista del paquete, en el caso de los paquetes *RR*.

Un ejemplo del intercambio de mensajes, utilizando la distribución de los nodos de la Figura 1, en el que el nodo A quisiera obtener una ruta hasta el nodo D, sería el siguiente:

A $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D] $K^{M}{}_{A-}$
B $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D, IP_B] $K^{M}{}_{AB-}$
C $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D, IP_B, IP_C] $K^{M}{}_{ABC-}$
D $\rightarrow$ C: REP $\|$ [N_A, IP_D, IP_A] $K^{M}{}_{D-}$ $\|$ N_A
C $\rightarrow$ B: REP $\|$ [N_A, IP_D, IP_A, IP_C] $K^{M}{}_{DC-}$ $\|$ N_A
B $\rightarrow$ A: REP $\|$ [N_A, IP_D, IP_A, IP_C, IP_B] $K^{M}{}_{DCB-}$ $\|$ N_A

Junto al envío de los paquetes *REP*, se hará necesario el envío del valor del identificador de descubrimiento de ruta al que se responde, ya que este valor, que no se envía dentro del paquete *REP*, es necesario para validar la firma.

III-C. Descubrimiento de rutas mediante rutas cacheadas

En el ejemplo descrito en el apartado anterior, si se permitiese el uso de las rutas cacheadas para responder a los paquetes de descubrimiento de ruta y el nodo B tuviese una ruta almacenada para llegar al nodo D, el intercambio de mensajes se habría reducido hasta quedar de la siguiente manera:

A $\rightarrow$ multidifusión: RR $\|$ [N_A, IP_A, IP_D] $K^{M}{}_{A-}$
B $\rightarrow$ A: REPC $\|$ [*Firma de REPC*] $\|$ [*Datos verif. firma*]

Se aprecia que con esta opción disminuiría el número de mensajes que tienen que ser enviados por la red. La firma que acompaña al paquete *REPC*, Replay cacheado, tiene la misma longitud que el resto de firmas del sistema. Y los datos extra necesarios para la verificación de la firma serán dos direcciones IP y un número de identificación de ruta. Por lo que el tamaño de los datos extra enviados será siempre mucho menor que el necesario para la transmisión de un único paquete DSR por la red.

Para la implementación de esta propuesta se ha diseñado un tipo de paquete cuyo contenido habilite a los nodos la verificación de firmas que asegure la fiabilidad de las rutas.

III-C1. Contenido de REPC: El paquete *REPC* se compone de dos paquetes *REP* dentro del mismo mensaje DSR. Esta opción está recogida dentro del RFC que define el protocolo DSR [12]. El primero servirá para validar la ruta que se ha seguido mediante el paquete de descubrimiento de ruta y el segundo validará la ruta desde el nodo intermedio hasta el nodo final, es decir, la ruta cacheada. Este segundo paquete deberá ser el que el nodo intermedio usó para validar la ruta entre él mismo y el nodo destino.

La variación que habrá que realizar sobre la opción ya recogida en el RFC, será la utilización de un bit, dentro de la zona reservada, que indique al nodo que reciba este tipo de paquetes que la forma de procesarlo será diferente a la recepción de un paquete *REP*. Otra opción sería utilizar un identificador de tipo, que indicase la función del mensaje.

III-C2. Procesado de REPC: Cuando un nodo reciba un paquete REPC sabrá que va a estar compuesto por dos paquetes *REP* y que el primero de ellos deberá procesarlo de manera habitual, sabiendo que la confirmación de ruta será parcial. El segundo paquete *REP* incluirá una ruta que permita alcanzar el nodo destino desde el nodo intermedio que respondió al *RR*.

De manera genérica, la ruta que contiene el segundo paquete *REP* no incluirá exclusivamente los nodos que conecten el nodo intermedio con el nodo destino, sino que formará parte una ruta mayor. Esto implica que la ruta a validar incluya nodos para los que no se ha solicitado información, pero que serán necesarios para realizar la comprobación de la firma del mensaje.

La diferencia existente entre usar una ruta cacheada mediante un paquete *RR* o *REP*, será que en un paquete *REP* se firma la ruta final entre dos nodos y en uno de *RR* se firma una ruta parcial entre los nodos de origen y destino. A la hora de utilizar estos paquetes para añadir una ruta a la tabla de rutas cacheadas de un nodo no existe ningún problema. Pero al utilizar estos paquetes como respuesta en un paquete *REPC*, el nodo que ha solicitado el descubrimiento de ruta deberá ser capaz de diferenciar entre uno y otro, ya que los datos empleados para la firma varían dependiendo de si es un *RR* o un *REP*.

Destino	Saltos	Paquete y firma con el que se validó la ruta	bitC
A	–	RR ‖ [N_A, IP_A, IP_D] $K^M{}_{A-}$	0
C	–	REP ‖ [N_B, IP_D, IP_B, IP_C, IP_D] $K^M{}_{DC-}$	0
D	C	REP ‖ [N_B, IP_D, IP_B, IP_C, IP_D] $K^M{}_{DC-}$	0

Tabla II: Rutas cacheadas del nodo B

Este detalle el nodo destino del *REPC* lo soluciona al inspeccionar el paquete que responde con la ruta cacheada ya que, si al analizar las direcciones IP de los nodos, la última dirección IP de la ruta intermedia, se corresponde con el nodo origen, sabrá que se ha creado a través de un paquete *REP*. Si por el contrario el nodo que se indica como destino no aparece en la lista de nodos intermedios, se tratará de un paquete *RR*.

Una vez interpretado el segundo paquete *REP*, el nodo receptor será el encargado de extraer la información parcial de la ruta que requiere. Para esto deberá localizar el nodo que originó el paquete *REPC* e ir añadiendo a la ruta validada en el primer *REP* los nodos hasta alcanzar el nodo destino.

La posible aparición de bucles en la ruta final se evita empleando los mecanismos ya descritos para ello en el empleo de rutas cacheadas no seguras. Es decir, el nodo que responde a un *RR*, verifica que la ruta que va a proporcionar está libre de bucles. En caso de no disponer de una ruta cacheada con la que no forme bucles, procesará el paquete *RR* como un nodo intermedio normal y lo retransmitirá en modo multidifusión.

Para evitar problemas en los que el tamaño y procesado del paquete *REPC* aumenten en exceso, no se permite a los nodos utilizar rutas aprendidas mediante rutas cacheadas para originar paquetes *REPC*. En la tabla de rutas cacheadas se marcará un bit en aquellas rutas que hayan sido formadas empleando este mecanismo, bitC.

En cuanto a la nueva firma agregada se construye de forma análoga a [11]. Agregamos dos firmas construídas con los mismos parámetros criptográficos que contienen varias firmas agregadas. Dado que estamos trabajando sobre un grupo abeliano, las operaciones cumplen la propiedad asociativa y por lo tanto las firmas también. De esta manera obtenemos una firma de longitud igual a las dos que agregamos. Para verificar dicha firma sólo necesitaremos recopilar las claves públicas de cada uno de los signatarios agregados y los datos extra necesarios para reconstruir los mensajes originales firmados. Con esto la comprobación de la firma se convierte en un simple proceso de verificación de una firma agregada.

Otra de las cuestiones que se podrían plantear sería que una comunicación entre A y B no garantiza que se pueda realizar la comunicación entre B y A. Pero tomamos como supuesto que la comunicación se produce en modo bidireccional.

III-C3. Ejemplo de uso de ruta cacheada: Para explicar la generación del paquete de respuesta *REPC* se va a explicar empleando la distribución de nodos de la Figura 1. Y el caso en el que el nodo A inicia el descubrimiento de ruta hacia el nodo C. Para ello genera y transmite el mensaje formado por el paquete *RR* y la firma del mismo:

$$A \rightarrow \text{multidifusión: RR} \parallel [N_A, IP_A, IP_C]\ K^M{}_{A-}$$

Cuando le llega el paquete al nodo B, este busca en su tabla de rutas cacheadas, Tabla 2, y como tiene una ruta almacenada para llegar al nodo C, inicia la generación del paquete *REPC*.

Primero crea un paquete REP_1 como si él fuese el destino del paquete de descubrimiento de ruta, el cual no tendrá nodos intermedios ya que la conexión entre A y B es directa, y lo firmará.

El paquete REP_2, lo genera con la ruta del paquete que empleó para guardar la ruta almacenada en su tabla de rutas cacheadas. En este caso concreto, ese paquete es un *REP* que se inició en el nodo D y cuyo destinatario era el propio nodo B. Como vemos, el nodo C se encuentra dentro de la ruta que conecta los nodos B y D. Por lo tanto, la ruta de nodos que indicaremos en REP_2 será C−D y los nodos origen y destino asociados serán D y B. Como el nodo origen coincide con el último de los saltos indicados, el nodo A será capaz de distinguir que este paquete que emplea una ruta cacheada se ha llevado a cabo a través de un paquete *REP* y no un *RR*.

Una vez que ya tiene los dos paquetes generados, el nodo B agrega la firma generada para el paquete REP_1 con la que tenía almacenada en la tabla de rutas, correspondiente a REP_2. En este momento ya será capaz de enviar el mensaje de respuesta al nodo A:

$$B \rightarrow A\text{: REPC} \parallel [\textit{Firma de REPC}] \parallel N_B \parallel IP_D \parallel IP_B$$

El nodo A será capaz con los datos facilitados de reconstruir los mensajes, y cada uno de los mensajes firmados por los nodos intermedios que componen la ruta para así verificar la ruta final. Es decir, será capaz de reconstruir una ruta segura desde A hasta el nodo intermedio B gracias a la primera parte del paquete *REPC*. Y después podrá conocer una ruta que una ese nodo intermedio B hasta el nodo destino C, gracias a la segunda parte del paquete *REPC*.

Como la ruta que almacenará la habrá descubierto gracias a la utilización de la ruta cacheada de otro nodo, deberá marcar el bitC de su tabla de rutas para no emplear esta ruta como respuesta a un *RR*.

En algunos casos, como en el del ejemplo, la ruta cacheada que permite a un nodo alcanzar el destino requerido, contiene más información de la solicitada. En este caso, el nodo A obtendrá también una ruta hasta el nodo D que podrá añadir a su tabla de rutas, marcando el bitC que indicará que conoce esa ruta como respuesta de una ruta cacheada.

IV. Conclusion

El empleo de las rutas cacheadas para responder a las solicitudes de descubrimiento de ruta, permite disminuir el número de paquetes que deben ser transmitidos por la red para lograr una ruta válida, sin disminuir por ello la seguridad alcanzada en [11]. Esto se ha conseguido utilizando las opciones y los tipos de paquetes descritos en el RFC que define el protocolo DSR [12]. Como único añadido se requiere el empleo de un bit de la zona reservada dentro del paquete de respuesta *REP*.

Además, el empleo de las rutas cacheadas ha dado como resultado que un nodo pueda aprender más rutas que la esperada y añadir más información a su tabla de rutas.

Como futuras mejoras se plantea que en los paquetes de descubrimiento de ruta, habrá que establecer un método, normalmente mediante la activación de un bit en el *RR*, que permita paquetes *REP* formados con rutas cacheadas.

El nivel de seguridad no varía de la anterior propuesta, ya que se emplea la misma primitiva criptográfica, sobre otro tipo de datos, que permiten ser verificados en una extensión mayor que la original. La única posibilidad que se vislumbra en el ataque es la posible inclusión de rutas caducadas, para lo que sería necesario establecer algún tipo de sincronización y sellado en el tiempo que será motivo de posteriores estudios.

Referencias

[1] Perkins, C.E., Royer, E.M.: “Ad-hoc on-demand distance vector routing´´, en WMCSA ’99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, IEEE Computer Society (1999) 90–100

[2] Johnson, D.B., Maltz, D.A.: “Dynamic source routing in ad hoc wireless networks´´, en Imielinski, Korth, eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996)

[3] Perkins, C.E., Bhagwat, P.: “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers´´, en SIGCOMM ’94: Proceedings of the conference on Communications architectures, protocols and applications, ACM Press (1994) 234–244

[4] Murthy, S., Garcia-Luna-Aceves, J.J.:“ An efficient routing protocol for wireless networks´´. Mob. Netw. Appl. 1(2) (1996) 183–197

[5] Park, V.D., Corson, M.S.:“A highly adaptive distributed routing algorithm for mobile wireless networks´´, en INFOCOM ’97: Proceedings of the INFOCOM ’97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, IEEE Computer Society (1997) 1405

[6] Toh, C.K.:“A novel distributed routing protocol to support ad-hoc mobile computing´´,en Proceedings of 15 IEEE Annual International Phenix Conference on Computers and Communications. (1996) 480–486

[7] Zapata, M.G., Asokan, N.: “Securing ad hoc routing protocols´´,en WiSE ’02: Proceedings of the 3rd ACM workshop on Wireless security, ACM Press (2002) 1–10

[8] Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., Belding-Royer, E.M.: “ A secure routing protocol for ad hoc networks´´, en ICNP ’02: Proceedings of the 10th IEEE International Conference on Network Protocols, IEEE Computer Society (2002) 78–89

[9] Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: “A secure on-demand routing protocol for ad hoc networks´´, en MobiCom ’02: Proceedings of the 8th annual international conference on Mobile computing and networking, ACM Press (2002) 12–23

[10] Kim, J., Tsudik, G.: Srdp: “Securing route discovery in dsr´´, en The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. (2005) 247–260

[11] Piles,J.J, Salazar, J.L.: “Encaminamiento seguro para redes ad-hoc´´, en IX Reunión Española sobre criptología y seguridad de la información (RECSI 2006) pp. 732–744. Septiembre. 2006.

[12] David B. Johnson, Yih-Chun Hu, and David A. Maltz, ”The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4”, Internet Request for Comments RFC 4728, February 2007.

[13] Okamoto, T.: “A digital multisignature scheme using bijective public-key cryptosystems´´, en ACM Trans. Comput. Syst. 6(4) (1988) 432–441

[14] Boneh, D., Gentry, C., Lynn, B., Shacham, H.: “Aggregate and verifiably encrypted signatures from bilinear maps´´, en Cryptology ePrint Archive, Report 2002/175. Volume 2656 of Lecture Notes in Computer Science. (2002) 416–432

[15] Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: “Sequential aggregate signatures from trapdoor permutations´´, en Proceedings of Eurocrypt 2004. Volume 3027 of Lecture Notes on Computer Science. (2004) 74–90

[16] Herranz, J.: “Deterministic identity-based signatures for partial aggregation´´, en The Computer Journal 49(3) (2006) 322–330

[17] Crépeau, C., Davis, C.R.: “A certificate revocation scheme for wireless ad hoc networks´´, en SASN ’03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, ACM Press (2003) 54–61

[18] Salazar, J.L., Ruíz, J., Gallardo, P.: “Desarrollo de un entorno seguro de comunicación en una red adhoc´´, en RECSI ’04: VIII Reunión Española sobre Criptología y Seguridad de la Información. (2004) 447–454

[19] Luo, J., Hubaux, J.P., Eugster, P.T.: “DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks´´, en IEEE Transactions on Dependable and Secure Computing 2(4) (2005) 311–323