INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

PACDRE: PROTOCOLO PARA ADMINISTRACIÓN DE LLAVES PARA REDES AD HOC CON DOS REPOSITORIOS

TESIS QUE PARA OPTAR EL GRADO DE
MAESTRO EN CIENCIAS COMPUTACIONALES
PRESENTA

GERARDO FRANCISCO DEL VALLE TORRES

Asesor: Dr. ROBERTO GÓMEZ CÁRDENAS

Comité de tesis: Dr. Nora Erika Sánchez Velázquez
M.C.C Adolfo Grego Kibrit
Dr. Eduardo García García

Atizapán de Zaragoza, Edo. Méx., Enero de 2005.

INTRODUCCIÓN

Las redes inalámbricas se pueden definir como un conjunto de nodos que se comunican entre sí a través de un canal inalámbrico. [2] Existen dos formas de configurar una red inalámbrica, la primera es con infraestructura fija, en la cual todos los nodos se conectan a un punto de acceso (AP) y la segunda cuando no existe ninguna infraestructura fija y en donde cada nodo de la red puede de una forma libre y dinámica entablar comunicación con otro nodo, este tipo de redes son conocidas como AD HOC. La mayoría de los nodos en una red AD HOC son móviles, lo que implica que la comunicación sea inalámbrica, estas redes también son conocidas como MANETS[1] (*Mobile Ad Hoc Networks*).

Los avances en las comunicaciones inalámbricas y los dispositivos portátiles pequeños y ligeros han hecho posible la computación móvil. Aunque los términos red móvil y red inalámbrica se utilizan juntos no se refieren a lo mismo puesto que los usuarios móviles no necesariamente requieren de interfaces inalámbricas, y las interfaces inalámbricas no necesariamente soportan movilidad. La popularidad de la computación móvil ha tenido un enorme crecimiento. La continúa miniaturización de dispositivos de cómputo móviles y el extraordinario incremento de poder de procesamiento disponible en dispositivos portátiles ha hecho posible introducir mejores aplicaciones en beneficio de los usuarios que día a día va en aumento. [10]

Las MANETS tienen problemas de seguridad específicos, la mayoría de las investigaciones realizadas se enfocan a la parte de ruteo y asumen que existe un acuerdo para establecer comunicaciones seguras entre nodos. Para establecer una comunicación segura, es necesario contar con una administración de llaves, para crear canales seguros de comunicación entre los nodos que conforman una MANET. El manejo de llaves, requiere del uso de una autoridad certificadora (CA) así como la expedición, renovación y revocación de certificados en MANETS. Existen propuestas para la administración de llaves las cuales hemos retomado como base para realizar un análisis de las ventajas y desventajas que ofrecen.

[1] En adelante nos referiremos a redes AD HOC como MANETS

La principal desventaja de alguna de las soluciones, es que centralizan la autoridad certificadora y las MANETS son distribuidas por naturaleza, por lo que el implementar éstas soluciones requieren de adaptaciones o adecuaciones. Otras soluciones hacen uso de un gran numero de mensajes y otras mas no contemplan un mecanismo para revocar, actualizar o renovar certificados.

El objetivo de este trabajo es presentar el protocolo PACDRE, que retoma lo mejor de diferentes propuestas y que puede ser implementado en MANETS. Nuestra propuesta esta inspirada del protocolo PGP y hace uso del concepto de programas de almacenamiento de SQL, que se ejecutan de acuerdo a una calendarización, sin que el usuario tenga que realizar alguna tarea adicional para ello. El protocolo permite la renovación, revocacion y actualización de certificados.

El presente trabajo se encuentra organizado de la siguiente forma: en el primer capítulo se introduce al lector en MANETS y criptografía. En el segundo capítulo se aborda la problemática de seguridad que presentan las MANETS y se presentan algunos tópicos básicos de seguridad en redes. El tercer capítulo esta dedicado a las soluciones existentes para la administración de llaves en MANETS. El cuarto capítulo se enfoca al análisis de cada una de las soluciones, presentado sus ventajas y desvetajas. El quinto capítulo se presenta a PACDRE, nuestra propuesta para el manejo de llaves en MANETS. Por último, en el sexto capítulo, se presentan nuestras conclusiones y posibles trabajos futuros.

GRACIAS

A mis Padres:

Papa por ser mi ejemplo y mi guía. Sé que desde donde estés te sentirás satisfecho por el nuevo objetivo alcanzado.

Mama por tu esfuerzo, amor, solidaridad y apoyo incondicional en todo momento que fueron imprescindibles para lograr esta meta.

A mis Hermanos y Familiares:

Chayo, Lulú, Martha y Pepis por creer en mi.
A mis familiares por el aliento y la confianza que me otorgaron

Al Tecnológico de Monterrey y Profesores:

Por abrirme una puerta de conocimiento y darme la oportunidad de obtener el conocimiento necesario para poder desarrollarme en el ámbito profesional. En especial al Dr. Roberto Gómez por su guía en este trabajo.

A Cathy:

Por ser mi inspiración cada día, por creer y confiar en que se podía alcanzar está meta.

A mis amigos y Compadres:

Por su cercana compañía y por todos los momentos que me han permitido compartir con ustedes.

En especial a mi amigo Edgar por contagiarme el entusiasmo de escalar montañas más altas.

RESUMEN

La tendencia en redes es clara, y esta orientada hacia lo inalámbrico. La empresa consultora *Garter* anunció que para el 2007 esperan que las redes inalámbricas sean comunes, ya sea en las organizaciones empresariales, instituciones educativas o gobierno.

El gran problema que tienen este tipo de redes es la seguridad. En una red alámbrica existen ciertas medidas de seguridad natural, donde los datos los contiene un entorno tangible (cable). En este sentido las redes inalámbricas representan un reto, ya que los datos viajan a través de ondas de radio y representa un escenario distinto desde su configuración a comparación con las redes alámbricas.

Existen dos formas de configurar una red inalámbrica, la primera es con infraestructura fija, en donde todos los nodos se conectan a un punto de acceso (AP) y la segunda que es cuando no existe ninguna infraestructura y en donde cada nodo de la red puede de una forma libre y dinámica entablar comunicación con otro nodo, este tipo de redes son conocidas como AD HOC.

La mayoría de los nodos en una red AD HOC son móviles, lo que implica que la comunicación sea inalámbrica, estas redes también son conocidas como MANETS (*Mobile Ad Hoc Networks*).

Las MANETS tienen problemas de seguridad específicos. Uno de ellos es la administración de llaves entre nodos. La mayoría de las investigaciones realizadas en MANETS en materia de seguridad, se enfocan a la parte de ruteo y asumen que existe un acuerdo para establecer comunicaciones seguras entre nodos.

Para resolver el problema, de que un nodo se pueda comunicar en forma segura con otro, se desarrolló en este trabajo una solución denominada *PACDRE*, que está basado en un protocolo de *Autoemisión de Certificados* al cuál se le incorporó un repositorio adicional para la administración de certificados vencidos. Cada repositorio es manipulado por procedimientos de almacenado que están programados con el lenguaje SQL, que es el estándar ANSI para manipulación de base de datos.

Con PACDRE se resuelve el problema de administración de llaves, porque cada nodo es responsable de sus certificados, sin que dependan de ninguna autoridad certificadora centralizada o semicentralizada, además de que puede generar, controlar, administrar, revocar y renovar sus llaves, disminuyendo el riesgo de que estas puedan ser comprometidas por un adversario.

CONTENIDO

ÍNDICE DE FIGURAS..... 7

ÍNDICE DE TABLAS..... 9

INTRODUCCIÓN..... 10

1. REDES INALÁMBRICAS AD HOC (MANETS) Y CRIPTOGRAFÍA...12

1.1 CAMBIOS EN LA MAC 13
1.1.1 ESTÁNDARES 802.11 15
1.2 REDES AD HOC 17
1.2.1 CARACTERÍSTICAS DE LAS MANETS 21
1.2.1.1 Características de las MANETS de acuerdo a la aplicación 22
1.2.2 APLICACIONES PARA MANETS 23
1.2.3 PROTOCOLOS DE RUTEO EN MANETS 25
1.2.3.1 Tipos de protocolo de ruteo para MANETS..... 26
1.3 INTRODUCCIÓN A LA CRIPTOGRAFÍA 28
1.3.1 ENCRIPCIÓN SIMÉTRICA..... 28
1.3.2 ENCRIPCIÓN DE LLAVE PÚBLICA..... 30
1.3.2.1 Diffie-Hellman (DH) 31
1.3.2.2. RSA 32
1.3.3 HUELLA Y FIRMA DIGITAL 33
1.3.4 CERTIFICADO DIGITAL 36
1.3.5 SECRETOS COMPARTIDOS 37
1.3.6 ADMINISTRACIÓN DE LLAVES..... 38
1.3.6.1 Administración de llaves a través de técnicas de llave simétrica..... 39
1.3.6.2 Administración de llaves a través de técnicas de llave pública 41

2. PROBLEMAS DE SEGURIDAD EN MANETS.....46

2.1 SEGURIDAD EN REDES 46
2.1.1 SERVICIOS DE SEGURIDAD..... 46
2.1.2 TIPOS DE ATAQUES 47
2.1.3 PROPIEDADES DE UN SISTEMA SEGUROS..... 48
2.2 PROBLEMAS DE SEGURIDAD EN MANETS. 49
2.2.1 PRINCIPALES PROBLEMAS DE SEGURIDAD EN MANETS..... 50
2.3 ASPECTOS DE SEGURIDAD EN MANETS 58
2.3.1 SISTEMAS DE DETECCIÓN DE INTRUSIONES 58
2.3.2 SEGURIDAD EN EL ENRUTAMIENTO 59
2.3.3. SERVICIOS DE DISTRIBUCIÓN DE LLAVES..... 61

3. SOLUCIONES PARA LA ADMINISTRACIÓN DE LLAVES EN MANETS 63

3.1 ADMINISTRACIÓN DE LLAVES EN MANETS 64

3.2 AUTORIDAD CERTIFICADORA (CA) PARCIALMENTE DISTRIBUIDA 64

3.2.1 ENVÍO, RENOVACIÓN Y RECUPERACIÓN DE CERTIFICADOS ... 65
3.2.2 MANTENIMIENTO DEL SISTEMA..... 66
3.3 AUTORIDAD CERTIFICADORA (CA) COMPLETAMENTE DISTRIBUIDA 66
3.3.1 MANTENIMIENTO DEL SISTEMA 68
3.3.2 ACTUALIZACIÓN DE LAS *K-PARTES* DE LA LLAVE PRIVADA SK_{CA} DE LA CA. 70
3.3.3 EMISIÓN Y RENOVACIÓN DE CERTIFICADOS..... 72
3.3.4 REVOCACIÓN DE CERTIFICADOS 74
3.4 AUTOEMISIÓN DE CERTIFICADOS 75
3.4.1 "*SMALL-WORLD*"..... 77
3.4.2 ALGORITMO "*SHORTCUT HUNTER*" 78
3.5 PEBBLENETS 79
3.5.1 PARÁMETROS DE LA SOLUCIÓN..... 80
3.5.2 FUNCIONES CRIPTOGRÁFICAS 82
3.5.3 FASE DE GENERACIÓN DE CLUSTER 82
3.5.4 FASE DE ACTUALIZACIÓN DE LLAVE. 84
3.6 IDENTIFICACIÓN DEMOSTRATIVA 84
3.6.1 INTERCAMBIO DE LLAVES 85

4. ANÁLISIS DE LAS SOLUCIONES PRESENTADAS.....87

4.1 ANÁLISIS DE LA SOLUCIÓN AUTORIDAD CERTIFICADORA PARCIALMENTE DISTRIBUIDA..... 87
4.2 ANÁLISIS DE LA SOLUCIÓN AUTORIDAD CERTIFICADORA COMPLETAMENTE DISTRIBUIDA 90
4.3 ANÁLISIS DE LA SOLUCIÓN AUTOEMISIÓN DE CERTIFICADOS 91
4.4 ANÁLISIS DE LA SOLUCIÓN PEBBLENETS 92
4.5 ANÁLISIS DE LA SOLUCIÓN DE IDENTIFICACIÓN DEMOSTRATIVA 93
4.6 COMPARATIVO DE SOLUCIONES..... 94

5. PACDRE: PROTOCOLO DE AUTOEMISIÓN DE CERTIFICADOS CON DOS REPOSITORIOS.....96

5.1 MARCO DE TRABAJO 97
5.2 MECANISMO PARA MANTENER EL REPOSITORIO DE CERTIFICADOS ACTUALIZADOS……98
5.3 MECANISMO PARA REVOCACIÓN Y RENOVACIÓN DE CERTIFICADOS.....102
5.4 COMPARATIVO DE PACDRE CON RESPECTO A LAS

SOLUCIONES PRESENTADAS.....107

6. CONCLUSIONES.....109
6.1 TRABAJO FUTURO.....110

REFERENCIAS.....111

ÍNDICE DE FIGURAS

Figura 1.1. Familia del estándar 802 y su relación con el modelo OSI..... 13
Figura 1.2. Frame genérico de 802.11 14
Figura 1.3. Envío de un acuse de recibo entre dos nodos..... 14
Figura 1.4. Envío de un acuse de recibo entre dos nodos..... 15
Figura 1.5. Red inalámbrica con infraestructura 17
Figura 1.6. MANETS en donde cada nodo se comunica con otro sin la necesidad de un punto de acceso (AP). 18
Figura 1.7. Frame genérico de 802.11 para MANETS..... 18
Figura 1.8. Tarjeta inalambrica PCMCIA. 18
Figura 1.9. Menú conexiones de red..... 19
Figura 1.10. Icono conexiones de red inalámbrica..... 19
Figura 1.11. Menú de conexiones de red inalámbricas 20
Figura 1.12. Ventana Propiedades de conexiones de red inalámbricas. 20
Figura 1.13. Ventana de configuración de una red AD HOC. 21
Figura 1.14. Clasificación de los protocolos de ruteo para MANETS. 27
Figura 1.15. Esquema de encripción simetrica..... 29
Figura 1.16 Esquema de encripción de llave pública. 30
Figura 1.17 Intercambio de llaves en Diffie-Hellman. 31
Figura 1.18 Ejemplo de firma digital. 35
Figura 1.19 Ejemplo de firma digital. 37
Figura 1.20. Categorías de la TTP. 39
Figura 1.21. Ejemplo de *Kerberos*..... 40
Figura 1.22. Componentes de una *PKI*. 43
Figura 1.23 Estructura de una CRL. 45

Figura 2.1 Requerimientos de seguridad para las MANETS 49
Figura 2.2 Funcionamiento del algoritmo WEP en la modalidad de cifrado 52
Figura 2.3. Funcionamiento del algoritmo WEP en la modalidad de descifrado..... 53
Figura 2.4 WarWalking, tratando de localizar redes inalámbricas caminando..... 54
Figura 2.5. Bote para aumentar la ganancia de la tarjeta inalámbrica. 55
Figura 2.6 WarDriving localizando redes inalámbricas en el automóvil 55
Figura 2.7. Tipos de ataque a MANETS 58

Figura 3.1. Arquitectura del sistema que contiene tres nodos servidores..... 65
Figura 3.2. Autoridad certificadora completamente distribuida..... 67
Figura 3.3. Fase de arranque de inicialización 68
Figura 3.4. Inicialización durante la fase operacional..... 69
Figura 3.5 Fases que se presentan en el tiempo de vida en las MANETS 72
Figura 3.6. Proceso que realiza el nodo *p* para solicitar la renovación de su certificado..... 73
Figura 3.7. Revocación de certificados y mantenimiento distribuido de las listas de revocación de certificados..... 75
Figura 3.8. Ejemplo de PGP. 76

Figura 3.9. Cadena de certificados locales para autenticar un usuario 77
Figura 3.10. Ejemplo del fenómeno “small-world”. 77
Figura 3.11. Grafo que muestra la emisión de certificados aplicando el algoritmo Shortcut Hunter 78
Figura 3.12 Diferentes fases durante el tiempo de vida de la red AD-HOC. 80
Figura 3.13. Segmentación de clustes y generación de cluster *backbone* 83
Figura 3.14. Canal primario y canal limitado entre dos nodos. 85

Figura 4.1. Nodos interconectados, red AD-HOC no segmentada 88
Figura 4.2 Red Segmentada 88
Figura 4.3. Red reensamblada 89
Figura 4.4. Ejemplo de propagación de certificados con PGP 91

Figura 5.1. Usuario con su repositorio 99
Figura 5.2. Usuario con dos repositorios 99
Figura 5.3. Proceso de actualización de certificados vencidos 101
Figura 5.4. Diagrama de flujo del proceso de actualización de certificados 102
Figura 5.5. Proceso de revocación de certificados vencidos 104
Figura 5.6. Diagrama de flujo del proceso de revocación de certificados vencidos 104
Figura 5.7. Proceso de renovación de certificados 106
Figura 5.8. Diagrama del proceso de Proceso de renovación de certificados 106

ÍNDICE DE TABLAS

Tabla 1.1 Actualizaciones al estándar 802.11 16

Tabla 2.1 Ataques y defensa contra éstos..... 61

Tabla 4.1 Comparativo entre las soluciones 95

Tabla 5.1 Comparativo entre las soluciones que emplean revocación de certificados..... 108

1 REDES INALÁMBRICAS AD HOC (MANETS) Y CRIPTOGRAFÍA

Una red inalámbrica local representa un sistema de comunicaciones establecido a través del uso de la tecnología de radiofrecuencia, que puede funcionar como extensión de una red cableada.

El origen de este tipo de redes data de 1990, cuando se conforma el comité *IEEE* [2] 802.11, que empieza a trabajar con el objetivo de generar una norma para redes inalámbricas. En 1996, finalmente, un grupo de empresas del sector de cómputo móvil y de servicios forman el *WLI Forum*[3] para desarrollar este mercado mediante la creación de un amplio abanico de productos y de servicios interoperativos. Entre los miembros fundadores de *WLI Forum* se encuentran empresas como *ALPS Electronic, AMP, Data General, Contron, Seiko Epson y Zenith Data Systems.* [7]

En cuanto al diseño del protocolo, desde la perspectiva del modelo OSI, los desarrolladores del estándar 802.11 incorporan componentes en la capa física así como en la de enlace de datos, en la capa MAC se determina el conjunto de reglas de como un nodo accede al medio y envía datos, pero los detalles de la transmisión los realiza la capa física. [4]

La imagen 1.1.ilustra la relación que existe entre los diferentes componentes de la familia 802 y el lugar que ocupan dentro del modelo OSI. Las especificaciones individuales de la serie 802 se identifican por el segundo número, por ejemplo 802.3 es la especificación para *CSMA/CD*[4], el cuál está relacionado a *Ethernet*, y el 802.5 es para *Token Ring*.

---

[2] Por sus siglas en inglés Institute of Electrical and Electronics Engineers
[3] Ídem Wireless LAN Interoperability Forum
[4] Ídem Carrier Sense Multiple Access network with Collision Detection

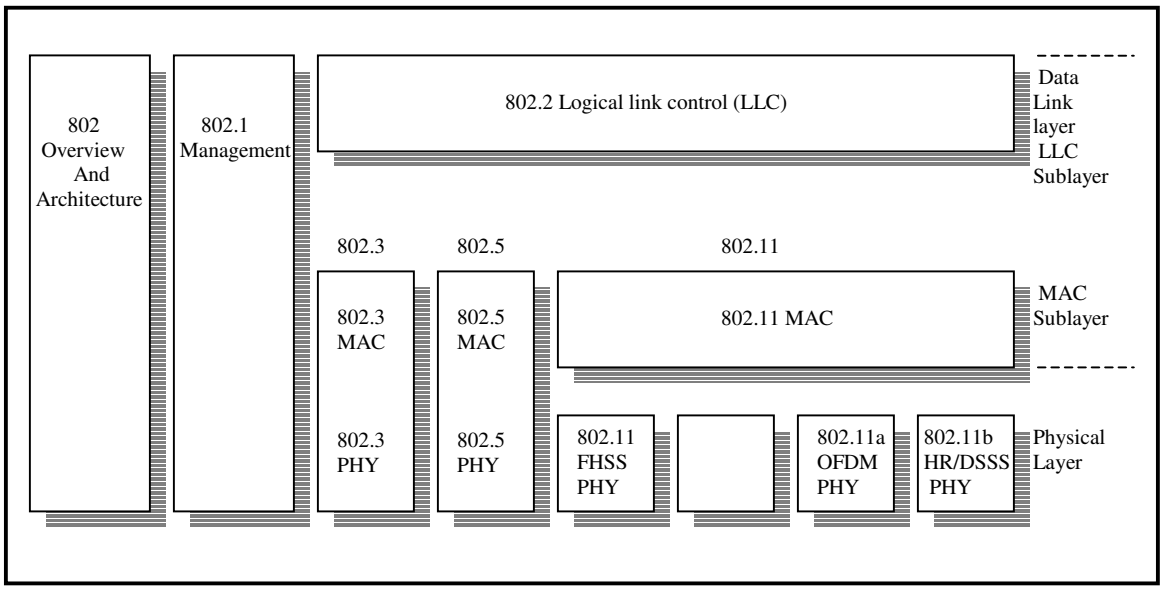


Figura 1.1. Familia del estándar 802 y su relación con el modelo OSI.

La especificación 802.11 incluye la capa de acceso al medio (MAC) y tres capas físicas: *FHSS*[5] , *DHSS*[6] *y OFDM*[7] . [4]

a) *FHSS*: Utiliza una portadora de banda estrecha que cambia la frecuencia a un patrón conocido por transmisor y receptor. Sincronizado en forma adecuada, es como tener un canal lógico único. Para un receptor no sincronizado *FHSS* es como un ruido de impulsos de corta duración.

b) *DSSS*: Genera un bit redundante por cada bit transmitido. Estos bits redundantes son llamados "*chipping code*". Entre mas grande sea la secuencia, mayor es la probabilidad de reconstruir los datos originales (también se requiere mayor ancho de banda). Para un receptor cualquiera *DSSS* es un ruido de baja potencia y es ignorado.

c) *OFDM*: Divide un canal disponible en diferentes subcanales y codifica una porción de la señal a través de cada subcanal en paralelo. La técnica es similar a la utilizada por algunos módems.

1.1 CAMBIOS EN LA MAC

Lo más trascendente de la especificación 802.11 es la MAC. Su principal función es el control de acceso al medio inalámbrico. Se encarga de la fragmentación, cifrado, manejo de la potencia y sincronización. Adicionalmente se encarga de proporcionar soporte de itinerancia en donde existen múltiples puntos de acceso. [4]

---

[5] Por sus siglas en inglés Frecuencv-Hopping Spread Spectrum.
[6] Ídem Direct-Sequence Spread Spectrum
[7] Ídem Orthogonal Frequency Division Multiplexing

Los frames de 802.11, no incluyen algunas de las características clásicas del frame de *Ethernet*, por ejemplo: el campo tipo/longitud y el preámbulo. El motivo de no incluir éstas, es que el preámbulo forma parte de la capa física, y los detalles de encapsulación que se manejan en el campo tipo/longitud, se cambiaron al encabezado del frame 802.11. [4]

La figura 1.2 muestra un frame 802.11 que puede soportar hasta 2346 bytes como máximo.

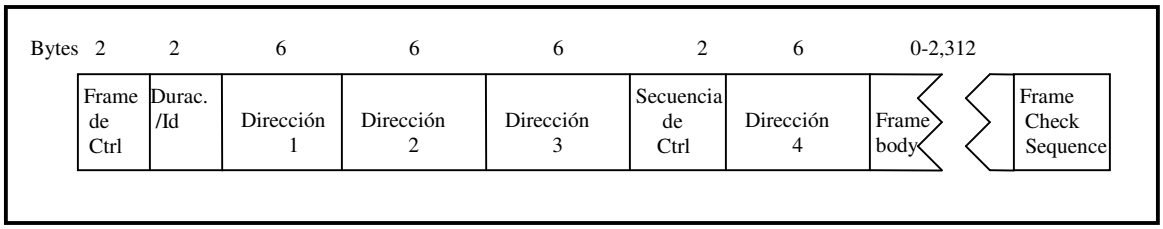


Figura 1.2. Frame genérico de 802.11.

En una red cableada cuando se transmite un frame se asume que llega a su destino en forma correcta, con radiofrecuencias es diferente, especialmente cuando las frecuencias se encuentran en un espectro que es de libre acceso, porque éstas se encuentran sujetas a ruido e interferencias.

Los diseñadores del 802.11 consideraron lo anterior, y a diferencia de otros protocolos, en la capa de enlace de datos incorporan un acuse de recibo en los frames. Todos los frames transmitidos entre los nodos deben tener un acuse de recibo, si por alguna razón el nodo no recibe éste el frame se considera perdido. [4]

La figura 1.3 ilustra una operación atómica en donde se muestra como el nodo A recibe el acuse de recibo del nodo B. Cabe señalar que el protocolo 802.11 permite a los nodos que estén realizando una operación atómica reservar el medio hasta que se realice la transmisión.

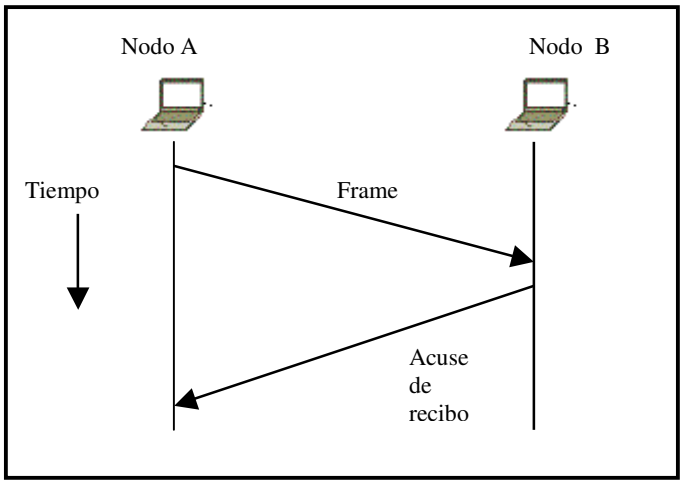


Figura 1.3. Envío de un acuse de recibo entre dos nodos.

En las redes cableadas, la recepción de transmisiones de los equipos de la red, dependen del mecanismo *CSMA/CD*[8], el medio físico contiene las señales y las distribuye a las estaciones de la red cableada. En las redes inalámbricas el estándar usa el mecanismo *CSMA/CA*[9] . [4]

[8] Por sus siglas en inglés Carrier Sense Multiple Access with Collision Detection.

La función de *CSMA/CA* es escuchar si el medio esta libre por un espacio de tiempo, si lo está, envía un paquete *RTS*[10] que contiene las direcciones origen y destino así como la duración de la siguiente transmisión.

El destino debe contestar con un paquete de reconocimiento *CTS*[11]. El resto de las máquinas activa su *VCS*[12] por la duración de la transmisión con lo que evitan que alguien trate de usar el medio. A continuación se ejemplifica el proceso descrito.

El nodo 1 requiere enviar un frame y para ello envía un frame *RTS*, éste frame además de reservar el enlace de radio para transmitir, silencia a las estaciones que estén escuchando. Si el nodo B recibe el frame *RTS*, éste responde con un frame *CTS* que al igual que frame *RTS,* silencia a todas las estaciones que estén dentro de su rango de transmisión.

Una vez que se ha completado el intercambio de frames *RTS/CTS*, el nodo 1 puede transmitir sus frames de datos sin que se presente ningún problema de interferencia y el nodo 2 por su parte una vez que el nodo1 finalice su transmisión enviará un frame de acuse de recibo. La figura 1.4 ejemplifica el proceso anterior

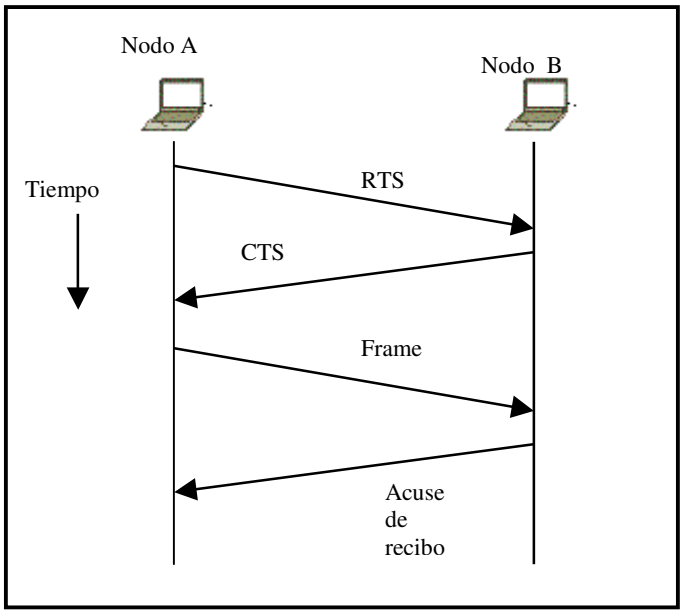


Figura 1.4. Envío de un acuse de recibo entre dos nodos.

1.1.1 ESTÁNDARES 802.11

[9] Ídem Carrier Sense Multiple Access with Collision Avoidance.
[10] Ídem Request to Send.
[11] Ídem Clear to Send.
[12] Ídem Virtual Carrier Sense.

Debido a que a raíz de su aparición surgieron dispositivos 802.11 de diferentes fabricantes, se hizo necesaria la existencia de recomendaciones (contenidas en los estándares), para permitir a los productos de estas firmas, una operación adecuada entre sí y que se cumpliera con un mínimo establecido de calidad y funcionalidades.

Los estándares de redes inalámbricas, como se ha mencionado, comenzaron con el estándar 802.11, desarrollado en los noventas por la *IEEE* y permitían una velocidad de transmisión de datos de hasta 2 Mbps, el protocolo se ha mejorado con el paso del tiempo y se han producido extensiones al mismo.

Las extensiones del protocolo se reconocen con la adición de una letra al estándar original, incluyendo 802.11a y 802.11b.

La tabla 1.1 muestra las variantes relacionadas con el estándar 802.11.

Tabla 1.1. Actualizaciones al estándar 802.11.

Estándar	Descripción
802.11	Estándar original. Soporta de 1 a 2 Mbps.
802.11a	Estándar de alta velocidad en la banda de los 5 GHz. Soporta hasta 54 Mbps.
802.11b	Estándar para la banda de 2.4 GHz. Soporta 11 Mbps.
802.11e	Está dirigido a los requerimientos de calidad de servicio para todas las interfaces IEEE 802.11.
802.11f	Define la comunicación entre puntos de acceso para facilitar redes 802.11 de diferentes proveedores.
802.11g	Establece una técnica de modulación adicional para la banda de los 2.4 GHz. Está dirigido a proporcionar velocidades de hasta 54 Mbps.
802.11h	Define la administración del espectro de la banda de los 5 GHz para su uso en Europa y en Asia Pacífico.
802.11i	Está dirigido a abatir la vulnerabilidad actual en la seguridad para protocolos de autenticación y de codificación. El estándar abarca los protocolos 802.1X, TKIP[13] y AES[14].

Dentro de las topologías de una red inalámbrica con 802.11 tenemos dos grandes grupos: el primero es con infraestructura o con punto de acceso (*AP*)[15] y el segundo es sin infraestructura o AD HOC.

[13] Por sus siglas en inglés Temporal Key IntegrityProtocol.
[14] Ídem Advanced Encryption System.
[15] Ídem Access Point.

En redes con infraestructura un nodo móvil se conecta y comunica con el AP más cercano que esta dentro de su radio de comunicación (ver figura 1.5.). Los nodos no se comunican unos con otros en forma directa, sino a través del AP. En éste tipo de redes, un AP puede dar servicio a 20 clientes y si se utilizan pocos recursos puede servir hasta 50. [4]

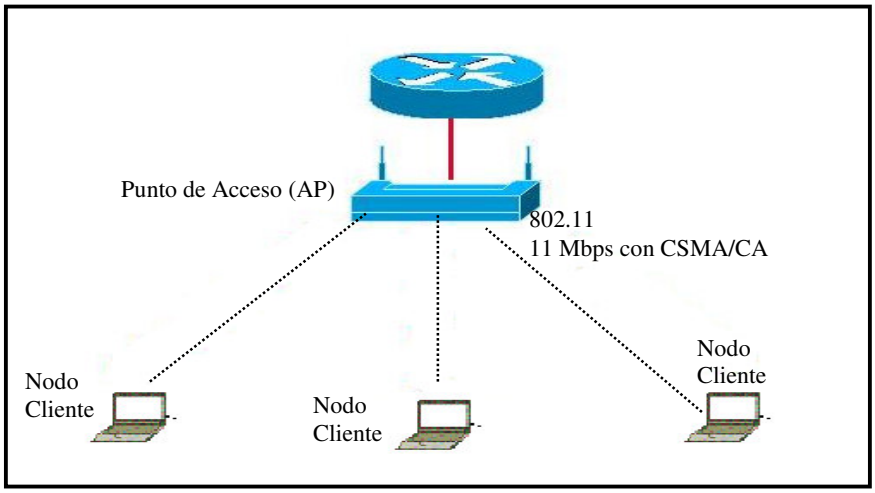


Figura 1.5. Red inalámbrica con infraestructura

1.2 REDES AD HOC

El segundo tipo de red móvil inalámbrica es la red sin infraestructura, comúnmente llamada red AD HOC. El origen de las redes AD HOC, no es nuevo y su origen se remonta a la época del rey de Persia *Darius I* (522-486 A.C) que entre otros desarrollos, creó un sistema de comunicación innovador, por medio del cuál enviaba mensajes o noticias de la capital, donde él se ubicaba, a las provincias mas lejanas, para ello ubicaba a los "mensajeros" en posiciones estratégicas que se encontraban en estructuras altas, de tal forma, que pudieran escuchar los gritos de su vecino más cercano y de está forma se podría replicar el mensaje hasta que llegará a su destino. [22]

Las redes AD HOC también son conocidas como MANETS[16] y se pueden definir como "un conjunto de nodos móviles que se mueven a voluntad y se comunican con otros" [15]. La comunicación entre los nodos móviles se realiza por medio de enlaces inalámbricos, los cuales utilizan tarjetas inalámbricas para comunicarse entre sí.

Debido a que este tipo de red no tiene ruteadores fijos, todos los nodos funcionan como ruteadores los cuales descubren y mantienen las rutas a otros nodos de la red. En este tipo de red todos los nodos son capaces de moverse *y* comunicarse dinámicamente de una manera arbitraria [11]. En la figura 1.6 se muestra un ejemplo de MANETS.

[16] En adelante nos referiremos a las redes inalámbricas AD HOC por sus siglas en inglés MANETS (Mobile Ad Hoc Networks)

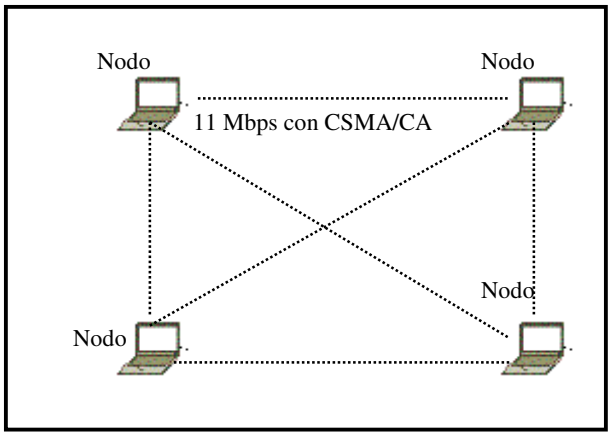


Figura 1.6. MANETS en donde cada nodo se comunica con otro sin la necesidad de un punto de acceso (AP).

En 802.11 el frame genérico para MANETS está conformado por tres campos de dirección. El primer campo identifica al receptor, el cuál es la dirección destino, el segundo campo es la dirección del emisor y el tercero es el BSSID[17] que es un identificador de frames de datos de 48 bits usado por todas los nodos en MANETS. (Ver figura 1.7)

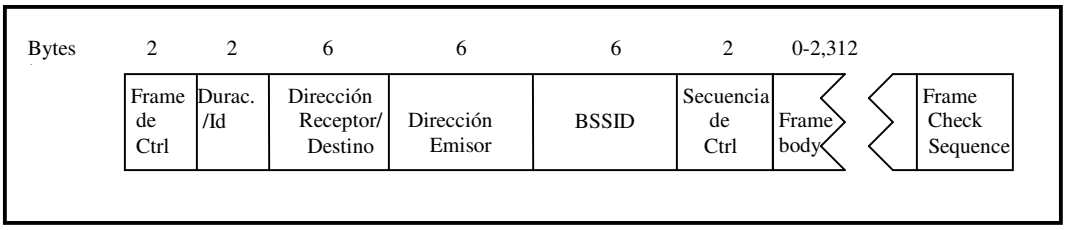


Figura 1.7. Frame genérico de 802.11 para MANETS.

Para crear una MANET con 802.11 en un ambienteWindows XP, se debe realizar:

1.- Instalar controladores de acuerdo a la tarjeta inalámbrica que se adquirió. (Ver figura 1.8)

Figura 1.8. Tarjeta inalambrica PCMCIA.

[17] Por sus siglas en inglés Basic Service Set Identifier.

2.- Una vez que se han instalado los controladores de la tarjeta, se debe abrir el menú: *configuración>conexiones de red.* (Ver figura 1.9)

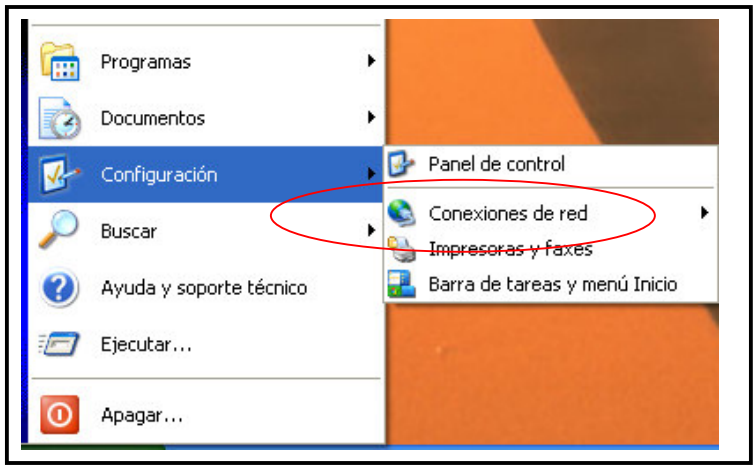


Figura 1.9. Menú conexiones de red.

3.- Cuando se selecciona el menú *conexiones de red*, se abre la siguiente pantalla (ver figura 1.10).

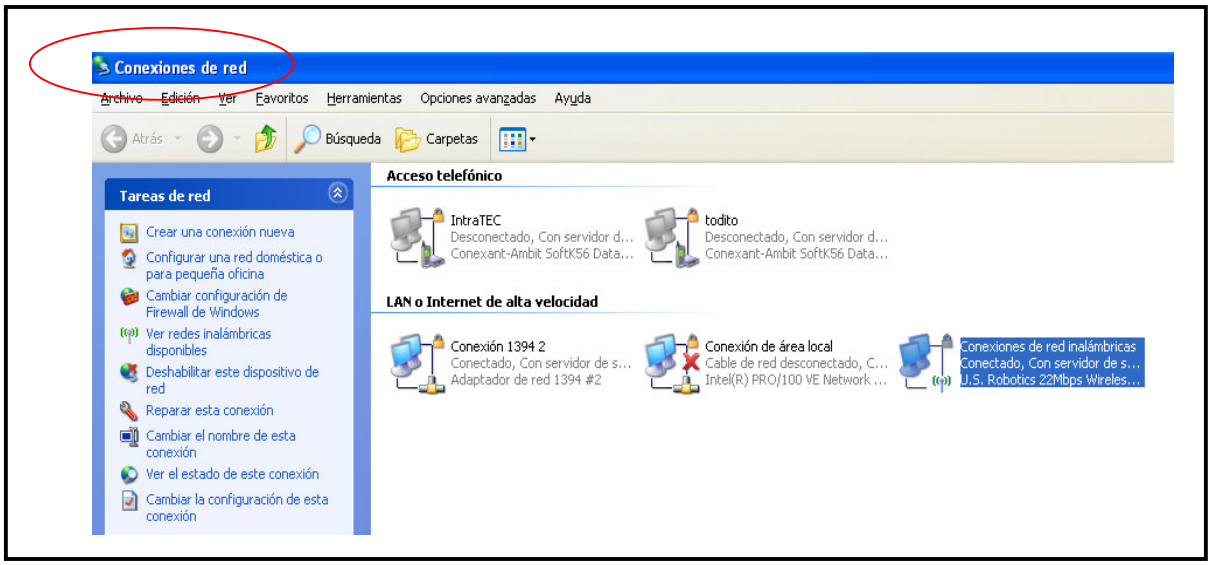


Figura 1.10. Icono conexiones de red inalámbricas.

4.- En la pantalla *conexiones de red*, se encuentra el icono de *conexiones de red inalámbricas*, el cuál se elige con el botón derecho y despliega el menú donde aparece la opción de *Propiedades*. (Ver figura 1.11)

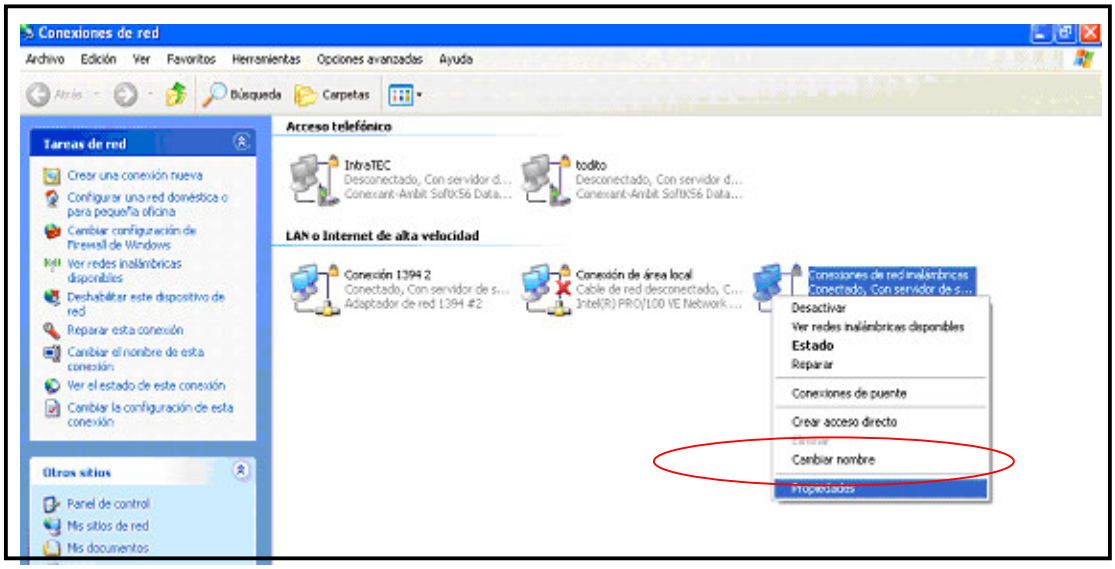


Figura 1.11. Menú de conexiones de red inalámbricas en donde se muestra la opción de Propiedades.

5.- Una vez que se ha seleccionado la opción *Propiedades*, se despliega la ventana *Propiedades de conexiones inalámbricas*, dentro de ésta, se selecciona la pestaña *Redes Inalámbricas*, y a continuación se elige el botón de Agregar (dentro de la misma ventana). (Ver figura 1.12)

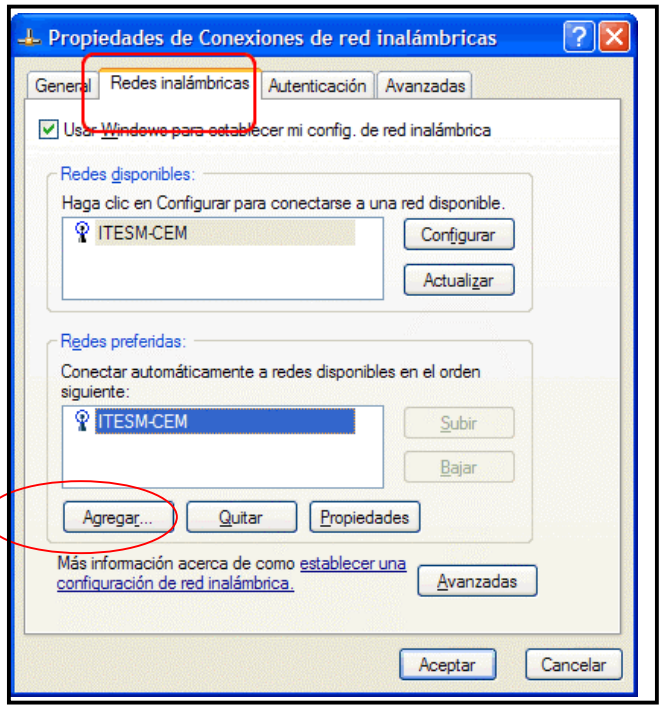


Figura 1.12. Ventana Propiedades de conexiones de red inalámbricas.

6.- Al elegir el botón Agregar se despliega la ventana de *Propiedades de red Inalámbrica*. Dentro de está, se debe escribir un nombre que identifica a la MANET (en este caso es prueba_cem). Posteriormente hay que seleccionar la casilla que se encuentra en la parte de debajo de la ventana, que dice *Esta es una red de equipo a equipo (ad hoc).* (Ver figura 1.13)

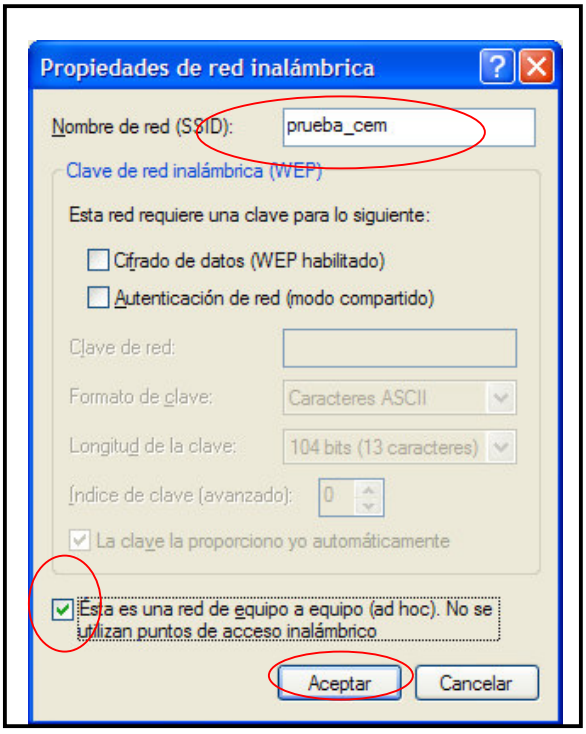


Figura 1.13. Ventana de configuración de una red AD HOC.

7.-Por último en la misma ventana, se selecciona y se da click en el botón aceptar para finalizar la configuración de la MANET. (Ver figura 1.13)

1.2.1 CARACTERÍSTICAS DE LAS MANETS

Una red inalámbrica AD HOC es un sistema autónomo que consiste en nodos móviles que no necesitan alguna infraestructura de red fija. La topología de las MANETS cambia de acuerdo a como se mueven los nodos en ésta y como ajustan sus características de transmisión o recepción. [13]

Las características más importantes de las MANETS son [13]:

- Topología dinámica. Los nodos se mueven de manera libre y arbitraria. Las condiciones del radio de propagación cambian de manera continua sobre el tiempo y la topología puede cambiar de forma arbitraria y rápida de manera impredecible.

- Exigencia de ancho de banda y capacidad de conexión variable. Las conexiones inalámbricas tienen una capacidad mucho menor que las conexiones cableadas. Debido a los efectos de acceso múltiple, ruido o señales de interferencia.

- Exigencia de energía de los nodos. Los nodos móviles requieren de baterías para su correcta operación. Como las MANETS contienen varios nodos, el agotamiento de las baterías en estos nodos tendrá gran influencia sobre todo el funcionamiento de la red. Es por eso que uno de los factores más importantes en el diseño de protocolos está

relacionado con la conservación de energía del dispositivo de red.

- Comunicación multisalto. Debido a las características de propagación de la señal en 802.11, las MANETS requieren de soporte de comunicaciones multisalto; es decir, los nodos móviles que no pueden alcanzar su destino directamente necesitan dejar sus mensajes a otros nodos para que estos los envíen al destino deseado.

- Seguridad limitada. Las redes inalámbricas móviles son generalmente más vulnerables a las violaciones de seguridad que las redes cableadas. La posibilidad de que la información sea leída de manera ilegal y la negación de servicio deben de considerarse cuando se diseñe la red inalámbrica. [13]

En general los nodos de las MANETS tienen capacidad limitada de procesamiento y memoria. Como resultado ciertos algoritmos que tienen un alto costo de procesamiento o de memoria, pueden no ser adecuados.

1.2.1.1 Características de las MANETS de acuerdo a la aplicación

Además de las características mencionadas anteriormente, existen otros aspectos que deben ser considerados y están en función de las aplicaciones que soportan las MANETS. A continuación se presentan las más importantes.

El primero es el origen de la red, existen dos formas de planear una MANET, que puede ser espontáneo o planeado. En MANETS espontáneas los nodos no tienen ninguna comunicación previa entre ellos y pueden formarse por corto tiempo, por ejemplo en un aeropuerto o una central de autobuses. En las MANETS planeadas los nodos tienen una relación prevista y que involucra un diseño de la infraestructura de éstas, ejemplo de MANETS planeadas son: los nodos que pertenecen a una institución educativa, empresa o alguna institución militar.

El segundo, es de acuerdo a su alcance. Se clasifican en local y distribuido. En MANETS locales, los nodos se encuentran dentro de un rango físico establecido, por ejemplo el salón de clases, el consultorio medico o la oficina. En MANETS distribuidas, los nodos se encuentran en un área ilimitada, sin que tengan la posibilidad de interactuar entre ellos. Para este tipo de redes se efectúan en frentes hostiles o de difícil acceso, por ejemplo un campo de guerra o un lugar donde no se pueda tener acceso por causas de desastres naturales.

El tercero es de acuerdo a la capacidad de los nodos, y los dos aspectos de clasificación son: uniforme y diverso. En MANETS que tengan capacidad uniforme, todos los nodos tienen las mismas capacidades de procesamiento, tamaño de memoria, disponibilidad de almacenamiento físico. Y en MANETS con capacidad diversa, los nodos difieren en forma significativa, ciertos nodos pueden tener alta capacidad de procesamiento mientras que otros se encuentra limitados, en capacidad de cómputo y/o de almacenamiento.

El cuarto es la transitoriedad de la red, que puede ser de corto o largo término. En MANETS de corto término, los nodos se reúnen por primera y única vez creando la red, cuando finalizan el propósito por el que fue creada, las MANETS se disuelven y no guardan ningún antecedente de

que alguna vez conformaron una MANET. En MANETS de largo término, los nodos son parte de alguna organización, conservan información sobre los nodos con los cuales forman MANETS para uso futuro. La vigencia de este tipo de redes es ilimitada y son comunes en organizaciones empresariales, educativas o militares.

1.2.2 APLICACIONES PARA MANETS

En un principio las características de las MANETS despertaron el interés de la milicia, debido a que se puede establecer comunicación de una manera rápida sin necesidad de infraestructura en ambientes hostiles. Sin embargo, como muchas otras tecnologías, ésta también ha pasado de ser exclusiva de los militares para pasar a los usuarios y organizaciones civiles.

Las agencias de rescate utilizan las MANETS para comunicarse en áreas donde no existe una cobertura de red, también pueden emplearse para seguridad nacional cuando se necesite establecer comunicación en crisis nacionales, debido a que la infraestructura existente no este funcionando, por causa de desastres naturales o conflictos armados.[13]

Actualmente las personas asisten a juntas y conferencias con sus computadoras portátiles y dispositivos inalámbricos. Por consiguiente, resulta atractivo tener información de la red en forma instantánea, además de compartir información sin la presencia de estaciones base fijas ni sistemas administradores. Es por eso que éste tipo de ambientes no tan hostiles representan un área de aplicación potencial de las MANETS.

Algunas aplicaciones para las MANETS pueden proveer las bases para la comercialización exitosa de distintos productos. De hecho, cualquier éxito comercial en la aplicación de una red puede ser considerada como un candidato para el desarrollo útil con nodos que pueden formar una red AD HOC. A continuación se presentan algunas de las aplicaciones comerciales más relevantes para MANETS. [10]

Redes Militares

Las primeras aplicaciones de MANETS se desarrollaron en el ámbito militar. Las MANETS permiten a las unidades de combate en un ambiente hostil, estar en cualquier momento comunicados sin la necesidad de tener una infraestructura física. Ejemplos de aplicaciones militares son Tactical Internet [10] y el concepto Saab NetDefence. [8]

Conferencias

Tal vez la aplicación prototipo requerida para el establecimiento de MANETS son las conferencias móviles. Cuando un usuario sale de su ambiente de oficina, la infraestructura de red de negocios se pierde. La necesidad de computación colaborativa puede ser, incluso más importante que en el ambiente diario de oficina. Dado que actualmente los proyectos están altamente computarizados, la necesidad de crear MANETS parece clara.

Redes caseras

La tendencia de los dispositivos de cómputo hacia lo portátil y movible es evidente. En los hogares cada vez es más común que se tengan computadoras portátiles, PDA's, celulares que tienen integrados dispositivos de hardware para conectarse a internet. El poder compartir la conexión a internet entre distintos nodos, es uno de los beneficios que se pueden obtener si se decide desarrollar en el hogar una red AD HOC.

Servicios de emergencia

Conforme el uso de internet va creciendo, la pérdida de la conectividad con la red durante desastres naturales puede representar un riesgo para millones de personas. Las aplicaciones que monitorean el clima, volcanes, tornados y tráfico vehicular entre otras, son importantes para el funcionamiento de los servicios de emergencia y es imprescindible que siempre estén disponibles, incluso cuando los elementos de la infraestructura de la red hayan sido deshabilitados como parte de los efectos de los desastres naturales. Las MANETS pueden ayudar a superar éstas continencias. Las unidades de emergencia pueden llevar consigo dispositivos de computo móviles con un dispositivo 802.11 que les permita crear MANETS y poder comunicarse entre si, durante momentos de pérdida de infraestructura de soporte. [10]

Por ejemplo, las patrullas de policía y las estaciones de bomberos pueden permanecer en contacto un mayor tiempo y proveer de información más rápido si ellos pudieran cooperar para formar una red AD HOC en lugares donde no hay conectividad a internet .

Redes de área personal (PAN)

El objetivo de este tipo de redes es crear una red formada por algunos nodos que estén asociados de manera muy cercana con una persona. Estos nodos pueden encontrarse en el cinturón de la persona o en una pulsera.

Visiones más exóticas incluyen dispositivos de realidad virtual ubicados alrededor de la cabeza y otros dispositivos más orientados al sentido del tacto. Estos dispositivos pueden o no necesitar estar conectados a Internet, pero es casi seguro que necesitarán comunicarse con otros mientras se asocian con las actividades del usuario.

Redes sensor

Las redes sensor son MANETS que para entablar comunicación entre nodos utilizan dispositivos llamados sensores.

Estos dispositivos que tienen un bajo precio de manufactura, pueden ofrecer información detallada acerca del terreno o condiciones del medio ambiente. Estos sensores pueden ser equipados con indicadores de posición, de manera alternativa, esta información de posición puede inferirse de la información de la red como es el número de saltos entre varios sensores. [9]

Por ejemplo, se puede presentar el caso de un derrame de químicos peligrosos, ya sea por alguna explosión o algún accidente. En lugar de enviar personal de emergencia, los cuales pueden exponerse a gases letales, sería mejor distribuir sensores que contengan tarjetas inalámbricas y puedan formar redes AD HOC. Los sensores podrían formar MANETS y cooperar entre sí, para recolectar la información deseada acerca del químico y su identificación.

Redes colaborativas

Este tipo de MANETS son muy intuitivas, un ejemplo de éstas es cuando un grupo de personas dentro de una junta necesitan compartir información entre sus computadoras portátiles o sus PDA's por lo que deben crear MANETS entre sus dispositivos y de está forma compartir información, cabe señalar que el crear la red AD HOC representaría un ahorro en tiempo en forma considerable que si se hiciera por medio de redes cableadas o con redes inalámbricas con infraestructura (AP).

Vehículos

Se considera la posibilidad de utilizar MANETS entre computadoras de automóvil y computadoras portátiles o PDAs que pueden acompañarnos durante un viaje en el automóvil. Las comunicaciones inalámbricas entre vehículos puede ser un sucesor lógico de los radios de banda civil.

1.2.3 PROTOCOLOS DE RUTEO EN MANETS

Para soportar la movilidad en MANETS un nodo debe ser capaz de comunicarse con los demás nodos, los cuales pueden o no estar dentro de su radio de transmisión. Las funciones de un protocolo de ruteo son: [22]

1. Determinar la topología de la red.

Un protocolo de ruteo debe determinar *y* monitorear el cambio de la topología en el tiempo. Debido a que las comunicaciones multisalto son necesarias en MANETS, los protocolos de ruteo deben asegurarse que las ligas entre las rutas tengan una fuerte conexión. Debe existir al menos una ruta de un nodo a cualquier otro en una red no particionada. Un nodo debe estar enterado de los nodos con los cuales puede comunicarse directamente. Debe tomarse en cuenta las dificultades relativas de formar ligas con esos nodos y los beneficios que proporcionarán esas conexiones al comunicar a toda la red. (por ejemplo, el uso de ancho de banda, el retraso en la transmisión, el consumo de energía). Existen 2 esquemas para proveer conexión en una red AD HOC: arquitectura de red plana y arquitectura de red jerárquica. En la arquitectura de red plana, todos los nodos son iguales y el ruteo de paquetes está basado en conexiones punto a punto. Por otro lado, en la arquitectura de red jerárquica, al menos un nodo en cada capa inferior se designa para servir como medio de comunicación o coordinador de las capas superiores.

2. Mantener la conectividad de la red durante los cambios en las condiciones de radio y movilidad.

Debido a que la ubicación de cada nodo puede cambiar en cualquier momento, la topología de la red cambia frecuentemente. Los cambios topológicos pueden ocurrir por:

• La ruptura de un nodo a causa de un ambiente hostil.

• La pérdida de la conexión debido a una señal de interferencia y a cambios en las condiciones de la propagación de la señal.

Por consiguiente, un protocolo de ruteo para MANETS, debe ser capaz de actualizar en forma dinámica el status de sus conexiones y reconfigurarse a si mismo para mantener una conectividad fuerte para poder soportar la comunicación entre los nodos.

3. Programación de transmisión y asignación de canal.

Debido a que una nueva transmisión de radio realizada por un nodo puede afectar una comunicación existente provocando interferencia, es necesario una programación de envío de paquetes y un algoritmo de asignación de canales para asegurarse de que la nueva señal no interfiera con otra.

4. Ruteo de Paquetes.

A diferencia de las redes alámbricas con nodos estáticos, las MANETS requieren de un esquema de ruteo altamente adaptable para hacer frente a los constantes cambios de topología. Esto implica que los protocolos de ruteo deben propagar los cambios de topología y recalcular las rutas al destino.

1.2.3.1 Tipos de protocolo de ruteo para MANETS

Los protocolos de ruteo se clasifican conforme a: [22]

1. Protocolos manejados por tablas o proactivos.
2. Protocolos sobre demanda o reactivos.
3. Protocolos híbridos.

Los protocolos proactivos, son los que guardan en una tabla, las rutas a cada uno de los nodos de la red. Una ventaja de este tipo de protocolo es que no se requiere esperar a que se descubra una ruta pues ya la tiene almacenada, una desventaja es que muchas veces contiene información de rutas que ya no están disponibles.

Los protocolos reactivos son los que buscan una ruta en el momento en que se necesita. Una de las ventajas de este protocolo es que no ocupa espacio con rutas que no se utilizan, pero tienen un retardo cuando se envía el primer paquete debido a que se tiene que descubrir la ruta antes de enviarlo.

A continuación se proporciona una pequeña descripción de cada protocolo. [22]

Destination-Secuence Distance-Vector (DSDV). Este protocolo se basa en un algoritmo proactivo que se basa en el mecanismo de ruteo clásico Bellman-Ford. A diferencia del mecanismo de ruteo clásico este protocolo no incluye enlaces en las tablas de ruteo.

El protocolo de ruteo de *Switch Gateway Clusterhead* (*CGSR*) difiere del protocolo DSDV en el tipo de direccionamiento y en el esquema de organización empleado en la red. En vez de ser una red plana, CGSR es una red inalámbrica móvil de multisalto con varios esquemas de ruteo heurísticos.

Wireless Routing Protocol (WRP). Tiene el propósito de mantener la información de ruteo en los nodos de la red. Cada nodo de la red es responsable de mantener 4 tablas: tabla de distancia, tabla de ruteo, tabla de costo de ruta y lista de retransmisión de mensajes.

Ad Hoc On-Demand Distance-Vector (AODV). El protocolo de ruteo vector distancia sobre demanda, está construido sobre el algoritmo utilizado en DSVD. AODV es una mejora de DSDV debido a que minimiza el número de publicaciones creando las rutas sobre demanda.

Dynamic Source Routing (DSR). El protocolo de ruteo dinámico de fuente, es un protocolo reactivo que se basa en el concepto de ruteo desde el origen. Los nodos móviles deben mantener rutas en cache que contengan el origen de las rutas.

Temporally Ordered Routing Protocol (TORA). El algoritmo de ruteo temporalmente ordenado es un algoritmo de ruteo distribuido sin lazos y altamente adaptable, que se basa en el concepto de conexiones reversivas.

Associativity-Based Routing (ABR). El protocolo de ruteo basado en asociatividad, es totalmente diferente debido a que no tiene lazos, paquetes duplicados y define una nueva métrica para MANETS llamada estabilidad de asociación.

Zone Routing Protocol (ZRP). El protocolo de ruteo de zona combina el protocolo proactivo de manera local y el protocolo reactivo de manera global. Es decir, este protocolo divide la MANET en zonas, en donde todos los nodos que estén a una distancia específica, forman parte de esa zona. Cada nodo tiene su propia zona y utiliza un protocolo proactivo para encontrar las rutas a los nodos dentro de su zona y utiliza un protocolo reactivo para encontrar una ruta hacia algún nodo que esté fuera de su zona.

La figura 1.14 muestra una clasificación de los protocolos de ruteo para MANETS de acuerdo a su tipo.

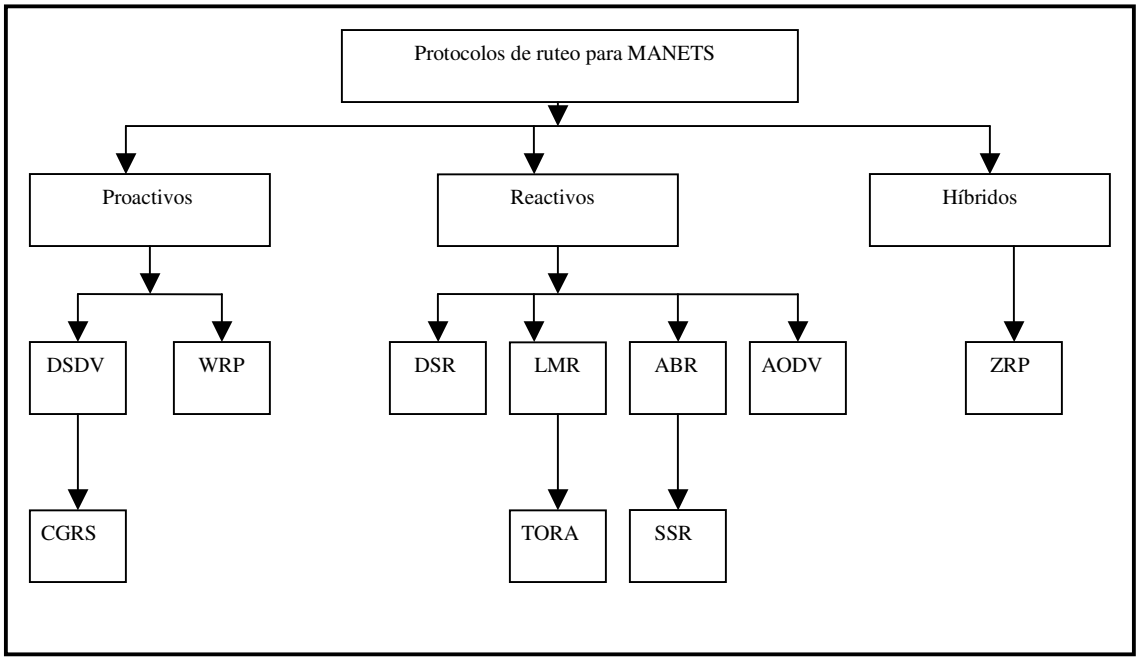


Figura 1.14. Clasificación de los protocolos de ruteo para MANETS.

1.3 INTRODUCCIÓN A LA CRIPTOGRAFÍA

La distribución de mensajes en las MANETS, puede representar un problema para la seguridad de la red debido a que un adversario puede "husmear" o monitorear el medio y corromper la confidencialidad de la red AD HOC. Por ello recobra importancia el uso de técnicas criptográficas para evitar que el adversario pueda acceder a información sensitiva de la organización.

El objetivo de la criptografía es solucionar problemas que involucren secrecía, autenticación e integridad. La idea de la criptografía es encriptar un mensaje en diferentes formas, para que la interpretación de este sea incomprensible para todos aquellos que no tengan la llave para desencriptar el mensaje. [1]

1.3.1 ENCRIPCIÓN SIMÉTRICA

Existen dos tipos de algoritmos para encriptar información: simétricos y de llave pública. Los algoritmos simétricos, algunas veces son llamados algoritmos convencionales. En el caso de los algoritmos simétricos, la mayoría de las llaves de encripción/decripción son las mismas y requieren que el emisor así como el receptor establezcan algún acuerdo común para intercambiar sus llaves, antes de que puedan entablar una comunicación segura. La seguridad, como en la mayoría de los algoritmos, reside en la llave. Dar a conocer la llave significa que cualquier entidad no autorizada puede llegar a encriptar y decriptar el mensaje. De acuerdo a [1], la encripción y decripción se denota como:

$$C=E_k(M)$$

$$M=D_k(C)$$

Los algoritmos simétricos pueden ser divididos en dos categorías:

- Encripción en flujo, las operaciones para encriptar se realizan con un bit a la vez, sobre información que fluye en un canal de comunicación determinado.

- Encripción en bloque, el texto claro es separado en diferentes bloques de datos y cada uno es encriptado independientemente de otros bloques.

El esquema de encripción simétrica puede ser empleado para otorgar confidencialidad e integridad. La confidencialidad y la integridad se logran si se protegen las llaves en el cifrado y en el descifrado. Es decir, se obtienen simultáneamente si se protege la llave secreta. Para ello, la llave secreta compartida, debe distribuirse por un canal seguro de comunicación. Para autenticación, solo se puede autenticar el mensaje pero no se puede autenticar al emisor. [1]

En la figura 1.15 se muestra un diagrama del proceso de encripción simétrica. El mensaje en texto plano es denotado como *m* y es encriptado usando la llave compartida *k*, dando como resultado el texto cifrado *c*. Para recuperar el mensaje encriptado, el texto cifrado es decriptado utilizando la misma llave que se usó para encriptar.

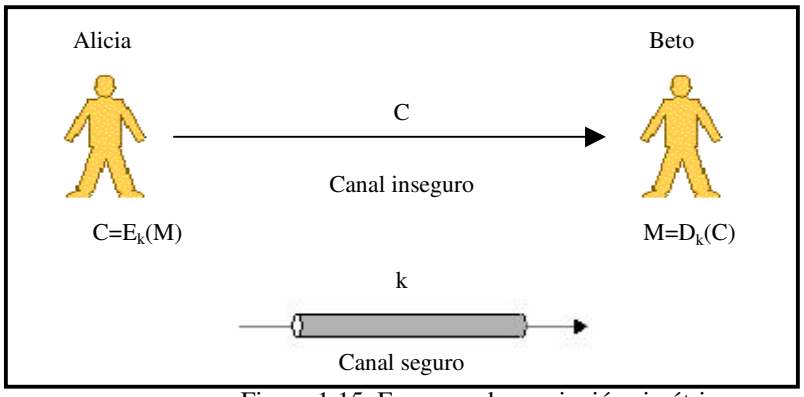


Figura 1.15. Esquema de encripción simétrica.

Algunos de los algoritmos de llave simétrica más populares son: *DES, IDEA, y AES.*

El *Data Encryption Standard* (DES) fue creado en 1974 por IBM y propuesto a raíz de una petición de la NIST (*National Institute of Standards and Technology*, USA) en 1972 y 1974. Está inspirado en el sistema LUCIFER de IBM. Fue aprobado y modificado por la *Nacional Security Agency* (NSA) que impuso la longitud de la llave. El algoritmo de cifrado que emplean es en bloque y simétrico. La longitud del bloque es de 64 bits y la longitud de la llave es de 56 bits, por lo que existen: 2^{56} =7.2 x 10^{16} llaves diferentes. La norma exige que DES se implemente mediante un circuito integrado. En 1981 ANSI adopto el DES con el nombre de Data Encryption Algorithm que no exige chip y puede ser programado. [2]

El *International Data Encryption* (IDEA) fue desarrollado por Xuejia Lai y James L. Massey de ETH Zuria. Encripta bloques de 64 bits de texto claro en bloques de 64 bits usando una llave de 128 bits. IDEA se caracteriza por su parecido a DES en que ambos operan en iteraciones y cuentan con una función rara que no necesita ser reversible para decriptar, dicha función es

ejecutada en el mismo sentido tanto para encriptar como para decriptar. IDEA y DES presentan la propiedad en común de que son idénticos en su encripción y decripción excepto por la llave de expansión. Utiliza tres operaciones para mapear, que son de 16 bits a 16 bits que son fáciles de realizar en software. Las operaciones son: una operación de *XOR* exclusivo ($\oplus$), una operación de suma modificada (+) y una operación modificada de multiplación ($\otimes$). [2]

En 1997 la entidad norteamericana *National Institute of Standards and Technology* (NIST) anuncia el substituto de DES: *Advanced Encryption Standard* (AES) que fue desarrollado por *J.Daemen* y *V.Rijmen*. El algoritmo propuesto por sus creadores tenía el nombre de *Rijndael.* Dicho algoritmo tiene una iteración de bloque cifrado, con un tamaño de bloque y llave variable. La llave puede tener un tamaño de 128, 192 o 256 bits y no usa otros componentes criptográficos. No tiene partes obscuras ni cosas difíciles de entender entre operaciones aritméticas. No deja espacio suficiente para esconder una puerta trasera. El modo encripción que utiliza es en bloque (ECB). [2]

1.3.2 ENCRIPCIÓN DE LLAVE PÚBLICA

En 1976 *Whitfield Diffie* y *Martin Hellman* cambiaron el paradigma de la criptografía al desarrollar el concepto de criptografía de llave pública. Los algoritmos de llave pública, también conocidos como asimétricos, utilizan diferentes llaves para encriptar y para desencriptar. La llave para desencriptar no puede ser derivada de la llave de encripción.

Los algoritmos son llamados de llave pública porque una de las llaves es conocida por todo el mundo; mientras que la otra llave, solo la conoce la persona que la generó. En la mayor parte de los casos, la llave de encripción se conoce como llave pública y la llave para desencriptar como llave privada. La llave pública que es empleada para encriptar se denota por [1]:

$$C=E_k(M)$$

Cabe señalar que a pesar de que son diferentes las llaves privadas y públicas, la llave privada para desencriptar el mensaje se denota como:

$$M=D_k(C)$$

En la figura 1.16 se ejemplifica el proceso que realizan dos usuarios, para intercambiar un mensaje en forma segura en un canal inseguro. Beto genera su par de llaves pública y privada pk_{Beto}/sk_{Beto} y las dio a conocer al dominio público. Si Alicia quiere enviar un mensaje encriptado a Beto, lo primero que tiene que hacer es obtener la llave pública de Beto del dominio público y asegurarse que la llave en realidad le pertenece a él, es decir autenticar la llave.

Una vez que Alicia ha autenticado la llave pública de Beto puede utilizar la llave pública pk_{Beto}, encriptar el mensaje m y obtener el texto cifrado c que solo puede ser desencriptado por Beto con su llave privada sk_{Beto} que solo él conoce.

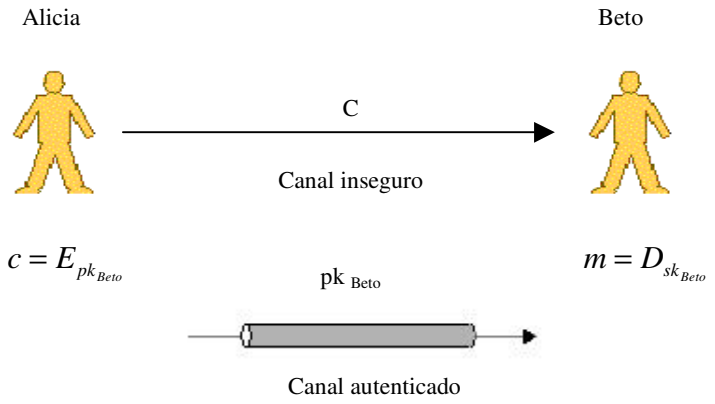


Figura 1.16. Esquema de encripción de llave pública.

Comparado con la encripción simétrica, la encripción de llave pública tiene menores requerimientos en cuanto al canal de comunicación sobre el cuál se distribuyen las llaves. La encripción de llave pública requiere un canal autenticado y en el esquema de encripción simétrica es necesario un canal seguro de comunicación para distribuir las llaves simétricas. La encripción de llave pública proporciona mecanismos de seguridad como no-repudio, confidencialidad, integridad y autenticación, sin embargo requiere más recursos computacionales que la encripción simétrica y en consecuencia el procesamiento es más lento. Es por eso que la encripción de llave pública es usada para encriptar cantidades no muy grandes de datos, por ejemplo: encripción de llaves simétricas y firmas digitales.

Desde 1976, fecha en que se publico la propuesta de *Diffie-Hellman*, han surgido numerosos algoritmos de criptografía de llave pública dentro de los cuales destacan: *RSA, ElGamal* y *Rabin.* En la práctica estos algoritmos han sido empleados para la distribución de llaves.

1.3.2.1 Diffie-Hellman (DH)

En 1976, los ingenieros *Whitfield Diffie* y *Martin Hellman* de la Universidad de Stanford, sugirieron usar problemas computacionalmente irresolubles para el diseño de criptosistemas seguros.

La idea que propusieron, consiste en encontrar un sistema de cifrado computacionalmente fácil, de tal forma que el descifrado sea computacionalmente irresoluble, a menos que se conozca la llave correspondiente. Para ello, se define una transformación criptográfica T_k de fácil aplicación, pero que sea muy difícil encontrar la transformación inversa T_k^{-1} sin la llave de decripción. [39]

El algoritmo que propusieron es: [2]

1. Alicia y Beto seleccionan públicamente un grupo multiplicativo finito, G, de orden n y un elemento de G.

2. Alicia genera un número aleatorio X_{Alicia}, calcula Y_{Alicia} en G y transmite este elemento a Beto.

3. Beto genera un número aleatorio X_{Beto}, calcula Y_{Beto} en *G* y transmite este elemento a Alicia.
4. Alicia recibe Y_{Beto} y calcula $(Y_{Beto})^{X_{Alicia}}$ en *G*.

5. Beto recibe Y_{Alicia} y calcula $(Y_{Alicia})^{X_{Beto}}$ $(Y_a)^{Xb}$ en *G*.

El algoritmo se ejemplifica en la figura 1.17.

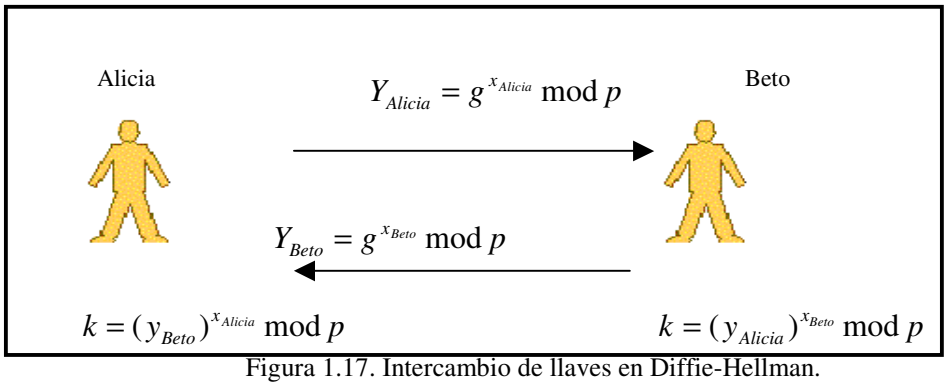


Figura 1.17. Intercambio de llaves en Diffie-Hellman.

La seguridad del algoritmo radica en la complejidad de calcular logaritmos discretos en un campo finito [1,2].

Las principales desventajas que se observan en este algoritmo son: el proceso de encripción/decripción es más lento comparado con un algoritmo de llave simétrica, el tamaño de las llaves es más grande que las que se emplean en los algoritmos de llave simétrica y que es factible que sufra un ataque de "hombre en medio" que intercepte el proceso de intercambio de llaves y que se haga pasar por un usuario valido.

1.3.2.2 RSA

Se considera el algoritmo de llave pública más popular debido a su facilidad para comprenderse e implementarse, fue desarrollado en 1977 por *Rivest*, *Shamir* y *Adleman* y se publicó en 1978. La fortaleza de éste algoritmo radica en la dificultad de factorizar números "grandes" y se basa en una función unidireccional con puerta trasera *(TOF)*[18] que emplea números primos. Se caracteriza porque el texto claro es encriptado en bloques que tienen un valor binario menor a un número *n* y su tamaño debe ser igual o menor a $log_2(n)$. [1,2]

El algoritmo de RSA puede dividirse en dos etapas: La de creación de las llaves y la encripción/decripción del mensaje.

La etapa de creación de mensajes consiste en los siguientes pasos:

[18] Por sus siglas en inglés *Trapdoor One-way Function*

1. Cada usuario elige un número $n = p * q$ (que pueden ser distintos).
2. Los valores p y q no se hacen públicos.
3. Cada usuario calcula $\phi(n) = (p-1)(q-1)$.
4. Cada usuario elige una llave pública e ($e < n$) y que cumpla: $mcd[e, \phi(n)] = 1$.
5. Cada usuario calcula la llave privada que cumpla: $d = inv[e, \phi(n)]$.
6. Se hace público el número n y la llave e.
7. Se guarda en secreto la llave d.

En la etapa de encripción/decripción, si se desea encriptar un mensaje se tiene que cumplir que $M<n$ y con la llave pública (e,n) se realiza:

$$C=M^e \bmod n$$

Para decriptar el criptograma C es necesario usar la llave privada (d,n) conforme a:

$$M=C^d \bmod n$$

La seguridad de éste algoritmo radica en la dificultad de factorizar números "grandes"(de 100 a 200 dígitos), las llaves privada/pública son funciones de un par de número primos "grandes". [1]

Los ataques a los que puede ser susceptible este algoritmo son:

- Fuerza Bruta. Para realizar este ataque es necesario probar todas las llaves posibles.
- Ataques matemáticos.
  - Para este ataque se pueden factorizar n en dos números primos y calcular: $\Phi(n) = (p-1)(q-1)$ y $d = e^{-1} \bmod \Phi(n)$.
  - Determinar directamente $\Phi(n)$, para ello se debe conocer $d = e^{-1} \bmod \Phi(n)$.
  - Deducir d directamente.
- Ataques de tiempo. El adversario compara tiempos de decripción.

1.3.3 HUELLA Y FIRMA DIGITAL

Las firmas manuscritas han sido usadas desde hace muchos años como prueba de alguna acción. Las principales características de una firma manuscrita son [2]:

- La firma es auténtica, el firmante deliberadamente firmo el documento.

- La firma es inolvidable, es prueba de que el firmante y no otra persona, deliberadamente firmo el documento.

- La firma no es reutilizable.

- La firma es parte del documento y ninguna persona puede moverlo a otro documento.

- El documento firmado es inalterable y después de que el documento fue firmado, no puede ser alterado.

- La firma no puede ser repudiada, el firmante no puede argumentar que el o ella no firmaron.

En seguridad informática, se han realizado distintos esfuerzos para llevar la mayoría de las características de una firma manuscrita a una huella y/o firma digital. Con el objeto de brindar integridad, confidencialidad y no repudio a un objeto (que puede ser un documento) determinado.

Una huella digital se define como "La salida producida por una función hash aplicada a un documento, es conocida con el nombre de huella digital de dicho documento". Cualquier cambio en el documento produce una huella diferente. [2]

Para garantizar que cualquier cambio en un documento produzca un "efecto avalancha" se utilizan funciones hash. El propósito de emplear este tipo de funciones es asegurar que, si la información es cambiada (aún en sólo un bit) un valor completamente diferente es producido. Ejemplos de estas funciones son: *MD5, SHA-1 y RIPEMD-160*.

MD5 toma como entrada un mensaje de longitud arbitraria, que puede tener cualquier longitud, y regresa como salida una huella digital de 128 bits del mensaje (también conocido como: *message-digest*). En la práctica es muy difícil obtener dos mensajes que produzcan la misma huella digital. Los pasos que sigue el algoritmo *MD5* son: [2]

1. Agregar bits de relleno (*Padding*).

2. Agregar longitud.

3. Inicializar buffer del *MD.*

4. Procesar el mensaje en bloques de 16 palabras.

5. Compendio del mensaje.

Otra función hash que puede ser empleada es *SHA-1.* Está fue desarrollada por el *National Institute of Standards and Technology* (NIST). Cuando se ejecuta está función lo primero que realiza es tomar un mensaje de entrada que se procesa en bloques de 512 bits, con una longitud máxima de 2^{64} bits y produce una salida de 160 bits. Los pasos que sigue SHA-1 son: [2]

1. Añadir bits de relleno (padding bits).

2. Añadir la longitud.

3. Inicializar el buffer de 160 bits *MD.*

4. Procesar el mensaje en bloques de 512 palabras.

5. Imprimir la salida.

Si se compara *MD5* y *SHA-1* en cuanto a la seguridad en contra de ataques fuerza bruta. Tenemos que tomar en cuenta que la huella *SHA-1* es 32 bits más grande que la de *MD5*. Para producir dos mensajes con la misma firma es de 2^{65} para *MD5* y 2^{80} para *SHA-1*. La dificultad para producir cualquier mensaje teniendo una firma de mensaje es del orden de 2^{128} operaciones para *MD5* y de 2^{160} para *SHA-1*.

Con respecto a sus fortalezas contra un criptoanálisis, MD5 es vulnerable a ataques de criptoanálisis desde su diseño. SHA-1 no es suceptible a tales ataques. Sin embargo se conoce muy poco acerca de los criterios de diseño de SHA-1, por lo que es más difícil de juzgar que MD5.

La función *RIPEMD-160* fue desarrollada en Europa como parte del proyecto PIPE en 1996 por investigadores involucrados en ataques a MD4/5. Se puede decir que es similar a *MD5* y a *SHA-1*. Utiliza 2 líneas paralelas de 5 iteraciones de 16 pasos y crea un valor hash de 160 bits. Se considera que es más lento pero más seguro que SHA-1. El procedimiento de ejecución de este algoritmo es:

1. Rellenar mensaje para que longitud sea 448 mod 512.

2. Añadir 64 bits de longitud del mensaje al final.

3. Inicializar un buffer de 5 palabras (160 bits). Por ejemplo: A = 67452301 D = 10325476 B = efcdab89 E = c3d2e1f0 y C = 98badcfe.

4. Procesar mensajes en bloques de 16 palabras (512 bits) usa 10 iteraciones de 16 operaciones de bits en el bloque de mensaje y el buffer.

5. Valor de salida hash: valor que queda al final del buffer.

En agosto de 1991, el *NIST*[19] propuso el uso *DSA*[20] para firma digital.[1] El objetivo de una firma digital es permitir al receptor verificar que la información esta intacta (integridad), la autenticidad del origen de la información y la no-repudiación.[2] Podemos decir que tiene el mismo propósito que la firma digital y tiene la ventaja de que no puede ser falsificada tan fácil como la firma escrita.

La figura 1.18 ilustra el uso de la firma digital. Alicia quiere enviar un mensaje a Beto pero no quiere que éste sea modificado durante el proceso de transmisión, por su parte Beto quiere asegurarse que el mensaje realmente es de ella. Alicia introduce el mensaje en una función hash,

[19] Por sus siglas en inglés *Nacional Institute of Standars and Technology*
[20] Ídem *Digital Signature Standard*

el resultado de la función hash lo encripta con su llave privada sk_{Alicia} y lo envía junto con el mensaje. Beto recibe el mensaje y lo introduce en la función hash, además verifica la firma del mensaje usando la llave pública pk_{Alicia} de Alicia. Si el resultado de las dos funciones es el mismo entonces el mensaje no ha sido modificado y está firmado por Alicia.

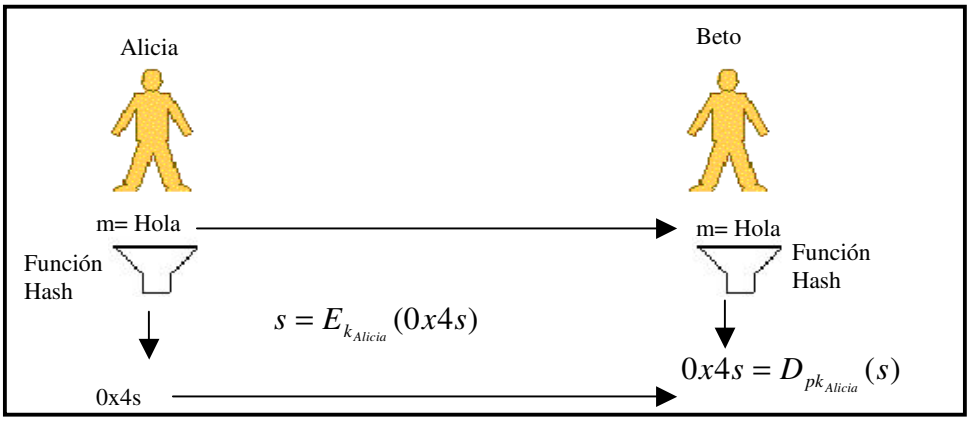


Figura 1.18. Ejemplo de firma digital.

1.3.4 CERTIFICADO DIGITAL

Un certificado digital asienta que una llave pública y su correspondiente llave privada pertenecen a un individuo en particular, certificando de esta manera la identidad de dicho individuo, ayudando a simplificar la tarea de verificar que la llave pertenece a la persona deseada. También son útiles para identificar a una entidad o individuo cuando manda mensajes a otras entidades en la red, es decir sirven como prueba de que un individuo es quien dice ser. [2]

Los certificados son expedidos por autoridades confiables conocidas como Autoridad Certificadora (CA)[21]. Estas entidades son responsables de certificar la identidad de un individuo y la posesión de su llave pública. [2]

Los certificados están conformados por:

1. Nombre del usuario y otra información como su e-mail.
2. Llave pública del usuario.
3. Nombre del emisor (CA).
4. Número de serie.
5. Período de validez.

[21] En adelante nos referiremos a las Autoridades Certificadoras por sus siglas en inglés CA

Para lograr la interoperabilidad entre sistemas de distintos fabricantes se definió el estándar público *X.509* por la *ISO*, que gobierna el formato y el contenido de los certificados digitales. Algunas implementaciones del formato se encuentran en el protocolo *Secure Sockets Layer* (SSL), en el protocolo de pago electronico SET y en el protocolo de encripción de correo S/MIME.

La figura 1.19 muestra la estructura de un certificado con el estándar *X.509*. El campo versión identifica el formato del certificado, el número de serie es un identificador único en la CA. El siguiente campo identifica el algoritmo que es usado para firmar el certificado junto con los parámetros necesarios, el campo enviante es el nombre de la CA. El campo de período de validez es el tiempo de vida del certificado, el campo asunto es el nombre del usuario y el campo información de llave pública contiene el nombre del algoritmo, los parámetros necesarios y la llave pública. El último campo es la firma de la CA. [1]

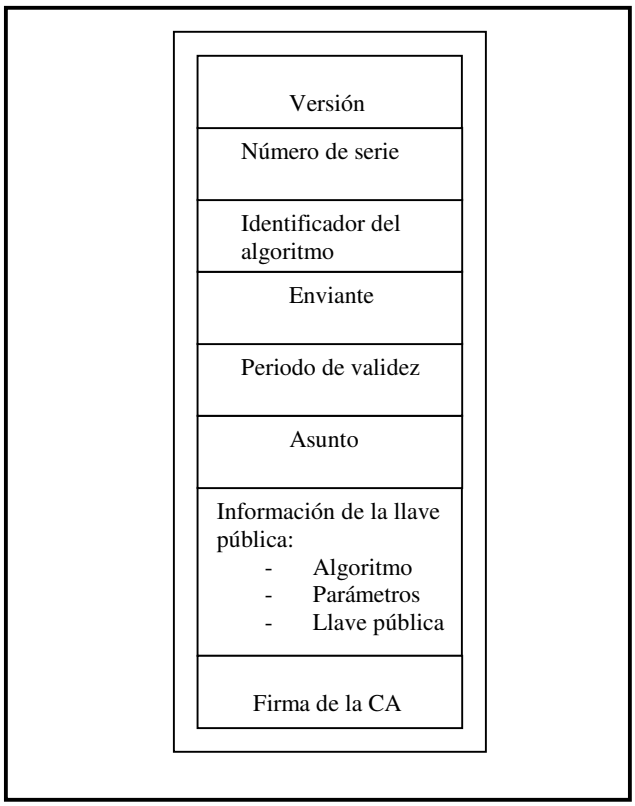


Figura 1.19. Ejemplo de firma digital.

1.3.5 SECRETOS COMPARTIDOS

Los esquemas umbral de secretos compartidos surgen por la necesidad de inventar una técnica que no dependa de una llave para proteger las llaves criptográficas. El esquema umbral de secretos compartidos relaciona protocolos para establecer una llave.

La idea de secreto compartido inicia con un secreto que se denota como *s,* se divide en *n* piezas llamadas compartidas, las cuales son distribuidas a todos los usuarios de tal forma que ningún usuario puede reconstruir el secreto hasta que no se encuentren todas *n* de *s* piezas juntas. [3]

La Teoría de Secretos Compartidos ha sido ampliamente estudiada por la criptografía y por la computación distribuida. Existen tres enfoques de ésta que son la de *Adi Shamir*[22], la de *Blakley*[23] y la de *Michael Rabin*[24].

El enfoque de *Blakley* se basa en geometría e intersección de planos. No es considerado perfecto ya que el tener una de las k partes permite conocer que el secreto es un punto dentro del hiperplano.

Por su parte *Michael Rabin p*ropone la utilización del *Algoritmo de Dispersión de Información* en donde:

1. Dado un archivo F longitud $L=|F|$ es dividido en n partes F_i.

2. Cada una de las partes tiene una longitud de $|F_i| = L/m$, de tal forma que con solo tener m partes se puede recuperar F.

Y el enfoque de *Shamir* se basa en la interpolación polinomial. Dados k puntos dentro de un espacio bidimensional, representados por (x_i,y_i), ..., (x_k,y_k) con x's distintas, solo existe una y solo una representación polinomial $q(x)$ de grado $k-1$ tal que $q(x_i) = y_i$ para toda i. Se asume que la información, o el secreto original, que va a compartirse esta representado por un número D. Para dividir éste en D_i partes escogemos un polinomio de grado $k-1$: [2]
$q(x) = a_0 + a_1x + \ldots + a_{k-1}x^{k-1}$, en donde $a_0 = D$ y se evalua: [2]

$D_1 = q(1)$
:
$D_i = q(i)$
:
$D_n = q(n)$

Dado cualquier subconjuto de k partes de estos valores D_i, se encuentran los coeficientes de $q(x)$ por interpolación y se evalua $D=q(0)$ y con ello se puede recuperar el secreto compartido. Cabe señalar que el conocer solo $k-1$ partes no permite calcular D. [2]

1.3.6 ADMINISTRACIÓN DE LLAVES

La mayoría de los mecanismos que son usados para proporcionar los servicios de seguridad, requieren el uso de algún tipo de llave criptográfica que necesita ser compartida entre las entidades que deseen comunicarse en forma segura.

---

[22] Shamir A., How to Share a Secret, Communications of the ACM, 22,1979, pp. 612--613

[23] Blakley, G.R. Safeguarding Cryptographic Keys. Proc. AFIPS 1979NCC, Vol. 48, Arlington, Va., June 1979, pp. 313-317

[24] M.O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing and Fault Tolerance", ACM Journal of the Association for Computing Machinery, Vol.36 No 2, April 1989, pp 335-348.

Se puede definir el concepto de administración de llaves como "El conjunto de procesos y mecanismos, los cuales soportan el establecimiento y el mantenimiento de llaves entre las partes, incluyendo el servicio de remplazar llaves con nuevas llaves cuando se requiera". [3]

La necesidad de administrar las llaves en un sistema criptográfico, es fundamental para asegurar que la comunicación entre las partes, se realice en forma confiable. De acuerdo a [3] el propósito que persigue la administración de llaves es:

1. Inicializar a los usuarios del sistema dentro de un dominio.

2. Generar, distribuir e instalar las llaves.

3. Controlar el uso de las llaves.

4. Actualizar, revocar y destruir las llaves.

5. Resguardar, respaldar y recuperar las llaves almacenadas.

La administración de llaves está expuesta a que un atacante:

- Comprometa la confidencialidad de las llaves y/o la autenticidad de las llaves y/o el generador de llaves.

- A que utilice llaves en forma indebida.

En la práctica la seguridad de un algoritmo criptográfico reside en la llave por lo que la administración de las llaves adquiere mayor importancia. Existen dos métodos distintos de manejar la administración de llaves, el primero está basado en técnicas de llave simétrica y el segundo en llave pública.

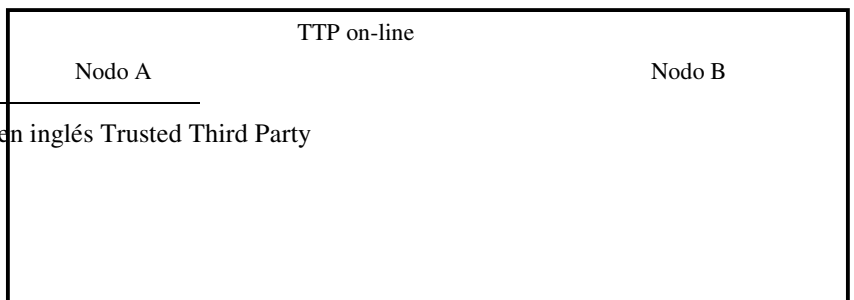
1.3.6.1 Administración de llaves a través de técnicas de llave simétrica

Se denomina *TTP*[25] a la entidad en la red la cual es confiable para todos los usuarios del sistema y es usada para proporcionar el servicio de administración de llaves. Pueden ser clasificados como: en línea o fuera de línea. [3]

En la clasificación en línea la *TTP* participa en forma activa pero solo con fines administrativos, la comunicación entre los usuarios es directa.

En la clasificación fuera de línea la *TTP* se comunica con los usuarios antes de que ellos establezcan una línea de comunicación. [3]

La figura 1.20 ilustra las diferentes categorías.



[25] Por sus siglas en inglés Trusted Third Party

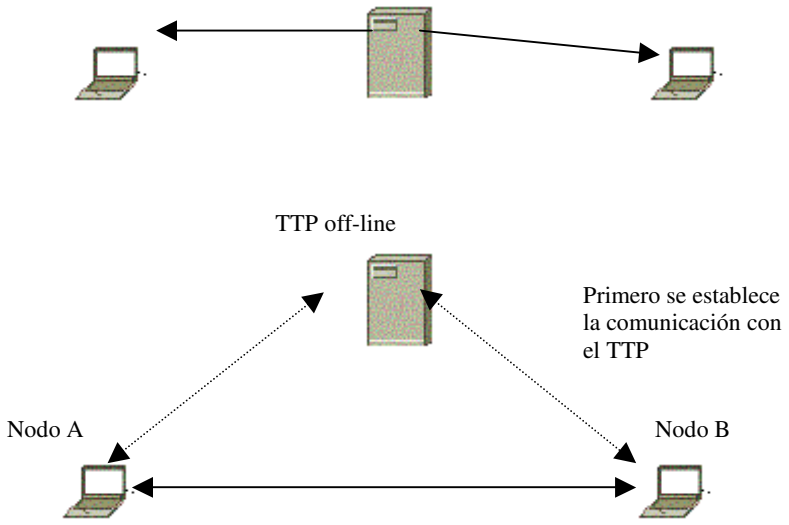


Figura 1.20. Categorías de la TTP.

Ejemplos de terceras entidades confiables son los centros de distribución de llaves *(KDC)*[26] , los centros de interpretación de llaves *(KTC)*[27] y las CA.

De acuerdo a [3] la diferencia entre un KDC y un KTC, radica en que el primero es usado solo para distribuir llaves entre usuarios, y el segundo además de distribuir llaves, también puede distribuir llaves de sesión.

Los *KDC* y los *KTC* son administradores de llaves simétricas y la CA es un administrador de llaves públicas. Los *KDC* y *KTC* son empleados para simplificar la administración de llaves, lo único que requiere un usuario para compartir su llave secreta es compartirla con la *TTP,* en lugar de compartirla con cada usuario individualmente, con lo cual la administración de llaves se simplifica de *n(n-1)/2* a *n* donde *n* es el número de usuarios.

Una aplicación de un KDC se puede ver en *Kerberos* que fue desarrollado por el equipo de desarrollo del MIT en el proyecto *Athena*. [1]

Kerberos es un protocolo de autenticación diseñado para redes, que utilicen el protocolo TCP/IP. Un servicio de *Kerberos* actúa como un árbitro confiable. *Kerberos* proporciona un medio seguro de autenticación entre clientes y servidores. Está diseñado para que los passwords no tengan que viajar en la red, donde intrusos pueden capturarlos y se basa en el sistema de encripción DES. Proporciona tickets a los usuarios, que los obtienen de un servidor central de distribución. Estos tickets les proporcionan a los usuarios acceso tanto a los hosts como a los servicios y son enviados encriptados por la red, de tal forma que si alguien los captura no pueda usarlos. [1,2]

La figura 1.21 ejemplifica el protocolo *Kerberos*. En (1) el cliente solicita al servidor de *Kerberos* un ticket para contactar al *Ticket-Granting Service* (*TGS*), que actúa como un KDC. En (2) *Kerberos* le envía el ticket al cliente para que contacte al *TGS*. El cliente en (3) le solicita al

[26] Por sus siglas en ingles Key Distribution Centers.
[27] Ídem Key Translation Centers.

(TGS) un ticket para que se pueda comunicar con el servidor. El *TGS* en (4) le envía el ticket al cliente y este a su vez se puede comunicar con el servidor en forma segura (5).

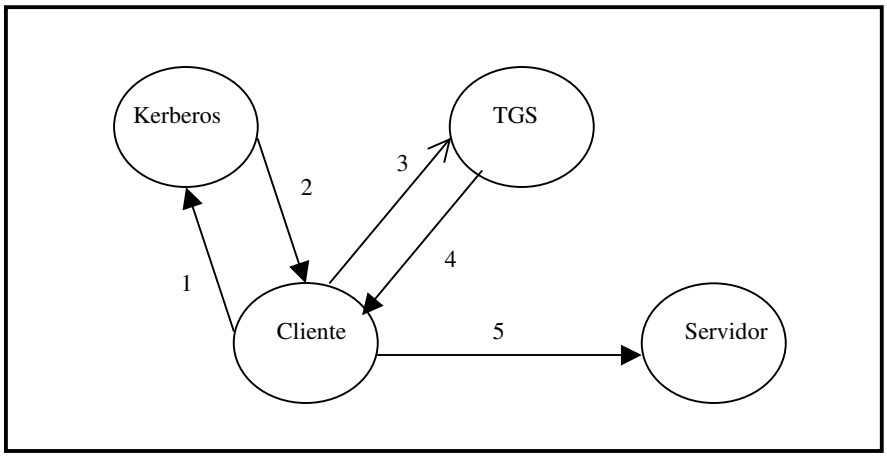


Figura 1.21. Ejemplo de *Kerberos*.

1.3.6.2 Administración de llaves a través de técnicas de llave pública

El uso de criptografía de llave pública requiere que se establezca la autenticidad de las llaves públicas. Si dos usuarios desean comunicarse en forma segura deben intercambiar sus llaves públicas, por lo tanto la distribución inicial de las llaves públicas es de *n(n-1),* debido a lo anterior resulta trascendente el uso de una *TTP* que pueda emitir certificados a cada usuario y con ello un usuario solo tiene que distribuir su llave pública a la *TTP* con lo cuál el tráfico de llaves entre usuarios se limita a uno solo. [2]

Existen dos razones fundamentales para emplear técnicas de llave pública para distribuir llaves: La primera es el uso de encripción de llave pública para distribuir una llave secreta entre los usuarios y la segunda es la distribución de llaves públicas por si mismo.

Diferentes técnicas han sido propuestas para la distribución de llaves públicas, que pueden ser agrupadas en: [23]

- Anuncio público. La llave pública es conocida en forma general por los integrantes de la red, por ejemplo, si se utiliza el algoritmo RSA cualquier integrante de la red puede envíar su llave pública a sus contrapartes, enviando un mensaje que contenga su llave pública a todos los miembros de la red.

- Directorio público. Si se mantiene un directorio disponible de llaves públicas, se alcanza un mayor grado de seguridad para los integrantes de la red. El mantenimiento y distribución del directorio público, se obtiene delegando la responsabilidad a una tercera entidad confiable.

- Autoridad de llave pública. Permite lograr una seguridad aceptable para la distribución de llave pública, aunado a que controle en forma estricta la distribución de llaves del directorio público. Puede combinarse con la técnica anterior fortaleciendo la seguridad de

la red.

- Certificados. El uso de certificados, permite a los usuarios de la red intercambiar llaves, que estarán contenidas en el certificado. Cabe señalar que debe existir una autoridad certificadora, de la cuál se obtendrán los certificados y con ello se podrá realizar la comunicación entre las partes en forma segura.

De las anteriores técnicas, una de las más destacadas es la de certificados. En está técnica el manejo y la confiabilidad en la distribución de certificados recaen en la CA.

Si se utiliza una CA se debe considerar la implementación de una *PKI*[28] , que se puede definir como "la arquitectura, organización, tecnología, prácticas, políticas y procedimientos que en conjunto soportan la implantación y operación de un sistema criptográfico de llave pública basado en certificados". [2]

La *PKI* se caracteriza por requerir la coexistencia de múltiples CAs. Estas CAs certificarán mutuamente sus llaves públicas. También es necesario definir un conjunto de estándares y servicios, que faciliten la administración y el mantenimiento de criptografía de llave pública, como de certificados.

Entre los beneficios de la utilización de las *PKIs,* actualmente se aprecian en distintos ámbitos, ya sea en transacciones financieras que se realizan en internet, o para hacer y recibir pagos de manera segura y confiable, o en el incremento y mejora de la efectividad de los procedimientos de negocio.

Las funciones de una *PKI* se resumen en:

- Emisión de certificados.
- Servicio de directorio (publicación).
- Actualización automática de llaves.
- Historial de llaves.
- Respaldo y recuperación de llaves.
- Renovación y cancelación de certificados.
- Políticas y procesos.

En cuanto a la estructura de una *PKI* está conformada de los siguientes elementos: [2]

- Una entidad final que es el usuario de un certificado.

[28] Ídem Public Key Infrastructure.

- La CA que es la responsable de enviar y revocar certificados.

- La autoridad de registro (*RA*)[29] que es la responsable de establecer la identidad del certificado y el mapeo entre el usuario y su llave pública. La función de registro puede ser realizada por la CA y por lo tanto la *RA* es un componente opcional.

La figura 1.22 ilustra los componentes de la *PKI*.

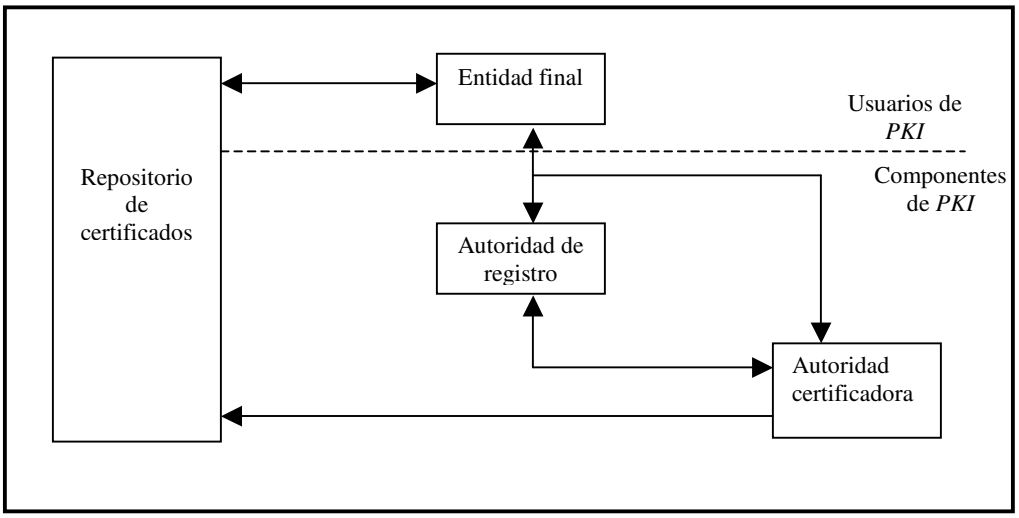


Figura 1.22. Componentes de una *PKI*.

Los servicios básicos que los componentes de una *PKI* pueden ofrecer son: Registro, inicialización, certificación, actualización de llaves y revocación.[3]

El servicio de registro establece, el mapeo entre una entidad final y su llave pública. Para ello a la *RA* se le proporciona la llave pública junto con cualquier información requerida para el certificado, por ejemplo: dirección de correo electrónico, organización, teléfonos, etc.

La *RA* puede requerir que una entidad final demuestre que tiene la llave privada correspondiente, por ejemplo que la entidad final genere una firma digital. Una vez que se realice la demostración, la *RA* necesita verificar la información recibida por la entidad final, si está es correcta contactará a la CA y le solicitará la generación del certificado, de lo contrario la solicitud de registro será denegada.

[29] Por sus siglas en inglés Registration Authority.

El servicio de inicialización requiere que, antes de que una entidad final pueda usar los servicios de ésta, debe ser inicializada. La entidad final solicita el certificado de la CA, el cuál contiene la llave pública necesaria para verificar cualquier certificado enviado por la CA. La inicialización también incluye la generación del par de llaves pública/privada de la entidad final.

El servicio de certificación, se ejecuta cuando se recibe la solicitud del certificado de la *RA*, la CA genera y firma el certificado. Esto incluye el proceso de llenado del certificado con la información que la *RA* proporciona. Por último la CA realiza la firma del certificado con su llave privada sk_{CA}.

El servicio de actualización de llaves, se encarga de validar la vigencia de las llaves, debido a que su validez es por un período de tiempo determinado. Éste servicio se encarga del proceso de renovación de las llaves y del envío del certificado correspondiente a la CA.

La CA es la responsable de mantener el estatus de los certificados que han sido enviados, entre sus funciones esta el servicio de revocación de certificados. Un certificado es inválido por alguna de las siguientes causas: [3]

- Baja solicitada por el usuario.
- Baja por exposición de llaves.
- Baja por finalización del periodo de vida del certificado.
- Baja por abandono de la organización.
- Baja por orden superior (mal uso del Certificado).

Después de que un certificado ha sido enviado necesita que este disponible para el dueño y para otros usuarios que quieran usarlo, para ello, una vez que la CA genera un certificado puede distribuirlo en diversas formas. En el caso de que un certificado sea revocado la *PKI* debe informar a los usuarios de esto.

Un método común es que la CA publique una lista de revocación de certificados (*CRL*)[30] en donde se incluyan todos los certificados que han sido revocados. Los usuarios de los certificados pueden consultar la *CRL* para verificar si un certificado es válido o no.

Un problema de la *CRL* es el tiempo que transcurre entre la notificación de que un certificado ha sido revocado y la revocación de éste. Esto puede representar un riesgo para la seguridad de la *PKI* ya que la *CRL* realiza las notificaciones en intervalos regulares y no las realiza en forma constate. La solución para este problema es un mecanismo de revocación en línea, que permita a los usuarios realizar consultas a la CA y que el resultado de éstas sea en tiempo real.

[30] Por sus siglas en inglés Certificate Revocation List

La figura 1.23 ejemplifica la estructura de una *CRL.* La cuál contiene el nombre de la entidad enviante, la fecha de publicación, el número de serie del certificado del usuario, la fecha de revocación y el motivo.

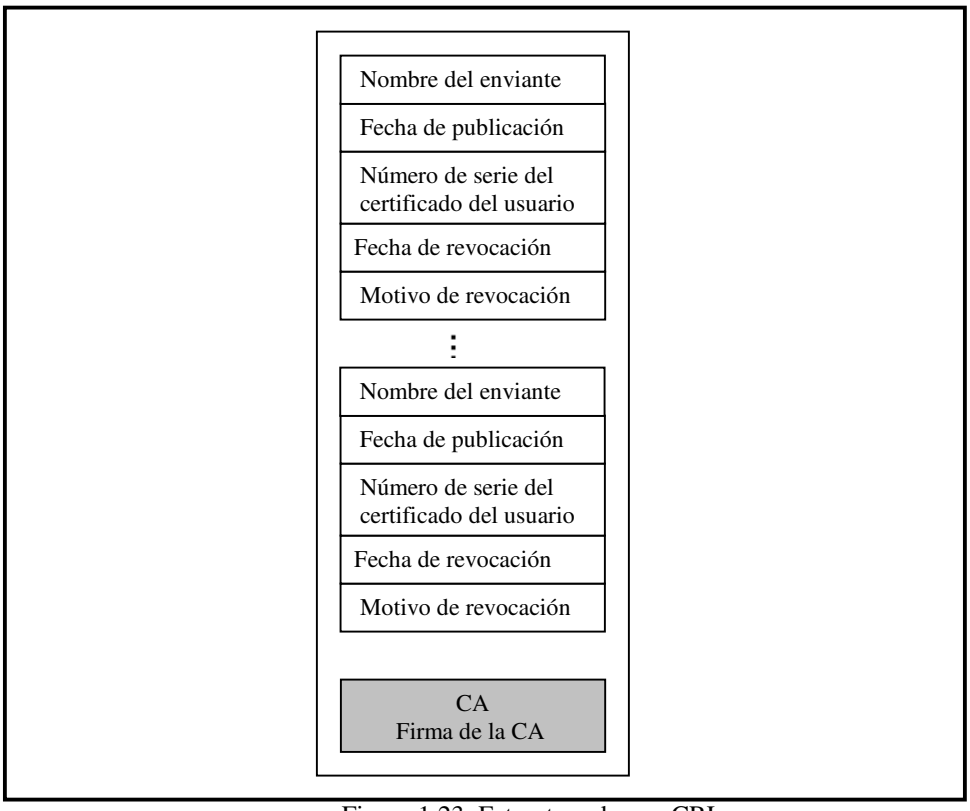


Figura 1.23. Estructura de una CRL.

2 PROBLEMAS DE SEGURIDAD EN MANETS

En este capítulo se abordan los problemas y aspectos de seguridad que presentan las MANETS[31]. Primero se repasarán algunos tópicos de seguridad en redes, posteriormente se muestran los principales problemas de seguridad en MANETS, por último se presentan los diferentes aspectos de seguridad, que deben ser considerados para garantizar que la seguridad de la red no se vea comprometida.

2.1 SEGURIDAD EN REDES

Cuando se analiza la seguridad en redes, cableadas o inalámbricas, se deben considerar tres aspectos: los servicios, los ataques potenciales y los mecanismos de seguridad. Los servicios de seguridad incluyen la funcionalidad que se requiere para tener un ambiente de red seguro, mientras que la seguridad contra los ataques potenciales, agrupa los métodos que un adversario puede emplear en contra de los servicios de seguridad. Finalmente los mecanismos de seguridad son las herramientas que se emplean para proporcionar los servicios de seguridad.

2.1.1 SERVICIOS DE SEGURIDAD

Para que una red se considere segura deben ser considerados los siguientes servicios: [23]

- Confidencialidad: Un sistema posee la propiedad de confidencialidad si los recursos manipulados por éste no son puestos al descubierto para usuarios, entidades o procesos no autorizados.

- Autenticación: Demuestra que el usuario o entidad es quien dice ser. Permite que una entidad verifique la identidad de la otra parte.

[31] En adelante nos referiremos a las redes inalámbricas AD HOC por sus siglas en inglés MANETS

- Integridad: Garantiza que los datos no han sido alterados durante la transmisión de éstos.

- No repudio: Asegura que la entidad o el usuario no niegue que ha realizado alguna actividad.

- Disponibilidad: Permite que los servicios de la red se encuentren utilizables y accesibles para los usuarios cuando sean requeridos.

2.1.2 TIPOS DE ATAQUES

La seguridad contra ataques potenciales, se clasifica conforme a la naturaleza del ataque en las siguientes categorías: [23]

- Ataques pasivos: El atacante solo observa o monitorea el tráfico de la red. Este ataque es muy sencillo de realizar en muchos ambientes de red ya sea cableado o inalámbrico.

- Ataques activos: El atacante además de observar la transmisión de datos en la red va a modificar, inyectar o borrar paquetes.

Una vez que se ha identificado el tipo de ataque, es necesario hacer una clasificación de acuerdo a la acción que realiza el intruso. La siguiente clasificación puede usarse para la mayoría de los ataques.

- Monitoreo: Este ataque es usado para obtener conocimiento de los datos que son transmitidos en la red, para prevenir el monitoreo de la red es recomendable utilizar algún esquema de encripción para proteger los datos transmitidos.

- Análisis de tráfico: A diferencia del ataque anterior, el análisis de tráfico no busca conocimiento directo sobre los datos transmitidos en la red, pero su objetivo es obtener información de las características de la transmisión, por ejemplo la cantidad de datos que son transmitidos o la identidad de la comunicación entre nodos. El atacante, con la información que consigue, puede deducir información sensitiva, ya sea los roles de comunicación entre nodos, o su posición o el sistema operativo. A diferencia del ataque anterior es más difícil de prevenir, pero si se encripta la comunicación entre los nodos, el adversario tendrá mayor dificultad para deducir la información sensitiva.

- *Impersonation*: El atacante utiliza la identidad de un nodo que tenga acceso a la red para hacer uso de los recursos de ésta. Generalmente este ataque se realiza antes que el ataque de monitoreo, si el adversario logra obtener la identidad de un nodo válido entonces podrá acceder a la llave de encripción, que es usada para proteger la transmisión de datos. Una vez que es conocida la llave por el atacante, podrá realizar el monitoreo de la red sin problema.

- Modificación: El objetivo es alterar los datos durante la transmisión entre los nodos.

- Inserción: Este ataque involucra una entidad no autorizada que inyecta nuevos datos atribuyéndole la responsabilidad a un nodo legítimo.

Esta relacionado con el ataque de *impersonation* (hacerse pasar por un nodo válido).

- Retransmisión: El atacante reenvía los datos que ya han sido transmitidos por un nodo legítimo.

- Denegación de servicio: El objetivo es limitar o negar el acceso a cierto recurso, el recurso puede ser un nodo, un servicio o toda la red.

2.1.3 PROPIEDADES DE UN SISTEMA SEGUROS

La mayoría de los servicios de seguridad mencionados en la sección 2.1.1, pueden ser implementados utilizando diferentes técnicas. La implementación de los servicios de seguridad se lleva a cabo mediante mecanismos que deben proporcionar los siguientes elementos:

- Confidencialidad: El requerimiento más común es que la información transmitida nunca debe ser expuesta a ninguna entidad no autorizada. El servicio de confidencialidad puede ser de dos formas distintas, la primera es dónde se requiere protección contra ataques de monitoreo, para ello se utiliza algún esquema de encripción. La segunda es cuando se quiere proteger la confidencialidad de los datos contra ataques de análisis de tráfico, lo que implica que además de utilizar un esquema de encripción se requiere un mecanismo adicional como redes privadas virtuales (VPNS). En las MANETS la confidencialidad asegura que los datos entre los nodos, no son accesibles a ninguna entidad no autorizada y que la información que es transmitida entre ellos no es comprensible para ningún otro nodo.

- Integridad: Garantiza que un mensaje que es trasferido de un nodo a otro, no ha sido alterado por una entidad no autorizada. Un mensaje puede ser modificado por alguna causa natural (lluvia o tormentas eléctricas) o porque un adversario en forma intencional altere el contenido de los paquetes que viajan en el medio.

- Autenticación: Permite que un nodo asegure la identidad del nodo con el que se comunica, sin autenticación un atacante puede hacerse pasar como un nodo válido en la red y de esta forma puede acceder a información sensitiva.

- No repudio: Asegura que un nodo no pueda negar que ha recibido o enviado un mensaje. Requiere el uso de criptografía de llave pública para proporcionar el servicio de firma digital.

- Disponibilidad: Garantiza que los servicios de red se encuentren utilizables y accesibles aún si se llegará a presentar un ataque denegación de servicio.

2.2 PROBLEMAS DE SEGURIDAD EN MANETS

La seguridad en las MANETS persigue los mismos principios que aplican en los sistemas de información: confidencialidad, integridad, disponibilidad, autenticación y no repudiación, pero con la particularidad de que los datos no son transportados a través de un medio físico (redes cableadas), sino que el medio de transporte es inalámbrico, por lo que existe la vulnerabilidad de que un usuario no autorizado tenga acceso a recursos e información sensitiva de la organización.

Los requisitos de seguridad en las MANETS, son los mismos que los que aplican en redes cableadas, sin embargo sus características generales como son: la topología dinámica, los enlaces de ancho de banda limitado, las limitaciones de energía, la capacidad de procesamiento en los nodos, los problemas para centralizar la autoridad certificadora y la seguridad física limitada, hacen que el cumplimiento de los requisitos de seguridad sean un problema mucho más complejo de abordar. Los requerimientos de seguridad en las MANETS se pueden resumir en:

- Algoritmo de encripción: Proporciona la privacidad de los datos.
- Integridad de Mensajes: Asegura que los frames de datos son confiables y son enviados por el nodo con el que se tiene el intercambio de mensajes.
- Autenticación del frame de trabajo: Facilita la autenticación de mensajes entre clientes.
- Algoritmo de autenticación: Valida las credenciales de un cliente.

La figura 2.1 representa como cada uno de los requerimientos deben estar integrados para garantizar que la seguridad de la red sea aceptable.

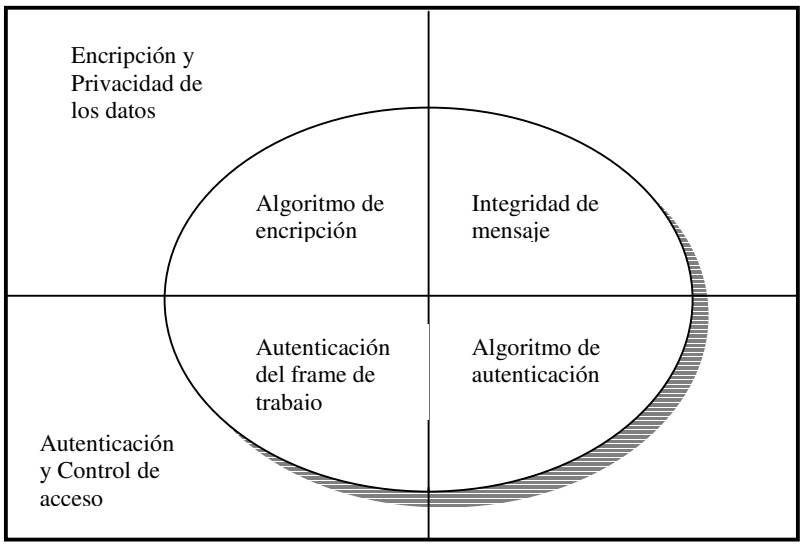


Figura 2.1. Requerimientos de seguridad para las MANETS.

Cabe señalar que las investigaciones de seguridad en MANETS, se han enfocado en el área de protocolos y asumen que existe un acuerdo de distribución de llaves o secreto compartido entre los nodos de la red, algunos protocolos que ejemplifican los anterior son: ARAN, ARIADNE, SEAD, SPINS, y SRP por lo que el problema de distribución y administración de llaves sigue vigente.

2.2.1 PRINCIPALES PROBLEMAS DE SEGURIDAD EN MANETS

Las MANETS son susceptibles a una gran cantidad de ataques debido a su naturaleza, como son: la movilidad, la espontaneidad y el ancho de banda limitado. Los ataques pasivos y activos pueden ser lanzados por un atacante con facilidad. En los ataques pasivos, el atacante solo se limita a monitorear el tráfico de las MANETS, sin realizar ninguna interferencia en la red. En contraparte los ataques activos le permitirán al adversario borrar mensajes, inyectar mensajes erróneos, modificar mensajes y hacerse pasar por un nodo válido (*impersonation*), logrando violar la disponibilidad, integridad, autenticación y no repudio en las MANETS, con ello la seguridad de la red se vería comprometida.

Cuando dos nodos establecen un canal de comunicación, son vulnerables a un ataque a la conexión del medio inalámbrico, lo que incluye monitoreo, ruido del canal, acceso a información sensitiva, distorsión de mensajes y denegación de servicio. Lo anterior pudiera mitigarse si se utiliza un esquema de administración de llaves y certificados para garantizar que los mensajes entre los nodos sean siempre cifrados.

La falta de una autoridad certificadora centralizada en MANETS, dificulta la implementación de un esquema de encripción y autenticación. Porque no puede existir una autoridad administrativa común para todos los nodos. En caso que se tuviera centralizada la administración de las llaves en un solo nodo, sería muy riesgoso para la seguridad de la red, porque si el nodo llegase a ser atacado por un adversario, la seguridad de las MANETS estaría comprometida.

Aunado a lo anterior existe el problema del medio físico de comunicación, las MANETS puede utilizar distintos protocolos para comunicarse entre si, algunos de ellos son: *802.11*, *bluetooth* o infrarrojo. Como se menciono en el capítulo 1, en este trabajo se toma como base para la comunicación entre nodos el protocolo *802.11*.

A continuación se presentan algunas vulnerabilidades, que no son exclusivas de MANETS, sino que también se presentan en redes inalámbricas con punto de acceso (AP). También se muestran algunas fallas que ha presentando el protocolo *802.11*.

a) Identidad:

Un elemento esencial en cualquier arquitectura de seguridad, es un mecanismo confiable que garantice la identidad del usuario. Cualquier atacante puede hacerse pasar por un usuario válido y comprometer el sistema, en una red inalámbrica con el protocolo *802.11* la dirección MAC de la tarjeta inalámbrica es utilizada como una forma para identificar usuarios y dispositivos, si un atacante logra usurpar la dirección MAC de un nodo válido vulneraría la seguridad de la red.

Por otra parte dentro de las características del protocolo *802.11* cada frame de datos tiene una dirección fuente pero no garantiza que el nodo enviante sea el que puso el frame en el medio, los atacantes pueden usar frames inválidos para re-direccionar el tráfico y corromper las tablas ARP[32], en un nivel muy simple, el atacante puede observar la dirección MAC de las estaciones de la red y adoptar esa dirección para trasmisiones maliciosas.

b) WEP (Wired Equivalent Privacy*)* [4]:

El protocolo *802.11* define un mecanismo de confidencialidad de datos conocido como WEP, que utiliza el algoritmo RC4 para el proceso de encripción con llave simétrica. El objetivo de WEP es la confidencialidad de los datos, pero este no ha podido cumplirse porque aparecieron fallas en el diseño provocando que se rediseñara el protocolo.

El proceso de encripción de una trama con WEP se realiza en tres etapas:

1. Suma de comprobación (*checksum*): En la primera se calcula una suma de integridad *c(M)* sobre el mensaje *M*. Se concatenan ambos para obtener un texto sin cifrar *P(texto plano) = (M, c(M))*, el cual será empleado como entrada para la segunda etapa. Nótese que *c(M)*, y por ende *P*, no depende de la llave *k*.

2. Encripción: En la segunda etapa se cifra el texto plano *P* con el algoritmo RC4. Se elige un vector de inicialización (*IV*). El algoritmo RC4 genera un *keystream* –una secuencia larga de bytes pseudo aleatorios- como una función de *IV* y la llave *k*. Una vez obtenido se realiza un or-exclusivo (XOR, denotado por (+)) sobre el texto plano con el *keystream* obtenido para obtener el texto cifrado:

   C=P(+)RC4(V,K)

3. Transmisión: Finalmente en la tercera etapa, se transmite *IV* y el texto cifrado a través del enlace de radio. Simbólicamente, este proceso se representa como:

   A -> B: v,(P (+) RC4(v, k)); donde P = (M, c(M))

El proceso de encripción se muestra en la figura 2.2

[32] Por sus siglas en inglés Address Resolution Protocol.

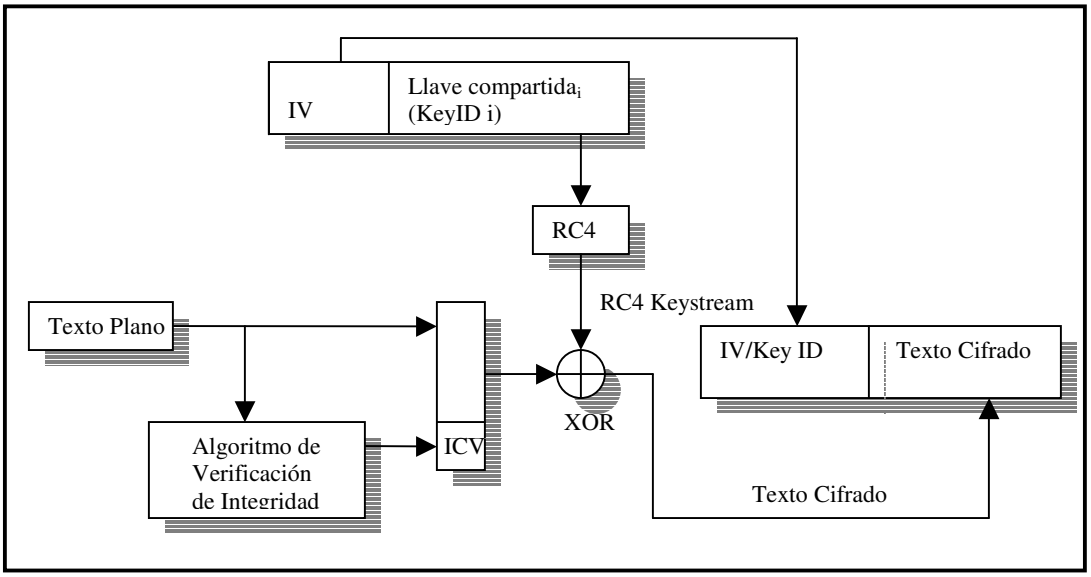


Figura 2.2. Funcionamiento del algoritmo WEP en la modalidad de cifrado.

Para desencriptar una trama cifrada por WEP, el receptor invierte el proceso de encriptado. Primero, genera el *keystream* RC4 (v, k) y efectúa un or-exclusivo contra el texto cifrado para recuperar el texto plano original [4]:

P' = C (+) RC4(v, k)

= (P (+) RC4(v, k)) (+) RC4(v, k)

= P

Después, el receptor verifica el *checksum* del texto desencriptado *P'* separándolo conforme a (M', c'), y recalcula el *checksum* c(M'), por último comprueba que coincide con el *checksum* recibido *c'*. Esto asegura que sólo las tramas con un *checksum* válido son aceptadas por el receptor.

El proceso de descifrado se muestra en la figura 2.3

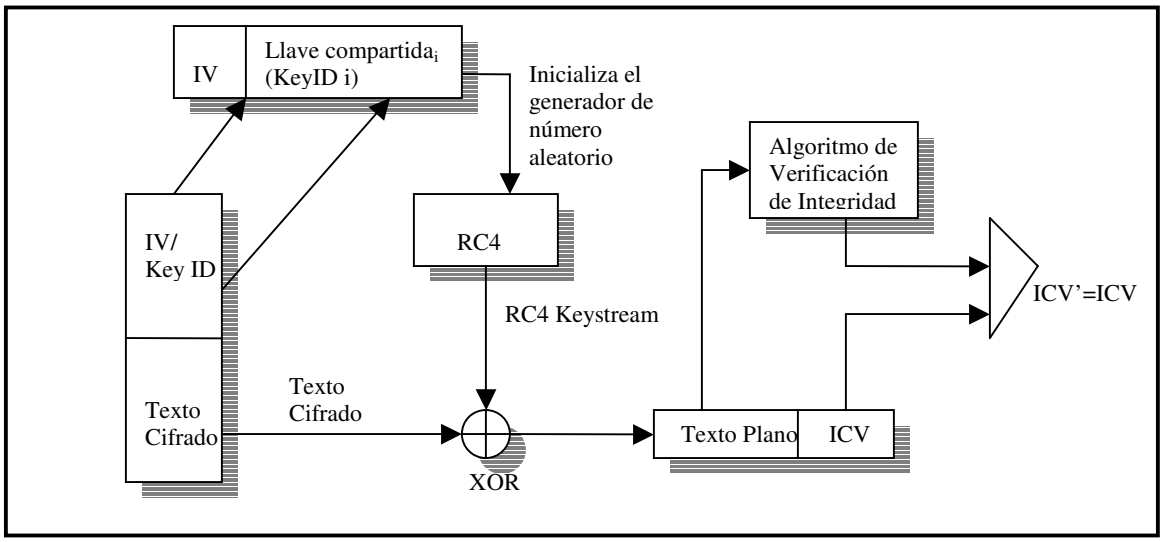


Figura 2.3. Funcionamiento del algoritmo WEP en la modalidad de descifrado.

Los principales problemas que presenta WEP en la implementación son:

- El valor de integridad (*IV*) de 24 bits es pequeño.

- El *checksum* CRC, llamado Valor de Chequeo de Integridad (*ICV*), usado por WEP para protección de integridad es inseguro y no previene que el atacante realice modificaciones o intercepte paquetes.

- WEP combina el valor de integridad (*IV*) con la llave habilitando un criptoanálisis, como resultado un atacante que este monitoreando el tráfico de la red podrá obtener la llave después de observar algunos miles de paquetes encriptados.

- No existe protección de integridad para el enviante y el destinatario

Además de los problemas anteriores, existen otras vulnerabilidades dentro de WEP. La primera es que el uso de WEP es opcional. La segunda es que WEP por default, utiliza una llave simétrica común para todos los nodos en una red inalámbrica, comúnmente la llave es almacenada en los nodos, si algún nodo es comprometido la única solución es cambiar la llave secreta de todos los dispositivos. En la práctica el problema más serio de WEP, radica en que las llaves encriptadas con RC4 pueden ser recuperadas fácilmente utilizando criptoanálisis. En MANETS la solución de implementar WEP, implica que cada nodo tenga una llave secreta compartida, haciendo que los requerimientos de movilidad y espontaneidad se dificulten.

c) Rendimiento y servicio

Las redes inalámbricas tienen capacidad limitada de transmisión, las redes basadas en *802.11b* tienen un promedio de transmisión de 11 Mbps, las redes basadas en *802.11a,* alcanzan hasta 54 Mbps, esta capacidad de transmisión es compartida entre todos los usuarios asociados, la velocidad de transferencia real se reduce a la mitad debido a una sobrecarga de la capa MAC, porque ésta es la encargada de realizar el control de acceso medio y constantemente se esta actualizando. No es difícil imaginar que el rendimiento de una aplicación empresarial que maneje

un control de inventarios ó nomina se vea afectado por está capacidad limitada, tampoco es difícil concebir, que un atacante pueda realizar un ataque denegación de servicio que inunde las MANETS con paquetes inválidos, aumentando el tráfico y reduciendo el rendimiento de la red.

La MAC del protocolo *802.11* esta diseñada para permitir que múltiples redes compartan el mismo espacio y canal de radio. Los atacantes que quieran dar de baja o sacar de servicio las MANETS podrían generar tráfico inválido y enviarlo al mismo canal de radio, la red destino acomodaría el nuevo trafico usando el mecanismo estándar de CSMA/CA[33] que tiene la función de controlar el acceso al medio.

d) Monitoreo

El protocolo *802.11*, no proporciona un mecanismo contra ataques pasivos que estén monitoreando el tráfico de la red. Los encabezados de los frames viajan en texto claro y son visibles para cualquiera que tenga un software que este monitoreando el medio. Con WEP se intento hacer un esfuerzo para mitigar este problema pero, como se explicó, este mecanismo no tuvo un diseño adecuado. Un atacante cuenta con diferentes técnicas para localizar redes inalámbricas, entre las cuales destacan:

- WarWalking , si se realiza caminando.
  .
- WarSkating, con patines.
- WarCycling, si es en bicicleta.
- WarDriving, en un automóvil.

La figura 2.4 nos ejemplifica el proceso de WarWalking, que consiste en caminar por la calle con un dispositivo portátil dotado de una tarjeta inalámbrica *802.11*, buscando la señal de redes inalámbricas y software para detección de redes inalámbricas, que se consigue libremente en internet.

Figura 2.4 .WarWalking, tratando de localizar redes inalámbricas caminando.

[33] Por sus siglas en inglés Carrier Sense Multiple Access/with Collision Avoidance.

Es recomendable si se desea aumentar la ganancia de la señal recibida el utilizar algún dispositivo (como un bote de aluminio o lata de papas) como muestra la figura 2.5.

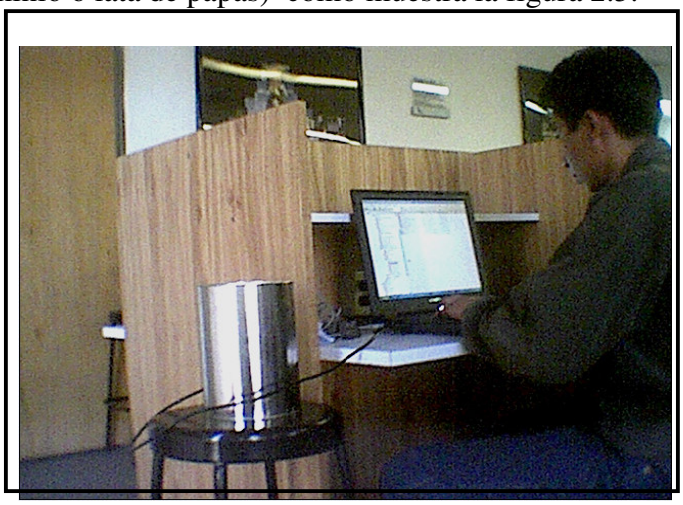


Figura 2.5. Bote para aumentar la ganancia de la tarjeta inalámbrica.

En la figura 2.6 se muestra un ejemplo de *wardriving*, que se realiza para localizar redes inalámbricas desde un automóvil.

Para este fin se necesita de un dispositivo portátil dotado de una tarjeta inalámbrica *802.11*, una antena adecuada (que se puede elaborar fácilmente con una lata de conservas o de papas fritas, ver figura 2.5), un GPS para localizar los puntos de acceso en un mapa, y software para detección de redes inalámbricas, que se consigue libremente en internet.

Figura 2.6. WarDriving localizando redes inalámbricas en el automóvil.

e) Denegación de servicio

Un ataque denegación de servicio puede ser lanzado a cualquier capa de una MANETS. En la capa física y de acceso al medio un atacante puede lanzar interferencias que afectan la comunicación con el canal físico. En la capa de red puede alterar el protocolo de ruteo y desconectar a los nodos de la red. En capas superiores se puede dar de baja los servicios de

administración de llaves, el cuál es fundamental para la seguridad de la red. Este tipo de ataque es de alto impacto en la seguridad de cualquier organización.

f) Disponibilidad

Debido a que las MANETS operan en condiciones dinámicas e impredecibles, la disponibilidad es un problema en este tipo de redes. Los nodos en una red alámbrica pueden estar en servicio o fuera de servicio, en las MANETS no se puede hacer la afirmación anterior porque no se pueden realizar suposiciones sobre la disponibilidad del sistema y no se tiene certeza de que los nodos estén disponibles, de ahí radica la importancia de que el protocolo de ruteo realice las actualizaciones correspondientes a la tabla de ruteo para mantenerla actualizada y de esta forma tener la certeza de que nodos se encuentran conectados en la red.

g) Autoridad certificadora centralizada

La autoridad certificadora en una red AD HOC no debe centralizarse, como ocurre en las redes con cableado estructurado. Una autoridad certificadora centralizada puede representar un problema para la administración de certificados, en cuanto a la distribución, renovación y revocación de certificados. Por lo tanto pensar en centralizar la administración de llaves en éste tipo de redes no puede considerarse como una solución viable, a pesar de esto, algunos autores han propuesto el utilizar esquemas de autoridades centralizadas o semi-centralizadas en MANETS, cada una de ellas serán revisadas en el capitulo 3.

h) Ataque de consumo de recursos [22]

Un nodo malicioso puede consumir en forma innecesaria recursos como: ancho de banda o procesamiento, que son limitados en algunas MANETS. El ataque puede lanzar consultas en forma constante al protocolo de ruteo o puede generar paquetes de anuncio de conexión (beacon frames) inválidos o reenviar paquetes caducos a los nodos.

i) Ataques al protocolo de ruteo

Existen distintos tipos de ataques que son lanzados al protocolo de ruteo, los cuales tienen el objetivo de corromper la operación de la MANET, estos se pueden clasificar en: ataques de interrupción al protocolo de ruteo y ataques de consumo de recursos. En el primer ataque, el adversario envía paquetes válidos para que sean enrutados en forma disfuncional. En el segundo ataque, el adversario inyecta paquetes en la MANET para consumir recursos valiosos en la red como son: ancho de banda, memoria o poder de computo. [33]

A continuación se describen algunos ataques al protocolo de ruteo:

- Ataque *blackhole.* [33]

  Un nodo malicioso dentro de la MANET, anuncia rutas falsas como correctas, hacia un nodo destino durante el proceso de búsqueda de rutas o en el proceso de actualización de mensajes. La intención de éste ataque es el de obstaculizar el proceso de búsqueda de rutas o el de interceptar todos los paquetes de datos que van a ser enviados al nodo correspondiente.

- Ataque *Byzantine.* [22]

En éste tipo de ataques, se requiere que algunos nodos estén comprometidos por el atacante y que éstos puedan formar una colisión, para crear rutas falsas o cíclicas de ruteo o borrado intencional de ciertos paquetes. Este tipo de ataque es muy difícil de detectar, porque la MANET va a presentar un comportamiento "normal" desde el punto de vista de los nodos, a pesar de que pueden estar recibiendo un ataque *byzantine*.

- Ataque *wormhole.* [33]

En este ataque el adversario recibe paquetes de un nodo válido y los túnelea a otra distinta, cabe señalar que en éste tipo de ataques deben estar coludidos por lo menos dos atacantes.

- Desbordamiento de la tabla de ruteo. [22]

En éste tipo de ataques, un nodo adversario anuncia a los nodos válidos de la red rutas hacia nodos no existentes. El objetivo de este ataque es causar un desbordamiento a las tablas de ruteo, el cual previene la creación de entradas a nuevas rutas de nodos autorizados.

- Envenenamiento de las tablas de ruteo. [22]

El nodo comprometido en la MANET, envía actualizaciones de rutas ficticias o modifica actualizaciones correctas de rutas enviando paquetes a los nodos no comprometidos. Este ataque puede dar como resultado un congestionamiento en algunas partes de la MANET o que ésas partes queden inaccesibles.

- Reenvío de paquetes. [22]

El nodo adversario reenvía paquetes obsoletos. Este ataque consume ancho de banda y recursos de batería de los nodos, causando confusión entre los protocolos de ruteo.

Conforme a los distintos ataques que pueden ser victima las MANETS, la figura 2.7 ilustra una clasificación de los ataques conforme a la capa de red que se ve afectada.

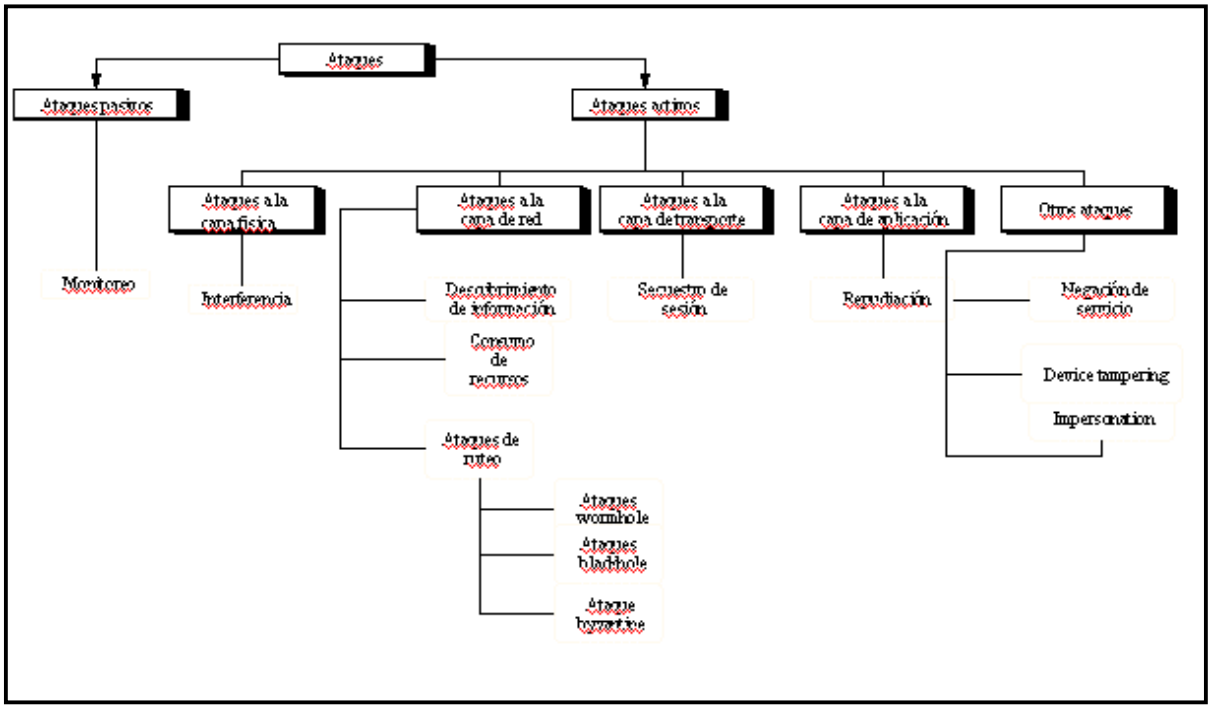


Figura 2.7. Tipos de ataques a MANETS.

2.3 ASPECTOS DE SEGURIDAD EN MANETS

La política de seguridad que se aplica en las MANETS depende de las aplicaciones que soporta y de la topología de la red. En el artículo de *Zheng Yan* [5] se identifican tres aspectos que deben ser cubiertos por cualquier política de seguridad en las MANETS que son: sistemas de detección de intrusiones (SDI), seguridad de los protocolos de enrutamiento y servicios de distribución de llaves. A continuación se retoman cada uno de ellos.

2.3.1 SISTEMAS DE DETECCIÓN DE INTRUSIONES (SDI)

Las técnicas de prevención, como encripción y autenticación, son necesarias como primera línea de defensa, sin embargo las MANETS presentan vulnerabilidades inherentes que no son fácilmente previsibles. La detección de intrusiones permite establecer una segunda línea de defensa que es necesaria para redoblar la seguridad de la red.

En el artículo de *Y. Zhang y W. Lee* [24] los autores proponen una arquitectura distribuida y cooperativa para la detección de intrusiones que utiliza un modelo de detección de anomalías. En el sistema propuesto, cada nodo de la red ejecuta un agente detector de intrusiones que se encarga de monitorear las actividades locales. Si el *SDI* detecta una anomalía, pero no tiene una evidencia concluyente de que se esté produciendo un ataque, puede iniciar un proceso cooperativo con sus nodos vecinos, de modo que puedan determinar finalmente si la intrusión ha tenido o no lugar. Los autores proponen además una estructura multicapa, en la que la detección se realiza a diferentes niveles en el *stack* de protocolos.

Por su parte O. Kachirski y R. Guha. [25] proponen un *SDI* distribuido basado en tecnología de agentes móviles. Un agente móvil es una entidad de software autónoma que puede ser actualizada dinámicamente y se ejecuta sobre ciertos nodos. Los autores indican que la tecnología de agentes es adecuada en MANETS, dónde los recursos de ancho de banda en los enlaces pueden ser limitados. Proponen una arquitectura en la cual las funciones a realizar por el *SDI* se distribuyen entre los diferentes tipos de agentes, de modo que la carga introducida por el *SDI* se distribuye de forma eficiente entre los nodos de la red. En cualquier caso, el empleo de técnicas de detección dependerá siempre de las características de la aplicación y del escenario concreto sobre el que dicha aplicación se ejecuta.

Dada la sobrecarga que pueden introducir estos mecanismos, en términos de transmisión sobre el medio inalámbrico, de procesamiento y almacenamiento en los nodos, su uso puede resultar justificable en aplicaciones con fuertes requisitos de seguridad, en las cuáles, los dispositivos involucrados cuenten con suficiente capacidad de almacenamiento y autonomía, para que la ejecución de un sistema de detección no imponga una limitación en los servicios ofrecidos al usuario final. Por otro lado, los modelos de detección tradicionales no son aplicables en este escenario. Las diferencias que existen con respecto a las redes convencionales deben tomarse en cuenta por cualquier *SDI* aplicable en MANETS.

2.3.2 SEGURIDAD EN EL ENRUTAMIENTO

Los protocolos de enrutamiento en MANETS se pueden dividir en tres grupos: proactivos, reactivos y basados en *cluster*. Los protocolos proactivos son aquellos que mantienen una ruta hacia todos los nodos, no obstante que en ese momento no la utilicen. En el caso de los protocolos reactivos optimizan el uso de ancho de banda descubriendo la ruta hacia un destino cuando se envía un paquete. Finalmente los protocolos basados en *cluster* son una mezcla de los dos anteriores, y se basan en definir jerarquías entre los nodos de la red y mantener información sobre la topología local.

Los nodos en las MANETS actúan como ruteadores y participan en el protocolo de enrutamiento, para descubrir y mantener rutas a otros nodos de la red.

En general, el objetivo de un algoritmo de enrutamiento es establecer una ruta adecuada entre cada par de nodos. Si el resultado de este algoritmo es manipulado, el funcionamiento normal de las MANETS puede verse seriamente afectado, actuando en contra del requisito de disponibilidad. Por este motivo la seguridad en el enrutamiento tiene un gran peso sobre la seguridad del sistema.

Los requerimientos para que el protocolo de ruteo en MANETS se considere seguro son: [22]

- Detectar nodos maliciosos

  Un protocolo seguro debe detectar la presencia de nodos maliciosos en MANETS y debe prevenir la participación de éstos en el proceso de ruteo. Si un nodo malicioso llegará a participar en el proceso de descubrimiento de rutas, el protocolo debe seleccionar aquellas rutas en las cuáles no este presente el nodo malicioso.

- Garantizar que el descubrimiento de rutas sea correcto

  Si existe una ruta entre el nodo emisor y el nodo destino, el protocolo de ruteo debe encontrar la ruta óptima y debe asegurar que en el proceso se elige al nodo correcto.

- Estabilidad contra ataques

  El protocolo de ruteo, debe ser estable por si mismo, en el sentido de que pueda ser capaz de conservar la estabilidad de las tablas de ruteo, aún si se encuentra en proceso de un ataque pasivo o activo.

A continuación se retoman algunas soluciones, en donde se aborda la seguridad del protocolo de enrutamiento y una clasificación de los tipos de ataques, a los que puede estar expuesto un nodo cuando un atacante intenta violar el protocolo de enrutamiento.

J. Lundberg. [26] en su artículo señala una serie de criterios que debe cumplir un protocolo de enrutamiento seguro. Además, se identifican los principales ataques contra los mecanismos de enrutamiento en las MANETS, clasificándolos en pasivos y activos.

En el artículo de *V. Kärpijoki* [27] menciona que los ataques activos se clasifican en externos e internos. Los ataques externos son realizados por nodos que no pertenecen a la red. Estos ataques incluyen la inyección de paquetes erróneos de enrutamiento, reenvío de información antigua de enrutamiento y distorsión de la información de enrutamiento intercambiada entre los nodos de la red. Las medidas de prevención, tales como encripción y autenticación, pueden establecer la defensa contra este tipo de ataques. Los ataques internos proceden de nodos comprometidos pertenecientes a la red. Esta es una amenaza grave y los *SDI* pueden jugar un papel fundamental en la detección de este tipo de ataques.

R. Ramanujan [28] propone el conjunto de técnicas para diseñar algoritmos de enrutamiento para MANETS resistentes a intrusiones llamado *TIARA*. Estas técnicas son independientes del algoritmo de enrutamiento, para su implementación los algoritmos de enrutamiento de las MANETS deben ser modificados. Y el mismo autor en su artículo [29] propone un esquema que permite construir MANETS resistentes a intrusiones. El enfoque se basa en extender las capacidades de los algoritmos de enrutamiento para MANETS existentes, sin tener que modificar dichos algoritmos. Esta propuesta utiliza mecanismos *TIARA*.

Por su parte *P. Papadimitratos y Z. J. Haas* [30] plantean un algoritmo de enrutamiento seguro denominado *SRP*, que proporciona información de conectividad correcta, actualizada y autenticada a cada par de nodos que desean establecer una comunicación segura. Para ello, el único requisito que se requiere, es la existencia de una asociación segura entre el nodo que inicia la comunicación y el nodo destino.

Y-C. Hu [31] describe otro protocolo de enrutamiento seguro, SEAD, que se basa en el protocolo de enrutamiento *Destination-Sequenced Distance-Vector* (*DSDV).* Los autores indican que *SEAD,* es robusto ante múltiples ataques no coordinados, que provocan un estado erróneo en la información de enrutamiento de cualquier nodo de la red. Y el mismo autor en su artículo [32]

propone un algoritmo de enrutamiento seguro bajo demanda, *ARIADNE*, que se basa en el uso de criptografía simétrica.

La tabla 2.1 muestra los diferentes ataques y las soluciones que existen empleando alguno de los protocolos de ruteo anteriores: [22]

Tabla 2.1 Ataques y defensa contra éstos.

Ataque	Destino en el stack de protocolos	Solución propuesta
Interferencia	Capa fisica y MAC	*FHSS, DSSS*
Wormhole	Capa de red	*Packet Leashes* [34]
Blackhole	Capa de red	[35]
Byzantine	Capa de red	[36]
De consumo de recursos	Capa de red	SEAD [31]
Descubrimiento de información	Capa de red	SMT [30]
Descubrimiento de ubicación	Capa de red	SRP [30]
De Ruteo	Capa de red	SRP [30], ARIADNE [32], SEAD [31], TIARA [28],
Repudiación	Capa de aplicación	ARAN [37]
Negación de servicio	Multi-capas	SEAD [31], ARIADNE [32]
Impersonation	Multi-capas	ARAN [37]

2.3.3 SERVICIOS DE DISTRIBUCIÓN DE LLAVES

Como todo sistema distribuido, la distribución de llaves en MANETS se basa en el uso de un sistema de administración de llaves. La topología de las MANETS son divergentes, por lo que una solución general para administrar llaves no es factible porque se tiene que adaptar a los requerimientos de cada MANETS.

Para proteger a los nodos de la MANETS contra un ataque de monitoreo se utiliza la encripción del tráfico, pero es necesario que antes los nodos establezcan un acuerdo de secreto compartido o intercambio de llaves públicas.

En redes cableadas el problema de distribución de llaves públicas se resuelve con el uso de una entidad confiable la cuál se encarga de certificar las llaves públicas y asociarlas con su dueño, esta entidad es denominada Autoridad Certificadora (*CA*). Para que una llave pública sea certificada, la *CA* emite un certificado firmado con su llave privada permitiendo que la llave pública pueda verificarse. La *CA* es un sistema administrador de llaves públicas. Los sistemas que administran llaves simétricas son denominados *KDC* y *KTC*. En un ambiente donde se

utilicen llaves simétricas, se requiere instalar un *KDC*, y por lo tanto un nodo no puede compartir su llave secreta con cualquier nodo porque sólo las puede compartir con el *KDC*.

El paradigma de las MANETS hace que la existencia de una sola *CA* no sea adecuado, debido a que cada nodo de la red puede dar el servicio de una *CA*, como consecuencia la mayoría de las soluciones para administrar llaves en las MANETS proponen que el rol de la *CA* sea distribuido en los nodos, utilizando algún esquema criptográfico.

En el siguiente capítulo se presentan propuestas de solución para la administración de llaves en MANETS así como la distribución, generación y revocación de certificados.

6. CONCLUSIONES

El concepto de MANETS permite que una gran cantidad de aplicaciones y servicios en red puedan beneficiarse de las ventajas que ofrecen, como son: su topología dinámica, movilidad y espontaneidad. Debido a estas características, los diseños de seguridad tradicionales no pueden ser incorporados directamente, es necesario cambiar su diseño.

En este trabajo se estudió una problemática en la seguridad de las MANETS, que es el manejo de llaves y certificados. Ésta no es simple debido a que las soluciones tradicionales proponen que la autoridad certificadora se encuentre centralizada, pero la naturaleza de las MANETS hace esto poco factible. Es necesario desarrollar, o en su caso adaptar, soluciones específicas que cumplan con los requerimientos necesarios para que el flujo de información entre los nodos se realice en forma segura. Se retomaron las soluciones más trascendentales, que cumplen con el objetivo de proporcionar seguridad a los nodos en MANETS. Sin embargo algunas de éstas, no cumplen con el requisito de no centralizar la autoridad certificadora.

Las soluciones de Autoridad Certificadora Parcialmente Distribuida y Completamente Distribuida manejan un esquema centralizado y semi-centralizado de administración de certificados. Por lo que, el objetivo de no depender de una autoridad certificadora centralizada no se cumple. La solución de Pebblentes utiliza un esquema de llave simétrica, y también centraliza el manejo de la autoridad certificadora. La solución de Identificación Demostrativa es adecuada para MANETS espontáneas, cumple con los requisitos de no centralizar la administración de certificados y de no centralizar la autoridad certificadora. El único inconveniente que presenta, es que depende de un dispositivo adicional al 802.11 para realizar un proceso de pre-autenticación.

De las soluciones presentadas, la que más se acerca con los requisitos de las MANETS es la solución de Autoemisión de Certificados, al no centralizar la autoridad certificadora. Pero presenta problemas con la revocación y actualización de certificados, debido a que no cuenta con ningun mecanismo para ello. Y debido a esto, se pueden presentar problemas de seguridad en MANETS porque no se tiene control sobre los certificados vencidos.

Nuestra solución, denominada *PACDRE*, es una modificación de este trabajo. Y da solución al problema de revocación y actualización de certificados al incorporar un repositorio adicional, así como, los mecanismos para la manipulación de éstos.

El manejo entre repositorios se realiza con SQL, la decisión de emplear este lenguaje es que es el estándar ANSI para manipulación de base de datos, además de su facilidad de uso.

La principal ventaja de *PACDRE* es que, con la incorporación del repositorio adicional, se logra el control de los certificados que son válidos y los que no lo son. Como consecuencia los nodos al implementar *PACDRE*, cuentan con un mecanismo que les permite controlar, autenticar y asegurar el intercambio de certificados, en forma segura. Aunado a lo anterior, otra ventaja que proporciona nuestra solución, es que el usuario no tiene que preocuparse por la renovación y revocación de certificados. Ésta tarea la realiza *PACDRE* con los procedimientos de SQL (*proc_actualiza, proc_renva*, y *proc_revoc*) que están programados para que se ejecuten de acuerdo a una calendarización.

Con PACDRE se podrá tener la certeza de que los certificados son válidos, pero se tiene el riesgo de que los certificados sean robados o falsificados, para ello se sugiere que adicionalmente se utilice un esquema de firma digital (MD5,SHA-1 o RIPEMD-160) para brindar integridad, confidencialidad y no repudio de la procedencia de los certificados. El proceso de la creación de un esquema de firma digital, queda fuera del alcance de éste trabajo de investigación que podrá retomarse en un trabajo futuro.

Por último hay que tener presente que cada MANET es distinta de otra, en el número de nodos y dispositivos que la conforman, como para definir una arquitectura general de seguridad para MANETS. Para ello, se tienen que tomar en cuenta factores como protocolo de ruteo a utilizar, esquema de manejo de llaves, dispositivos para comunicación de nodos, capacidad de almacenamiento y poder de cómputo, entre otros. Pero si se requiere de un diseño que pueda satisfacer el esquema del manejo de llaves y certificados, *PACDRE* proporciona los mecanismos necesarios para garantizar que los certificados de los nodos serán válidos y que pueden contar con un mecanismo de revocación de certificados autónomo. Y como consecuencia *PACDRE* es un medio para minimizar los riesgos de que una llave de un nodo llegase a ser comprometida por un adversario, porque elimina la posibilidad de que un nodo cuente con certificados vencidos.

6.1 Trabajo a futuro

Las soluciones que se presentaron en este trabajo son teóricas en el sentido que se presentan los protocolos y las primitivas criptográficas que son necesarias para proporcionar los servicios de administración de llaves. La implementación en la práctica no puede ser descartada pero requerirá en algunos casos, de la modificación de códigos, como en el caso de la solución *PACDRE* para incorporar otro repositorio y los mecanismos para el manejo de éstos, así como la creación de un esquema de firma digital para garantizar que los certificados sean autenticados.

REFERENCIAS

[1] SCHNEIER, B. *Applied Cryptography: Protocols, algorithms and source code in C*, John Wiley & Sons, Inc, 1996.

[2] KAUFMAN, C. *et al*. *Network Security: Private Comunication in a PublicWorld*, Prentice-Hall,1995.

[3] MENEZES, A, *et al*. *Handbook of Applied Cryptography*, CRC Press 1997.

[4] Gast, M.S. *802.11 Wireless Networks The Definitive Guide,*O'Reilly,2002.

[5] L. ZHOU, Z. J. HAAS. "Securing Ad Hoc Networks", IEEE Networks, Vol. 13, Issue 6 1999.

[6] H. LUO, S. LU. "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks", Technical Report 2003, UCLA Computer Science Department 2000.

[7] FREEDMAN,A. *Diccionario de Computación*, http:// www.unincca.edu.co/boletin/indice.htm.

[8] SAAB,E. *NetDefence,* http://www.saab.se/future/node2567.asp.

[9] J. M. KAHN, *et al*. "Next Century Challenges: Mobile Networking for *Smart Dust*", ACM Press 1999.

[10] PERKINS, C. *Ad Hoc Networking*, Addison-Wesley 2001.

[11] Royer, E. & Toh, C. "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications,* Vol. 6, No. 2, pp. 46-55, Abril 1999.

[12] J-P. HUBAUX, *et al*. "The Quest for Security in Mobile Ad Hoc Networks", ACM 2001.

[13] TOH, C. *Ad Hoc Mobile Wireless Network.* Prentice Hall, USA, 2002.

[14] GARFINKEL, S. *PGP: Pretty Good Privacy*, O'Reilly & Associates 1995.

[15] Y TSENG, *et al*. "Location Awareness in Ad Hoc Wireless Mobile Networks," IEEE Network Magazine, pp. 46-52, Junio 2001.

[16] S. CAPKUN, *et al*. "Small Worlds in Security Systems: an Analysis of the PGP Certificate Graph", ACM New Security Paradigm Workshop (NSPW), 2002.

[17] S. BASAGNI, *et al*. "Secure Pebblenets", ACM 2001.

[18] D. BALFANZ, *et al*. "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks", Internet Society, Conference Proceeding of NDSS, Conferencia 2002.

[19] J. KATZ, B. SCHNEIER. "Implementation of Chosen-Ciphertext Attacks against PGP and GNuPGP", 9th USENIX Security Symposium, 2000.

[20] J-P. HUBAUX, *et al*. "Self-Organized Public-key Management for Mobile Ad Hoc Networks", IEEE Transaction on mobile computing, Vol.2 , no1 Marzo 2003.

[21] L.M. KORNFELDER, "Toward a Practical Public-Key Cryptosystem", bachelor's thesis, Dept. Electrical Eng., Massachusetts Inst. Of Technology, Cambridge, 1978.

[22] RAM, C.S., MANOJ, B.S., *Ad Hoc Wireless Networks:Architectures and Protocols*, Prentice-Hall,2004.

[23] STALLINGS,W. *Crytography and Network Security: principles and practice*, Prentice-Hall, 1998.

[24] Y. ZHANG, W. LEE. "Intrusion detection in wireless AD HOC networks". Mobile Computing and Networking, pp 275-283, 2000.

[25] O. KACHIRSKI, R. GUHA. "Intrusion detection using mobile agents in wireless ad hoc networks", IEEE Workshop on Knowledge Media Networking, pp153 -158, 2002.

[26] J. LUNDBERG. "Routing security in ad hoc networks".
http://citeseer.nj.nec.com/400961.html

[27] V. KÄRPIJOKI. "Security in ad hoc networks".
http://citeseer.nj.nec.com/karpijoki01security.html

[28] R. RAMANUJAN, *et al*. "Techniques for intrusion-resistant ad hoc routing algorithms (TIARA)". IEEE Military Communications Conference (MILCOM'00), vol.2, Los Angeles, CA, USA, 22-25, 2000.

[29] R. RAMANUJAN, *et al*. "Intrusion-resistant ad hoc wireless networks". MILCOM 2002. IEEE Transaction, Vol. 2 , pp 890 -894, 2002.

[30] P. PAPADIMITRATOS, Z. J. HAAS. “Secure routing for mobile ad hoc networks”. SCS Communication Networks and Distributed Systems Modeling and Simulation (CNDS),2002.

[31] Y-C. HU, *et al*. “SEAD:Secure efficient distance vector routing in mobile wireless ad hoc networks”, IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 2002.

[32] Y-C. HU, *et al.* “Ariadne: a secure on-demand routing protocol for ad hoc networks”. 8th ACM International Conference on Mobile Computing and Networking (MobiCom), 2002.

[33] Y-C. HU, A. PERRIG. “A Survey of Secure Wireless Ad Hoc Routing”, IEEE Security and Privacy, 2004.

[34] Y.HU, *et al.* .”Packet Leashes: A Defense Against Wormhole Attacks in Wireless Ad Hoc Networks”, IEEE INFOCOM 2003, vol 3, pp. 1976-1986, 2003.

[36] B. AWERBUCH, *et al.* “ An On-Demand Secure Routing Protocol Resilient to Byzantine Failures”, ACM Workshop on Wireless Ad Hoc Networks 2002,pp. 21-30,2002.

[37] K. SANZGIRI, *et al.*” A secure Routing Protocol for Ad Hoc Networks”, IEEE ICNP 2002,pp. 78-87,2002.

[38] KOCHHAR,N. *Introduction to Oracle:SQL and PL/SQL:Student Guide Volume 1*”, Oracle Education,1988.

[39] CABALLERO GIL,P.”*Introducción a la Criptografía*”, Alfaomega, 2003.

[40] SINGH. S, *et al*. “Power-Aware Routing in Mobile Ad Hoc Networks”, ACM MOBICOM 1998, pp 181-190,1998.

3. SOLUCIONES PARA LA ADMINISTRACIÓN DE LLAVES EN MANETS

Una vez que se abordaron los problemas y soluciones que presentan las MANETS, se observó que uno de los grandes problemas que éstas enfrentan radica en la administración de llaves y el mecanismo de distribución, emisión y revocación de certificados.

Las soluciones tradicionales para la administración de llaves no pueden ser implementadas en MANETS, debido a su naturaleza que implica movilidad y espontaneidad. El uso de una tercera entidad que proporcione los servicios de administración de llaves, implica que la entidad este dedicada a ese servicio, en las MANETS no es posible tener una entidad dedicada, ya que la mayoría de las veces éstas se forman espontáneamente, pero existen soluciones que nos permiten proporcionar el servicio de intercambio de llaves en forma segura y con ello garantizar que las comunicaciones en la red AD HOC sean seguras, diversos autores han publicado las siguientes soluciones:

- Autoridad certificadora parcialmente distribuida
- Autoridad certificadora completamente distribuida
- Autoemisión de certificados
- Pebblenets
- Identificación demostrativa

No obstante que se han publicado otras soluciones, se han seleccionado aquellas que son más divergentes entre sí y proporcionan un panorama amplio de como implementar un esquema de intercambio de llaves en forma segura en una red AD-HOC.

3.1 ADMINISTRACIÓN DE LLAVES EN MANETS

Como menciona *B. Scheneir* en [1] "La administración de llaves es la parte mas delicada de la criptografía, el diseño de protocolos y algoritmos criptográficos no es una tarea fácil, los criptoanálistas atacan tanto los criptosistemas de llave pública como de llave simétrica concentrando sus esfuerzos en la administración de llaves". En un medio inalámbrico no es la excepción, por lo tanto el implementar un esquema adecuado de administración de llaves en MANETS ayudará a disminuir los riesgos de que un atacante pueda comprometer la red.

Las soluciones tradicionales para el intercambio de llaves no siempre pueden ser implementadas en MANETS, por lo que es necesario adaptarlas de acuerdo a los requerimientos de éstas.

A continuación se dan conocer las soluciones consideradas las más adecuadas para garantizar que la autenticación de los nodos de la red, se realice en forma segura con la ayuda de un esquema de administración de llaves.

Cada una de ellas presenta protocolos y primitivas criptográficas que son necesarias para proporcionar el servicio de administración de llaves. Se retoma el uso de autoridades certificadoras.

3.2 AUTORIDAD CERTIFICADORA (CA) PARCIALMENTE DISTRIBUIDA

Está solución es propuesta por *Zhou y Hass.* [5] Utiliza un esquema de secretos compartidos (*k,n*) [34] para distribuir los servicios de autoridad certificadora a un conjunto de nodos servidores especializados. Cada uno de ésos nodos tiene la capacidad de generar certificados parciales, utilizando su *k-parte* compartida del certificado firmado por la llave $sk_{CA.}$ Para obtener un certificado válido se deben combinar las *k-partes* de los certificados parciales.

La solución es adecuada para MANETS planeadas y con una duración de largo plazo. El esquema de encripción que emplea ésta solución es de llave pública, por lo tanto cada nodo debe realizar los cálculos pertinentes para el proceso de encripción/decripción, en el caso de MANETS que utilicen dispositivos de red con 802.11 el proceso anterior puede realizarse sin ningún problema. Se asume que el subconjunto de nodos puede jugar el rol de servidor especializado.

En esta solución se propone que el sistema puede contener cuatros tipos de nodos, los cuáles son: cliente, servidor, mezclador y negociador. Los nodos clientes, son los usuarios "normales" de la red mientras que los nodos servidor y mezclador son parte de la CA. Los nodos servidores, son los responsables de generar certificados parciales y almacenarlos en una estructura de directorio que permita a los nodos clientes solicitar certificados de otros nodos. Los nodos mezcladores, los cuáles son también nodos servidores, son los responsables de convertir certificados parciales en

[34] Técnica para dividir un secreto, D, en *n* partes, de tal manera que para recuperarlo se requieren *k* partes, donde k<n. Para detalles consultar la sección 1.3.5

certificados válidos. La única entidad en el sistema que tiene conocimiento del certificado "absoluto" firmado por la llave sk_{CA} es el nodo denominado negociador.

Cada nodo de la red tiene un par de llaves: pública y privada, el nodo negociador tiene dos responsabilidades, la de emitir el certificado inicial para la llave pública de los nodos y la de distribuir la llave pública pk_{CA} de la CA a todos los nodos con el objetivo de verificar los certificados.

La CA tiene un par de llaves: publica y privada denominadas pk_{CA}/sk_{CA} de las cuáles la llave pública es conocida por todos lo nodos. La llave privada sk_{CA} se comparte entre los nodos servidores conforme al esquema de secretos compartidos de *Shamir*.

En el proceso de emisión de certificados, un nodo que quiera integrarse a la red, necesita obtener un certificado válido del nodo negociante antes de que pueda unirse a ésta, al mismo tiempo el nodo solicitante debe ser abastecido del certificado de la CA junto con los parámetros requeridos.

La figura 3.1 ilustra los diferentes componentes del sistema.

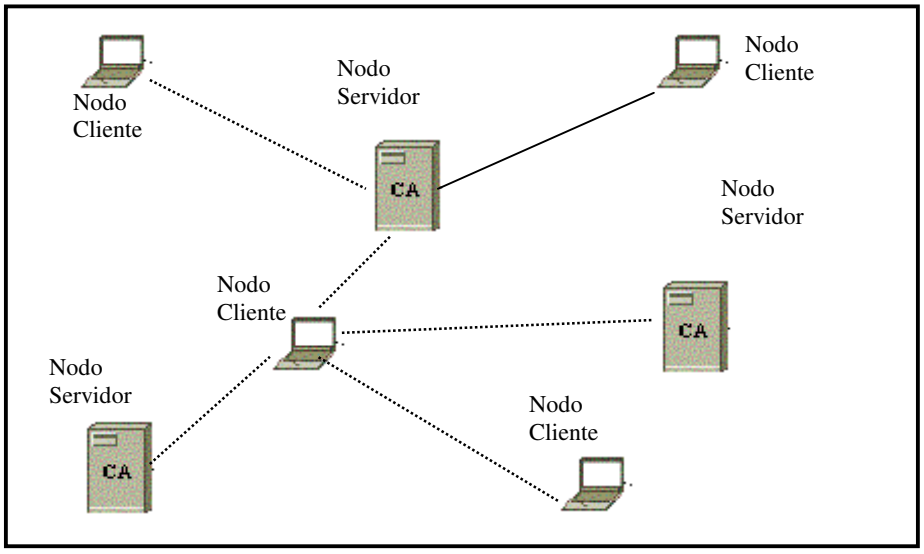


Figura 3.1. Arquitectura del sistema que contiene tres nodos servidores de los cuáles uno de ellos tiene el rol de ser combinado o mixto.

3.2.1 ENVÍO, RENOVACIÓN Y RECUPERACIÓN DE CERTIFICADOS

Antes de que un nodo pueda asociarse a la red debe obtener un certificado válido del nodo negociador *off-line*[35], al mismo tiempo al nodo se le debe suministrar el certificado de la CA, con los parámetros requeridos.

Los certificados tienen validez por un periodo de tiempo determinado, por lo que deben ser renovados antes de que éste expire. Para realizar el proceso de renovación de su certificado, un nodo debe crear una solicitud ante un mínimo de k nodos servidores. Si la solicitud es otorgada,

[35] Por sus siglas en inglés fuera de línea.

cada uno de ésos *k* nodos, genera un certificado parcial con una nueva fecha de expiración. Una vez realizado esto, los certificados parciales son enviados a un nodo mezclador, que puede ser cualquiera de los *k* nodos servidores, cuya función es combinar los certificados parciales.

En caso de que alguno de los servidores llegue a ser comprometido por un adversario, el servidor comprometido generará un certificado parcial inválido, el cuál será enviado al nodo mezclador que no tiene la capacidad de verificar si los certificados son válidos o no, por lo que el certificado producido por éste también será inválido. Este tipo de ataque puede representar una negación de servicio a la red AD HOC y puede ser prevenido si el nodo mezclador verifica la validez del certificado antes de aceptarlo, si detecta que el certificado es inválido éste solicitará un nuevo conjunto de certificados parciales hasta obtener un certificado válido.

Con respecto a la recuperación de certificados, los nodos servidores son los responsables de almacenar los certificados de todos los nodos en las MANETS. Esto permite que cualquier nodo pueda solicitar la llave pública de otros nodos, para ello se tiene que solicitar el certificado adecuado a cualquiera de los nodos servidores. Este servicio requiere que todos los nodos registren su certificado con los servidores donde inicialmente se unieron a la red. Los servidores tienen que contar con un mecanismo de sincronización con sus directorios de certificados para el momento que se realice una renovación o actualización de algún certificado.

3.2.2 MANTENIMIENTO DEL SISTEMA

El mantenimiento de la CA radica en dos acciones fundamentales, la expedición inicial las *k-partes* y la actualización proactiva de las *k-partes*, para protección contra adversarios móviles. [5] La actualización de la *k-partes* permite al sistema cambiar su configuración dentro del esquema de secretos compartidos, por ejemplo si tiene inicialmente el parámetro (3,8) puede cambiarlo a (2,5).

En el proceso de inicialización de la red, el nodo mezclador genera *n* partes de la llave privada sk_{CA} de la CA y las envía a cada uno de los *n* servidores de la red ad hoc. En intervalos periódicos el servidor actualiza sus *k-partes* de la llave privada sk_{CA} de la CA. Cuando inicia la fase de actualización cada servidor genera en forma aleatoria los valores del esquema de secretos compartidos (*n,k*) y distribuye la *k-parte* a los otros servidores, cada uno de ésas *k-partes* son denominados como sub-partes. Cada servidor tiene *n* sub-partes de diferentes servidores los cuales son añadidos a sus *k-partes* anteriores tomando de ellas su nueva *k-parte* [36] actualizada. [5]

3.3 AUTORIDAD CERTIFICADORA (CA) COMPLETAMENTE DISTRIBUIDA

Está solución es propuesta por Luo y Lu. [6] Utiliza un esquema de secretos compartidos (*k,n*)[37] para distribuir un certificado RSA a todos los nodos en la red AD HOC. Similar a la solución de

[36] Para detalles del proceso del esquema de secretos compartidos ver la sección 1.3.5
[37] IDEM

certificados parciales está solución es adecuada para MANETS planeadas y con una duración de largo plazo. Los nodos en la solución deben ser capaces de efectuar encripción de llave pública, en el caso de una implementación con dispositivos de red 802.11 el proceso de encripción/decripción puede realizarse sin ningún problema porque cuenta con los campos requeridos para manejar el mecanismo de encripción. La característica de ésta solución es que el servicio de administración de certificados, es distribuido entre todos los nodos cuando éstos se unen a la red, por lo tanto no es necesario elegir a ningún nodo servidor especializado. Pero cuando se inicia la red AD HOC se requiere de una entidad que funja como autoridad administrativa para distribuir los certificados iniciales.

Las capacidades de la CA, como se mencionó, se distribuyen a todos los nodos de las MANETS, como lo muestra la figura 3.2. Cualquier operación que requiera la llave privada de la CA's denominada como sk_{CA} solo puede ser realizada por una coalición de k nodos como lo indica el esquema de secretos compartidos.

Los servicios que la CA ofrece pueden agruparse en: servicios de certificados relacionados y servicios de mantenimiento del sistema. Los servicios de certificados relacionados incluyen la renovación y revocación de certificados. Los servicios de mantenimiento del sistema incluyen la incorporación de nodos en la autoridad certificadora, por ejemplo cuando se les proporciona a los nodos la *k-parte* de la llave privada sk_{CA} de la CA. Esté servicio es conocido como inicialización. El mantenimiento del sistema, también incluye la actualización proactiva de las *k-partes* de la llave privada de la CA, para prevenir que ésta llegase a ser comprometida por un adversario. Esté servicio se denomina actualización de las *k-partes* de la llave privada sk_{CA} de la CA. [6]

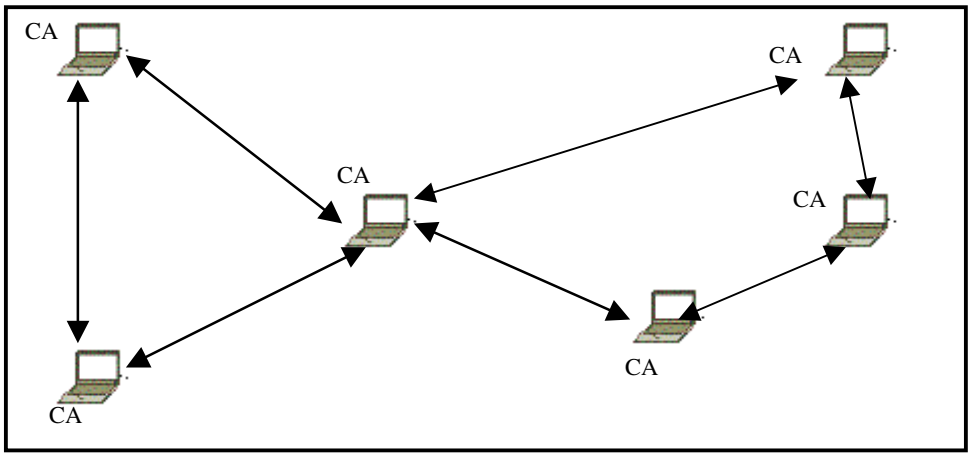


Figura 3.2. Autoridad certificadora completamente distribuida, se muestra que todos los nodos en la red son iguales y pueden manejar una k-parte de la llave de la CA.

La disponibilidad del servicio se basa en el supuesto de que cada nodo debe tener un mínimo de k nodos vecinos (que estén a un solo salto) y que los nodos suministren un certificado válido antes de que se puedan unir a la red. El sistema proporciona los servicios para actualizar los certificados iniciales.

3.3.1 MANTENIMIENTO DEL SISTEMA

El mantenimiento es requerido para manejar a los nuevos nodos que se van uniendo a la red y para proteger el servicio contra posibles atacantes que intenten comprometer el servicio de la CA.

En la primera fase denominada en [6] como auto-arranque del sistema, la autoridad administrativa de las MANETS inicializa los primeros *k* nodos. Con el proceso de inicialización se les suministra a los nodos su propio certificado denominado $cert_{id}$, el certificado de la CA se denomina $cert_{CA}$ y sus *k-partes* de la llave secreta sk_{CA} de la CA.

La figura 3.3 ilustra dicho proceso.

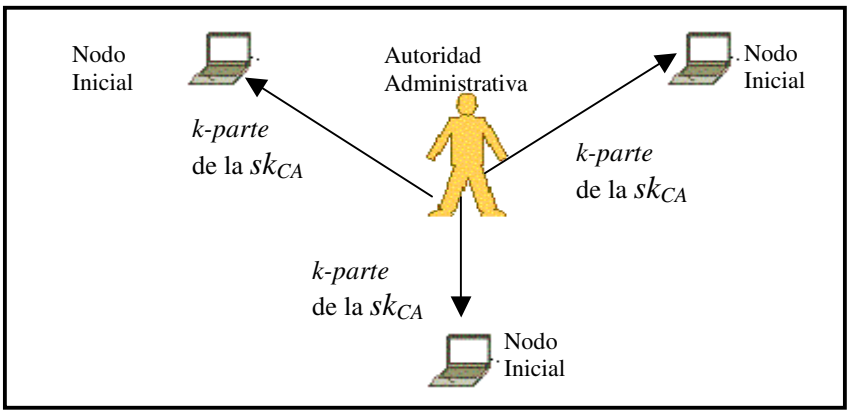


Figura 3.3. Fase de arranque de inicialización. La autoridad administrativa proporciona a cada nodo su k-parte de la llave.

La autoridad administrativa es la única entidad que tiene acceso al certificado firmado por la llave sk_{CA} y con ello puede emitir certificados iniciales como se menciono anteriormente. Para inicializar los primeros *k* nodos de la red se necesita realizar el siguiente procedimiento: [6]

1. La autoridad administrativa genera el polinomio:

   $f(x) = a_0 + a_1 x + ... + a_{k-1} x^{k-1}$ mientras: $a_0 = sk_{CA.}$

2. Cada *k* nodos inicializados se identifican por: id_i, $i=1,\ldots k$ que es enviado en forma segura con el polinomio:

   $Si = f(id_i) \bmod N.$

3. La autoridad administrativa envía un mensaje brodcasts firmando los coeficientes del polinomio: $\{g^{a_0},..., g^{a_{k-1}}\}$, posteriormente elimina el polinomio.

4. Cada nodo verifica que el valor compartido recibido (del paso 2) es válido, comprobando que:

$$g^{S_j} = g^{a_0} * (g^{a_q})^{id_j} * ... (g^{a_{k-1}})^{id_j^{k-1}}$$

Después de la inicialización de los primeros *k* nodos, la autoridad administrativa es la única entidad responsable del registro, inicialización y certificación inicial de cualquier otro nodo que se quiera unir a la red. [6]

Cualquier nodo que se haya unido a la red se incorpora dentro de la CA y con ello tendrá su propia *k-parte* del certificado de la CA firmado por la llave $sk_{CA.}$ Cabe mencionar que debido a las limitaciones en cuanto al alcance de la longitud de onda del protocolo 802.11, la autoridad administrativa no puede abarcar todas las MANETS en su conjunto, por lo que el mecanismo de distribución compartida necesita ser manejado por los nodos que ya estén inicializados como se muestra en la figura 3.4

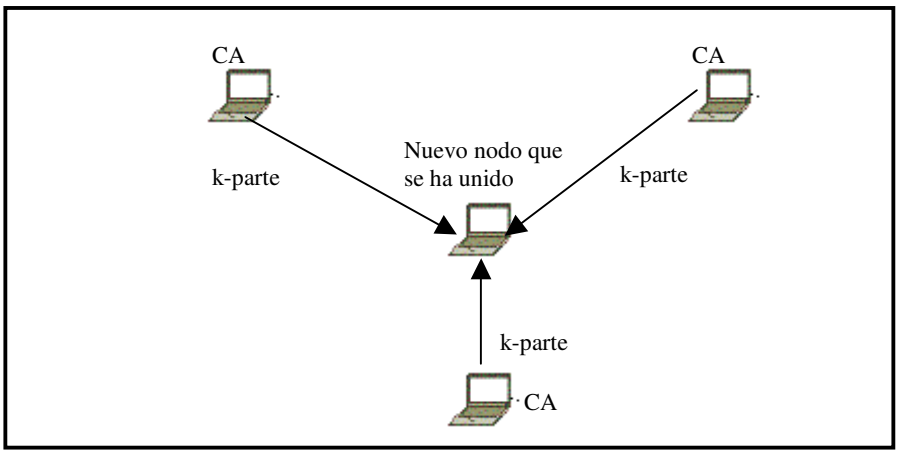


Figura 3.4. Inicialización durante la fase operacional, se aprecia que un nuevo nodo se ha unido a la red. Los nodos vecinos ya pertenecen al servicio de CA y generan una nueva k-parte e inicializan al nuevo nodo.

Para proteger la confidencialidad de los secretos compartidos dentro de la coalición de nodos, los nodos mezclan o combinan las *k-partes* antes de enviarlas al nuevo nodo asociado. El proceso de combinar las *k-partes* asume que *p* es el nodo que se va a inicializar y consiste en:[6]

1.- El nodo *p* localiza una coalición *B* de *k* nodos, que se denotan como: $B = \{id_1, ..., id_k\}$ y envía un mensaje *broadcasts* solicitando la inicialización.

2.- Cada nodo en la coalición verifica el certificado, $cert_p$, del nodo *p* y comprueba que no este revocado. Si se comprueba que el certificado es revocado la solicitud es denegada.

3.- Dentro de la coalición, cada par de nodos *{i,j}* necesitan acordar un factor de mezcla o combinado que se denota como: $d_{i,j.}$ Uno de los nodos genera el factor combinado, lo encripta con la llave pública del otro nodo y lo firma. También genera y firma una prueba o evidencia del factor combinado $g^{d_{ikj}}$. La prueba o evidencia es necesaria para poder detectar e identificar cualquier comportamiento extraño de la coalición de nodos, que puedan generar un valor inválido. Los factores combinados así como las pruebas o evidencias son enviados al nodo solicitante *p*.

4.- El nodo *p* distribuye los factores combinados y las pruebas o evidencias que ha recibido a toda la coalición de nodos.

5.- Cada nodo *j* en la coalición, genera la k-parte compartida: $S_p^j = S_j * l_{id_j}(id_p)$ del nodo *p* y la mezcla utilizando los factores recibidos en el paso anterior. La mezcla parcial de las *k-partes* se denominan como: $\bar{S}_p^j$ que se generan de la siguiente forma:

$$\bar{S}_p^j = S_p^j + \sum_{i=1, i\neq j}^{k} [sign(id_i - id_j) * d_{i,j}] \bmod N$$

donde sign(x)= $\begin{Bmatrix} -1, x \leq 0 \\ 1, x > 0 \end{Bmatrix}$

6.- Cada nodo de coalición *j* envía su k-parte de las mezclas parciales $\bar{S}_p^j$ hacia el nodo *p.*

7.- El nodo *p* verifica cada una de las *k-partes* de las mezclas parciales recibidas $\bar{S}_p^j$ comprobando que:

$$g^{\bar{S}_p^j} = g^{S_p} \prod_{i=1, i\neq i}^{k} (g^{d_{i,j}})^{sign(id_i - id_j)}, \text{donde } g^{S_p} = g^{a_0} * (g^{a_1})^{id_p} * ... * (g^{a_{k-1}})^{id_p^{k-1}}$$

Si la verificación del nodo *p* falla, entonces borra las *k-partes* de las mezclas no válidas y emite una nueva solicitud de inicialización, excluyendo los nodos inválidos que fueron detectados. El nodo *p* revoca los certificados inválidos.

8.- Si todas las *k-partes* de las mezclas parciales son correctas, el nodo *p* puede obtener su nueva k-parte adicionando las *k-partes* de las mezclas parciales, para ello deberá realizar la siguiente sumatoria:

$$S_p = \sum_{i=1}^{k} S_p^j \bmod N$$

Después de ser inicializado con sus *k-partes* del certificado firmado por la llave sk_{CA}, el nodo *p* será parte de la CA y podrá participar en los servicios que atañen a las autoridades certificadoras, como son: la renovación de certificados, revocación e inicialización de cualquier nodo que quiera unirse a la red con su k-parte del certificado $sk_{CA.}$

3.3.2 ACTUALIZACIÓN DE LAS *K-PARTES* DE LA LLAVE PRIVADA SK_{CA} DE LA CA

El propósito de usar secretos compartidos en está solución, es la de prevenir que un adversario pueda comprometer algún(os) nodo(s) de las MANETS, y con ello sea capaz de reconstruir el secreto compartido, que en esté caso sería el certificado firmado por la llave $sk_{CA.}$ A continuación se describe el mecanismo que se emplea para prevenir que un atacante pueda reconstruir el secreto compartido.

El proceso de actualización de las *k-partes* está ligado al tiempo de vida de la red, el cuál está divido en periodos de tiempo, en donde cada periodo está compuesto por dos fases: la fase de actualización de las *k-partes de la llave privada* sk_{CA} *de la CA*[38] y la fase operacional. A continuación se presentan cada una de ellas.

a) Actualización de las *k-partes*, en está fase se realiza el siguiente proceso:

1. Generación colaborativa del polinomio de actualización $f_{update}(x)$.

2. Distribución del polinomio de actualización a todos los nodos de la red.

3. Evaluación distribuída de la actualización de las k-partes:

   $\bar{S}_p = f_{update}(id_p)$ de todos los nodos *p*.

   Cuando inicia la fase de actualización de las *k-partes*, cada nodo comienza el proceso de actualización con una probabilidad *1/n* en donde *n* se conoce como el estimado del total de nodos en la red. Cuando un nodo inicia la actualización de las *k-partes* lo primero que hace es localizar la coalición de *k* nodos que generan el polinomio de actualización: $f_{update}(x) = b_1 x + b_2 x^2 \ldots + b_{k-1} x^{k-1} \bmod N$.

   Cada coeficiente de los polinomios es encriptado, firmado y enviado a la red AD HOC. En este momento cada nodo de la red ha recibido: $\{ E_{pk_{CA}}(b_1),\ldots,E_{pk_{CA}}(b_{k-1}) \}$ el cual es autenticado para verificar las firmas. Esta medida previene que un atacante pueda enviar a la red un polinomio de actualización falso.

   Una vez que han sido enviados los parámetros anteriores, cada nodo *p* en la red puede generar su actualización de las k-partes: $\bar{S}_p = f_{update}(id_p)$.

   Cada nodo de la coalición devuelve el valor de la actualización parcial de las *k-partes* al nodo solicitante *p* quien a su vez las añade para obtener la actualización de las *k-partes* completa que se define como: $\bar{S}_p$. Una vez que se completa lo anterior, se obtiene la actualización de la *k-partes* de la sk_{CA} de la CA.

b) Fase operacional, en está fase los nodos pueden renovar sus certificados y solicitar la inicialización de las *k-partes*, el proceso de renovación de certificados se detalla en la sección 3.3.3.

En la figura 3.5 se muestran la fase operacional y la fase de actualización de las k-partes, el tiempo de vida de la red está divido en periodos de tiempo, en donde cada periodo contiene dos fases, la fase operacional y la fase de actualización compartida.

[38] En adelante a la actualización de las k-partes de la llave privada sk_{CA} de la CA será denominado como: actualización de las k-partes

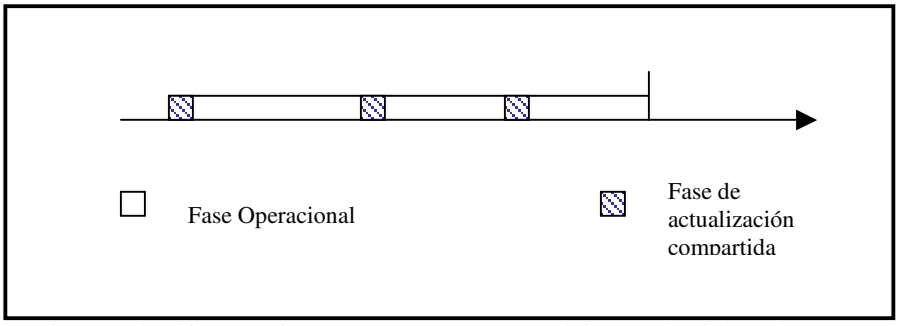


Figura 3.5. Diferentes fases que se presentan en el tiempo de vida en las MANETS .

3.3.3 EMISIÓN Y RENOVACIÓN DE CERTIFICADOS

En está solución se asume que todos los nodos han sido inicializados, registrados y emiten certificados válidos antes de unirse a la red.

La CA distribuida nunca emite nuevos certificados, únicamente los administra una vez que han sido creados.

La responsabilidad de inicializar, registrar y certificar a los nuevos nodos pertenece a la autoridad administrativa.

Los certificados tienen validez por un tiempo limitado, por lo que deben ser renovados antes de que expiren. Cuando un nodo *p* renueva su certificado *cert* debe solicitar una renovación de certificados a la coalición de nodos vecinos (que estén a un solo salto). Para ello cada nodo *i* en la coalición verifica primero que el certificado anterior no haya expirado y que no este revocado. Si los nodos de la coalición acuerdan atender la solicitud, generarán un nuevo certificado parcial $cert_i$ y se lo enviarán al nodo *p*. El nodo *p* combinará los certificados parciales para obtener su certificado actualizado $cert_{updated}$.

Los detalles del proceso se describe a continuación: [6]

1. El nodo *p* que desea actualizar su certificado envía una solicitud de renovación a todos los nodos vecinos (que estén a un solo salto). Ver figura 3.6.

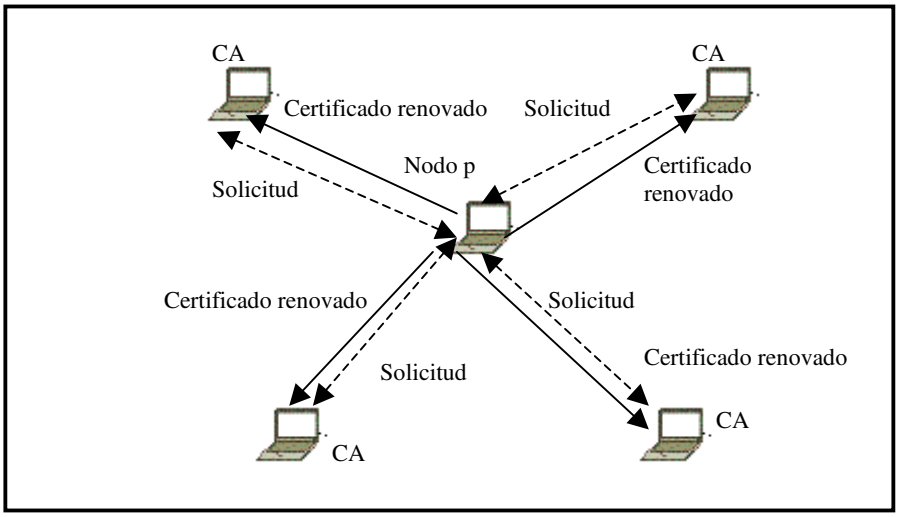


Figura 3.6. Proceso que realiza el nodo *p* para solicitar la renovación de su certificado.

2. Cada nodo que recibe la solicitud, verifica que el certificado *cert* del nodo solicitante *p* no haya expirado o este revocado. Si el certificado es válido y el nodo receptor *i* decide atender la solicitud, genera el certificado parcial $cert_i$ para utilizarlo con su *k-parte* del certificado de la CA firmado por la llave sk_{CA}. El certificado parcial se genera conforme a: $cert_i = (cert)^{S_i} \ mod \ N$ donde S_i es la i-esima *k-parte* de sk_{CA} del nodo.

3. Cada nodo *i* que esté atendiendo la solicitud, genera en forma aleatoria *u* y calcula: $A_1 = g^u$ y $A_2 = cert^u$, $c = Hash\ (g^{Si}, cert_i, A_1, A_2)$ y $r = u - c * S_i$. El valor de A_1 A_2 y *r* son usados para verificar el certificado parcial generado $cert_p$.

4. Cada nodo *i* devuelve $cert_i$, *A1, A2*, y *r* al nodo solicitante *p*.

5. El nodo *p* elige los certificados parciales recibidos y verifica cada uno de ellos asegurándose que: $g^r * (g^{S_i})^c = A_1$ y que el $cert^r * (cert_i)^c = A_2$. Una vez realizado lo anterior el nodo *p* puede generar *c* para aplicar la misma función *hash* que se utilizo en el paso 3. Del paso 4 se conoce A_1, A_2 y *r* y se puede calcular: $g^{S_i} = g^{a_0} * (g^{a_1})^{id_i} * ..., * (g^{a_{k-1}})^{id_i^{k-1}}$

6. Si fallará la verificación de algún certificado parcial, el certificado del nodo que generó el certificado inválido es revocado y se escoge un certificado parcial de los que son recibidos (paso 4) y se válida (paso 5). Si se reciben menos *k* certificados válidos la renovación falla.

7. El nodo *p* combina los *k* certificados parciales válidos para obtener un candidato a certificado conforme a:

$$cert_{updated} = \prod_{i \in B} (cert_i)^{l_{id_i}(0)}$$
$$= (cert)^{i \in B} = (cert)^{t*N+sk_{CA}}$$

8. El certificado actualizado debe cumplir con: $cert_{update} = cert^{sk_{CA}} \neq (cert)^{t*N+sk_{CA}}$. Para obtener el certificado actualizado, que se aplica a un candidato a certificado, se emplea el algoritmo llamado "*k-bounded offsetting*", el algoritmo se presenta a continuación:

1. $Y_0 = cert_{updated}$
2. $Z = cert^{-N} \bmod N$
3. j=0,W=1
4. while j≤ do
5. $Y = Y_0 * W \bmod N$
6. $W = W * Z \bmod N$
7. If($cert \equiv Y^{pk_{CA}} \pmod N$) then
8. Actualizar certificado encontrado devolver Y
9. end if
10. j=j+1
11. end while

3.3.4 REVOCACIÓN DE CERTIFICADOS

El mecanismo de revocación de certificados, se basa en el supuesto de que todos los nodos monitorean el comportamiento de sus vecinos cercanos (vecinos que están a un solo salto) y que mantienen sus propias listas de revocación de certificados. Con el protocolo 802.11 el monitoreo puede llevarse a cabo sin que presente problemas en el manejo de mensajes. Si un nodo descubre que uno de sus vecinos tiene un comportamiento "extraño" añade el certificado del nodo "malicioso" a la lista de revocación de certificados (*CRL*)[39] y envía una "denuncia" o "acusación" contra el nodo.

Cualquier nodo que reciba una denuncia, debe verificar su *CRL* para ver si ésta no es originada por un nodo al cual se le ha revocado su certificado, si el certificado del denunciante ha sido revocado la denuncia es ignorada. Por otra parte, si la denuncia es originada por un nodo válido, el nodo denunciado es marcado como sospechoso. Cuando se recibe el límite de *k* acusaciones contra el mismo nodo, el certificado del nodo acusado es revocado. [6]

La figura 3.7 muestra la secuencia de eventos que suceden cuando el nodo *D* (malicioso) es detectado por dos nodos el *B* y el *C*. Primero, los nodos *B* y *C* revocan el certificado del nodo *D* y posteriormente envían la denuncia. Una vez que se han recibido las denuncias, el nodo *A* las válida verificando en su propia *CRL* para asegurarse que el certificado del nodo *B* y *C* no han sido revocados.

---

[39] Por sus siglas en inglés Certificate Revocation Lists.

Dado que el limite de denuncias que se han recibido (en éste caso *k*=2) el nodo *A* también agrega al nodo *D* a su *CRL*. El nodo *A* retransmite las acusaciones del nodo *B* y *C*. El nodo *E* y *F* reciben las acusaciones y realizan el mismo proceso que el nodo *A*.

El nodo *A* retransmite las denuncias de los nodos (*B* y *C)* y las envía a la red. Las denuncias necesitan ser enviadas a un número limitado de saltos, la razón de esto es que la validez de los certificados es por tiempo limitado, que se denota como: t_{cert}.

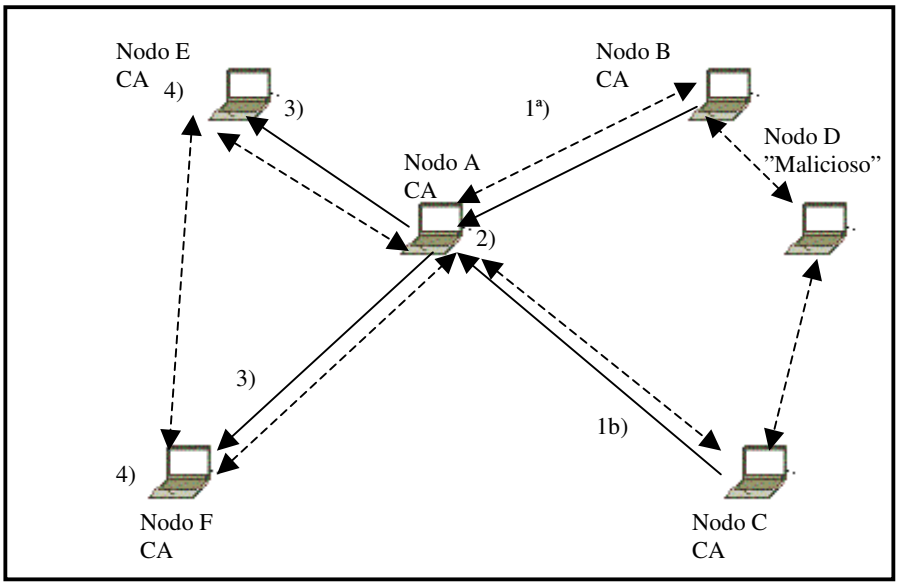


Figura 3.7. Revocación de certificados y mantenimiento distribuido de las listas de revocación de certificados.

Para determinar el número de saltos que una acusación debe ser enviada, se emplea la siguiente relación: [6]

$$TTL \geq \left\lceil \frac{t_{cert} * 2s_{\max}}{d} \right\rceil$$

Donde *TTL* es el número de saltos que la acusación debe ser enviada, s_{max} es la velocidad máxima de un nodo y *d* es el rango de transmisión del primer salto del medio inalámbrico utilizado. Un nodo que tiene su certificado revocado, por cierto nodo, puede ser removido de la *CRL*. Esto puede ocurrir si uno de los denunciantes está también dentro de la *CRL*. Si éste nodo envía una denuncia, se nulifica y si está trae el número de denuncia contra un nodo culpable esta revocación también es nulificada. En la figura 3.7 los certificados del nodo *D* fueron revocados por el nodo *A* debido a que fue denunciado por los nodos *B* y *C*. Si después uno de los denunciantes, por ejemplo el nodo *B*, es culpable y su certificado revocado, el nodo *A* remueve el nodo *D* de su *CRL*.

3.4 AUTOEMISIÓN DE CERTIFICADOS

Esta solución es propuesta por *Hubaux* [12] y utiliza el esquema de administración de llave pública similar a *PGP*[40] [14], dejándole a los usuarios la responsabilidad de emitir sus propios certificados, sin involucrar a ninguna autoridad certificadora.

Difiere de las soluciones de llave pública que se han presentado en este trabajo al no involucrar a ninguna tercera autoridad administrativa. El objetivo de está solución es que trabaje en MANETS espontáneas en dónde no exista ningún contacto previo entre los nodos. La propuesta se basa en encripción de llave pública y requiere que los nodos tengan capacidad de cómputo para realizar cálculos criptográficos. En el caso de dispositivos con 802.11 lo anterior no representa ningún problema, porque poseen capacidad para manejar criptografía de llave pública.

Al igual que *PGP,* el problema de distribución y entrega de llaves se extiende en está solución. *PGP* no cuenta con una solución de infraestructura de llave pública (*PKI*), por lo tanto las llaves públicas no están certificadas por ninguna tercera entidad confiable, por ejemplo una CA. En vez de esto cada usuario tiene la capacidad de certificar las llaves públicas de otros usuarios, por lo que es responsabilidad de cada usuario determinar la confianza que tendrá un certificado.

La figura 3.8 muestra un ejemplo del funcionamiento de *PGP*. Beto expide su certificado a Carlos, de esta forma declara que pk_{Carlos} es realmente la llave pública de Carlos. Alicia también expide su certificado a Beto, indicando que pk_{Beto} es la llave de Beto. Alicia confía que Beto no ha emitido ningún certificado falso, por lo tanto Alicia siempre confiara en cualquier certificado emitido por él. De esta manera teniendo los certificados $cert_{Alicia,Beto}$ y $cert_{Beto,Carlos}$, Alicia puede verificar que la llave pk_{Carlos} es autentica y también comunicarse en forma segura con Carlos, aún cuando entre ellos no exista ningún vínculo previo.

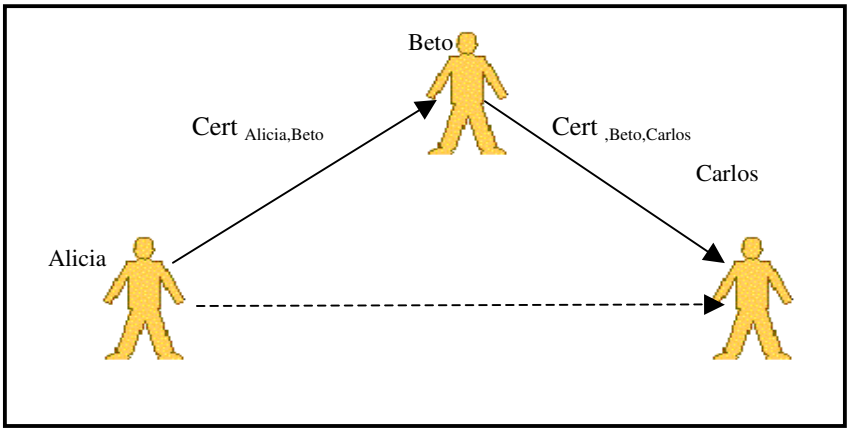


Figura 3.8. Ejemplo de PGP.

En *PGP* existen servidores de llaves públicas, que son usados para distribuir certificados, pero en MANETS no puede existir una entidad que conjunte a todos los certificados y los distribuya, la solución propone otorgarles a los usuarios la responsabilidad de distribuir y almacenar los certificados. Cada usuario tendrá algunos certificados que podrá emitir.

La figura 3.9 ejemplifica el proceso de distribución de certificados, en este caso Alicia desea obtener el certificado $Cert_{Capy,Alicia}$, para ello deberán encontrar una cadena de certificados con sus

[40] Por sus siglas en inglés Pretty Good Privacy

vecinos cercanos, utilizando los certificados que están almacenados en sus repositorios. Para obtener el certificado que Alicia requiere, contacta a su vecino cercano Capy, la cuál no cuenta con el certificado pero envía una petición a su vecino cercano Beto que si cuenta con el certificado en su repositorio, que se lo envía a Capy y ella a su vez lo reenvía a Alicia.

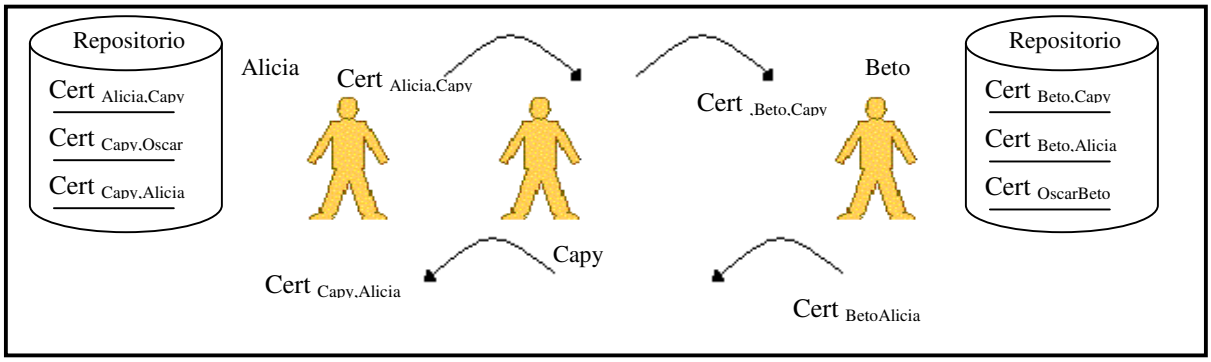


Figura 3.9. Cadena de certificados locales para autenticar un usuario.

El algoritmo de selección propuesto por los autores es "Shortcut Hunter" que esta basado en un fenómeno conocido como "*small-world*" [16] y que nos da una garantía probabilística de obtener una cadena de certificados como se muestra en la figura 3.9.

3.4.1 "*SMALL-WORLD*"

El comportamiento teórico del fenómeno "small-world" [16] está fuera del alcance de esté trabajo, pero se retoman algunos conceptos básicos de su funcionamiento para comprender el funcionamiento, del algoritmo "*Shortcut Hunter*". Los grafos "small world" se pueden definir como: "Grafos que tienen en promedio un diámetro pequeño y que están muy agrupados".

Un ejemplo de éstos, es un principio conocido como "seis grados de separación" [16]. Dicho principio nos dice que dos personas se conocen mutuamente hasta por un máximo de seis personas. En la figura 3.10 se muestra un ejemplo, en este caso Alicia quiere contactar a Beto, como lo indica la línea punteada, ella solo conoce a Carmen y por medio de ella puede conocer a Edgar, a Hector, a Mary, a Lulú y por último contactará a Beto, con lo cuál se cumple el principio de seis grados de separación.

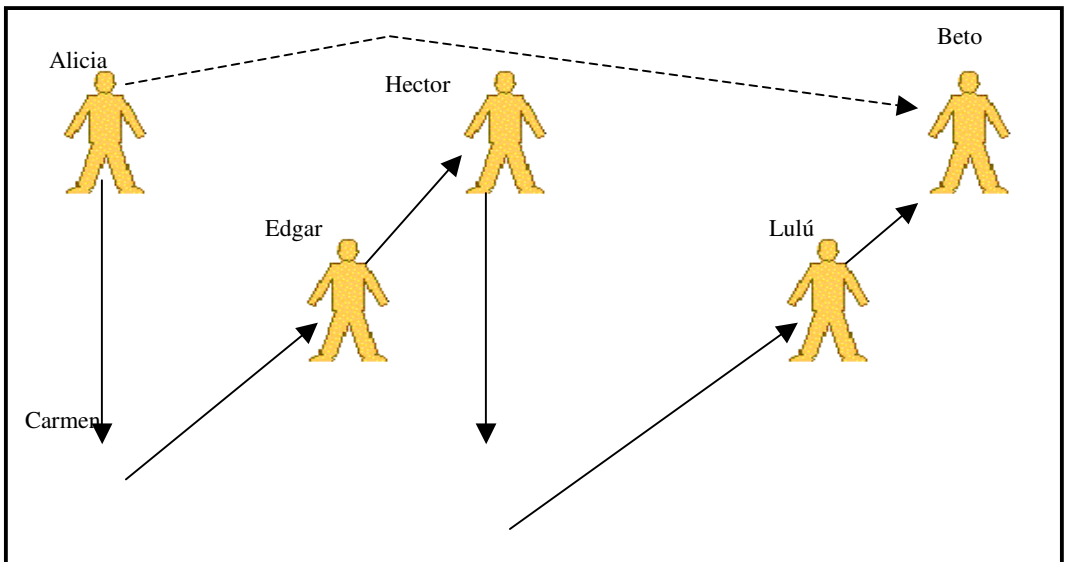


Figura 3.10. Ejemplo del fenómeno “small-world”.

3.4.2 ALGORITMO “*SHORTCUT HUNTER*”

El algoritmo “*Shortcut Hunter*” es el algoritmo propuesto por *Hubaux* [12] para la selección de certificados, describe que cada usuario puede almacenar un número limitado de certificados, los cuales pueden clasificarse como:

1. Certificados emitidos por el usuario.

2. Certificados enviados hacia el usuario.

Cabe señalar que un usuario conoce los certificados que ha emitido y puede almacenarlos en su repositorio local de certificados.

El algoritmo modela los certificados que son enviados y los usuarios como un grafo “small-world”, el certificado se representa por una arista dirigida. En la figura 3.11, se ejemplifica el proceso que realiza el algoritmo, en este caso Alicia envía un certificado a Beto y se representa como una arista dirigida del vértice representado por Alicia al vértice representado por Beto. Cuando Alicia le expide un certificado a Beto, ella almacena el certificado en su repositorio local y le informa a Beto que le ha enviado un certificado. De esta forma todos los usuarios obtienen los certificados de la clasificación 1 y 2.

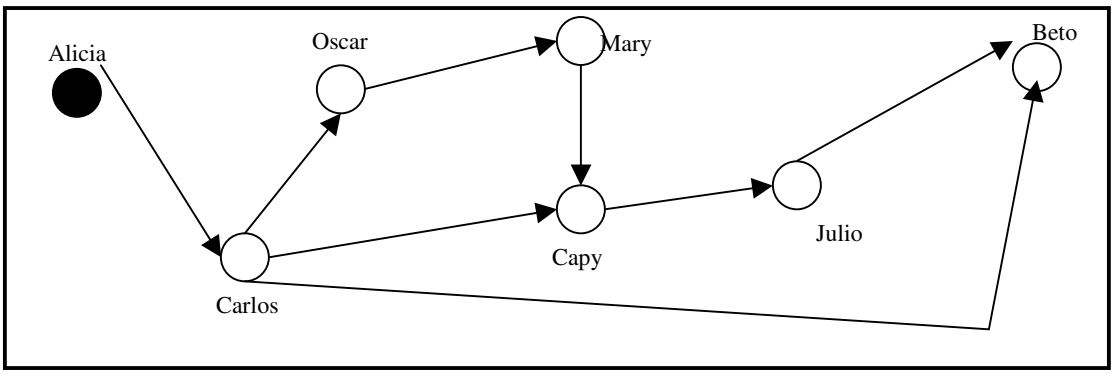


Figura 3.11. Grafo que muestra la emisión de certificados aplicando el algoritmo Shortcut Hunter.

El algoritmo consiste de dos iteraciones. En la primera iteración se selecciona la cadena del certificado a emitir y en la segunda, la cadena del certificado a recibir. La selección del certificado se determina inspeccionando el número de “atajos”que tiene el certificado. El algoritmo se presenta a continuación: [12]

1) V(S)={u},E(S)=Ø, N= Ø,w=u,i=0
2) T={(w,z)ε E(G) ; z∉ V(S) y z∉ N}
3) Si *T= Ø* entonces

a. Si *w=u* entonces ve a 9
b. Agrega *w* a *N*
c. Borrar (*v,w*) de *E(S*) y *w* de *W(S)*
d. *w=v,i=i-1*
e. Ir a 2

4) Seleccionar (*w,z*) εT
5) Si c=0 seleccionar (w,z) ε*T*
6) Añadir (*w,z*) a *E(S)* y *z* a *V(S)*
7) *w=z, i=i+1*
8) Si *i<s* entonces ir a 2
9) Devolver el certificado representado por *V(S)* y *E(S)*, y finalizar.

Del algoritmo anterior se denota a:

- *V(S)* como los vértices en el sub-grafo seleccionado.
- *E(S)* como las aristas correspondientes.
- *E(G)* a todas las aristas del grafo "small-world".
- *u* como el vértice que esta ejecutando el algoritmo.
- *s* como un parámetro específico de la longitud máxima de la cadena del certificado seleccionado.
- *N* como los vértices que han sido procesados pero no han sido seleccionados.

Para ejemplificar el algoritmo, se retoma el grafo de la figura 3.11, en donde Alicia es el usuario que ejecuta el algoritmo, la longitud máxima de la cadena de certificados es: *s=3.* Dado que Alicia tiene el certificado $cert_{Alicia,\ Carlos}$, este es seleccionado en la primera iteración, después se verifican los certificados que son emitidos por Carlos. Tanto Beto como Capy tienen tres "atajos" y se escoge en forma aleatoria uno de ellos, por ejemplo se elige el certificado de Capy $cert_{Carlos,Capy}$, y se añade a la cadena de certificados de salida de Alicia. Finalmente se verifica el certificado emitido por Capy y dado que ella solo ha emitido un certificado, éste es elegido y como se han seleccionado tres certificados el algoritmo se detiene y el repositorio local de Alicia tiene los certificados $\{cert_{Alice,Chris}, cert_{Chris,Trent}, cert_{Trent,Jane}\}$.

3.5 PEBBLENETS

Solución propuesta por *Basagni*[17], que plantea un sistema de administración de llaves distribuida basada en encripción simétrica. La solución ofrece autenticación de grupo, integridad de mensajes y confidencialidad. Es adecuada para MANETS planeadas y distribuidas; los nodos pueden tener capacidad limitada de procesamiento y/o de almacenamiento. Con dispositivos 802.11 puede ser empleada sin que presente algún problema para el proceso de encripción/decripción.

Todos los nodos de la red comparten una identidad de grupo identificado por la llave $k_{GI,}$ la cuál se usa tanto para autenticación como para derivar llaves adicionales que serán utilizadas para proporcionar confidencialidad. El periodo de la llave de identidad del grupo permanece mientras las MANETS estén formadas, las llaves de confidencialidad son actualizadas en intervalos regulares. [17]

El tiempo de vida de la red se divide en periodos de tiempo, cada uno de ellos tiene tres fases que son la fase operacional, la fase de generación de cluster y la fase de actualización de la llave. A continuación se describe cada una de ellas.

Durante la fase operacional los nodos utilizan la llave de identidad de grupo k_{GI} para fines de autenticación e integridad de mensajes, la llave k_{TEK} se emplea para encriptar el tráfico de la red y con ello se garantiza la confidencialidad del mensaje.

En la fase de generación de clusters, se realiza la actualización distribuida de la llave encriptada, la red es segmentada en clusters, en donde cada uno de ellos tiene un nodo designado como cluster raíz.

Por último, en la fase de actualización de llave, uno de los clusters raíz es elegido como administrador de llaves, que será el responsable de generar una nueva llave de encripción y distribuirla a todos los clusters raíz. A su vez cada uno de ésos clusters raíz distribuye la nueva llave de encripción a todos los nodos miembros de la red.

La figura 3.12 ejemplifica cada una de las fases mencionadas.

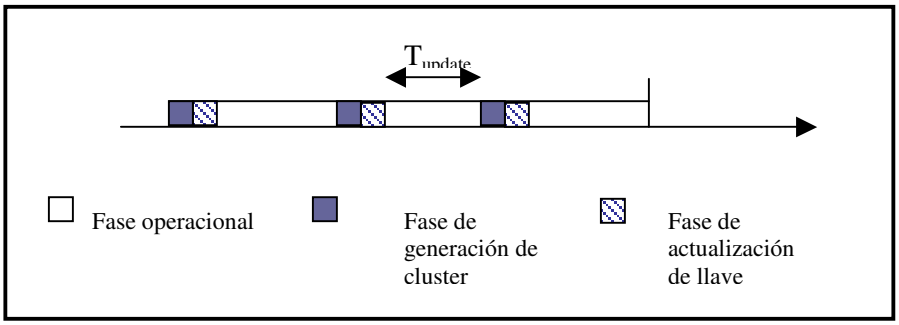


Figura 3.12 Diferentes fases durante el tiempo de vida de la red AD-HOC.

El proceso de autenticación se limita a los miembros del grupo, debido a que existe la complejidad natural de distribución de llaves, por lo tanto es necesario que todos los nodos de la red compartan una llave. Si se requiere autenticar un nodo es necesario administrar: $n*\left(\frac{n-1}{2}\right)$ llaves simétricas.

Los nodos de la red que tengan capacidad de almacenamiento, pueden almacenar la llave de identidad de grupo k_{GI}. [17]

El tiempo de vida en MANETS está dividido en periodos de tiempo y es donde se actualizan las llaves de encripción, por lo tanto se requiere que exista un mecanismo de sincronización de los relojes de los nodos. El proceso de sincronización queda fuera del alcance de está tesis.

3.5.1 PARÁMETROS DE LA SOLUCIÓN

La solución propone dos tipos de parámetros que son: No-criptográficos y criptográficos.

Dentro de los parámetros no-criptográficos es necesario que los nodos, antes de que puedan unirse a la red, cuenten con siguientes parámetros:

- El periodo de actualización de la llave, $t_{update.}$
- Un identificador de nodo único, $id_{i.}$
- Un promedio de retraso estadístico Δ.

El periodo de actualización de la llave define el tiempo entre las actualizaciones (ver figura 3.12). Si el periodo de actualización de la llave es corto, la seguridad de la red aumenta pero tiene la desventaja de que emplea más recursos.

Cada nodo tiene un identificador único *id,* el cuál se emplea en la fase de generación de cluster y en la elección del administrador de llaves.

El promedio estadístico de retraso Δ, se usa para minimizar el riesgo de que múltiples nodos sean administradores de llaves.

Dentro de los parámetros criptográficos, los siguientes son empleados por los nodos en las MANETS: [17]

- Llave de identidad del grupo, $k_{GI,}$ que es utilizada como semilla para otras llaves y para fines de autenticación.
- Llave de encripción de trafico, $k_{TEK,}$ sirve para encriptar datos.
- Llave de cluster, $k_{C,}$ su función es encriptar la comunicación interna en el cluster.
- Llave *backbone* k_B que es la encargada de encriptar la comunicación del cluster raíz.
- Llave de saludo k_H empleada durante la fase de selección del cluster raíz.

La llave identidad del grupo k_{GI} y la de encripción del tráfico de la red k_{TEK} se usan durante la fase operacional, mientras que las otras llaves son utilizadas durante la fase de generación de cluster y la de actualización de llave. La llave *backbone* y la de *saludo* son derivadas de la llave de identidad k_{GI} tal y como lo describe la siguiente formula: [17]

$$k_B^0 = k_{GI}$$

$$k_H^i = h(k_B^{i-1}) = h^i(k_{GI})$$

$$k_B^i = h(k_H^i) = h^{i+1}(k_{GI})$$

En donde: k^i representa la llave que se usa durante el *i-esimo* periodo de actualización, h^j constituye las *j* aplicaciones de la función hash *h*. La llave de encripción de tráfico de la red, se genera en forma aleatoria por el administrador de llaves, en cada periodo de actualización y la llave del cluster k_C se genera por cada cluster raíz, para comunicación dentro del cluster.

3.5.2 FUNCIONES CRIPTOGRÁFICAS

Para poder determinar que nodo desempeñará el papel del nodo raíz y cuál será el nodo administrador de llaves, es necesario calcular un valor que se denomina "peso", éste se define como: "valor que representa su estado actual con respecto a la batería disponible que tenga el nodo o la distancia hacia otros nodos" [17].

En el caso en donde los nodos no manejan parámetros criptográficos, durante la fase de generación de cluster y la fase de actualización de llave, todos los nodos deberán ser capaces de calcular su "peso, para ello se emplea la siguiente formula:

$$w_{id} = \sum_{k \in K} c_k P_k$$

En donde: C_k es el peso para el parámetro del sistema P_k y K es el conjunto de parámetros del sistema.

En contraparte, si los nodos utilizan parámetros criptográficos, las siguientes funciones son requeridas:

- Función hash de un sentido.
- Algoritmo de encripción simétrico.
- Algoritmo de generación de llave.

La función hash y el algoritmo de encripción simétrico, se emplean para fines de autenticación y confidencialidad, el algoritmo de generación de llave, se emplea para generar la llave de encripción de tráfico k_{TEK} en la fase de actualización de llave.

3.5.3 FASE DE GENERACIÓN DE CLUSTER

En ésta fase cada nodo decide de acuerdo a su peso *w* y al peso de sus vecinos más cercanos (de un solo salto), si actuará como cluster raíz o como un miembro ordinario del cluster. Una vez que se han seleccionado todos los clusters raíz, se deben descubrir mutuamente y configurar un cluster *backbone* raíz, el cual será empleado para distribuir las llaves de encripción de tráfico de la red.

Los pasos para la generación de cluster son: [17]

1. Descubrir nodos vecinos.

2. Seleccionar cluster raíz.

3. Crear cluster *backbone* raíz.

En el paso 1 cada nodo *i* calcula su peso w_i y envía el siguiente mensaje a todos sus nodos vecinos (de un solo salto):

$$E_{k_H^i}(w_i|id_i|MAC_{k_{GI}}(w_i|id_i))$$

En éste mensaje la llave de saludo k_H se utiliza para fines de confidencialidad, así como para proporcionarle al grupo la llave de identidad, para facilitarle la autenticación de mensajes.

Una vez que el nodo *i* recibió los mensajes de todos los nodos vecinos, inicia el paso 2 y debe decidir el rol que jugará dentro de las MANETS, ya sea como cluster raíz o como nodo ordinario. Para tomar esta decisión debe calcular su peso como se muestra en la fórmula anterior.

Si el nodo *i* no es el que tiene el mayor peso entonces su rol será de nodo ordinario, pero si es el que tiene el mayor peso, su rol será el del nodo raíz del cluster y mandará el siguiente mensaje, a todos sus vecinos informándoles su nuevo rol:

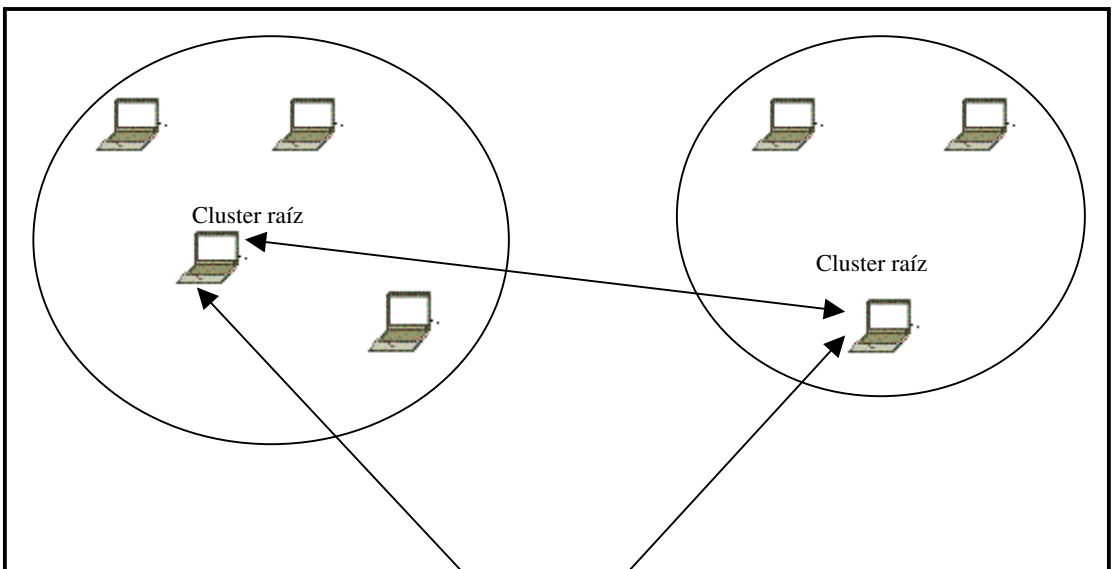
$$E_{k_H^i}(w_i|id_i|rol|MAC_{k_{GI}}(w_i|id_i|rol))$$

El valor del rol puede ser *CH* o id_j. Si el rol es igual a *CH,* significa que el nodo ha decidido tener el rol de cluster raíz; en cambio si es id_j, quiere decir que el rol del nodo es ordinario.

Después de que se han seleccionado todos los nodos raíz de la red, cada cluster raíz *c* genera una llave aleatoria de cluster k_C y la distribuye a cada miembro del grupo con el siguiente mensaje:

$$E_{k_H^i}(id_c|k_C^i|MAC_{k_{GI}}(id_c|k_C^i))$$

Con el mensaje anterior el paso dos concluye e inicia el paso 3. Los miembros del grupo, informan el identificador *id* del cluster raíz y su peso a cualquier otro cluster raíz que este en una cercanía hasta de tres saltos, posteriormente la red es segmentada en clusters internos que son interconectados a través de un cluster *backbone* como se muestra en la figura 3.13.



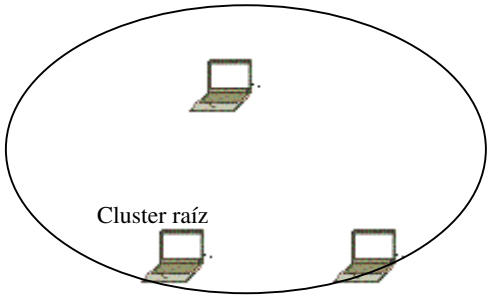


Figura 3.13. Segmentación de clustes y generación de cluster *backbone.*

3.5.4 FASE DE ACTUALIZACIÓN DE LLAVE.

El cluster raíz seleccionado en la fase anterior, desempeña en está fase, el rol de administrador de llaves y genera una nueva llave para encriptar el tráfico de la red, que distribuirá entre los otros clusters raíz.

Para seleccionar el administrador de llaves, todos los clusters raíz deciden primero quien de ellos puede ser un potencial administrador, verificando si sus clusters raíz vecinos tienen un peso mayor al de ellos, en caso contrario el nodo toma la decisión de ser el administrador de llaves. Posteriormente se calcula un retraso exponencial de acuerdo Δ (ver 3.5.1) y el administrador de llaves generará la nueva llave para encripción del tráfico de la red, que se define como: k_{TEK}^{i}, la cual es distribuida a todos los cluster raíz por medio del cluster *backbone* utilizando el siguiente mensaje:

$$E_{k_B^i}(id_c \| w_c \| k_{TEK}^i \| MAC_{k_{GI}}(id_c \| w_c \| k_{TEK}^i))$$

Si existiera otro potencial administrador de llaves *b* y recibiera el mensaje anterior se realizan cualquiera de las siguientes acciones:

1. Si *b* no ha generado una llave para encripción del tráfico de la red, el nodo *b* cancela su contador de tiempo y acepta el mensaje.
2. Si *b* ha generado una llave de encripción para el tráfico, descarta la llave recibida si la llave de *b* tiene un peso mayor que la llave del cluster raíz.
3. Si *b* ha generado una llave de encripción pero su peso es menor que la del cluster raíz, *b* aceptara la llave recibida como la nueva llave de encripción de trafico.

La nueva llave de encripción es distribuida por cada cluster raíz a sus miembros con el siguiente mensaje:

$$E_{k_C^i}(id_c \| k_{TEK}^i \| MAC_{k_{GI}}(id_c \| k_{TEK}^i))$$

3.6 IDENTIFICACIÓN DEMOSTRATIVA

Solución propuesta por *Balfanz* [18] que presenta un mecanismo de relaciones de confianza en la fase de arranque inicial de las MANETS, aún si no existe ninguna relación previa entre los nodos. La solución requiere que los nodos no tengan ningún vínculo previo y es adecuada para MANETS espontáneas y no planeadas.

Para establecer relaciones de confianza entre dos nodos, lo primero que se tiene que realizar es intercambiar datos entre ellos para autenticarse. Incorporando un dispositivo adicional al 802.11 se realiza un intercambio de pre-autenticación, en donde el canal es limitado y direccional, ejemplos de estos dispositivos adicionales son: el dispositivo infrarrojo, *bluetooth*[41], audio, etc. La figura 3.14 ejemplifica el uso de los dos canales:

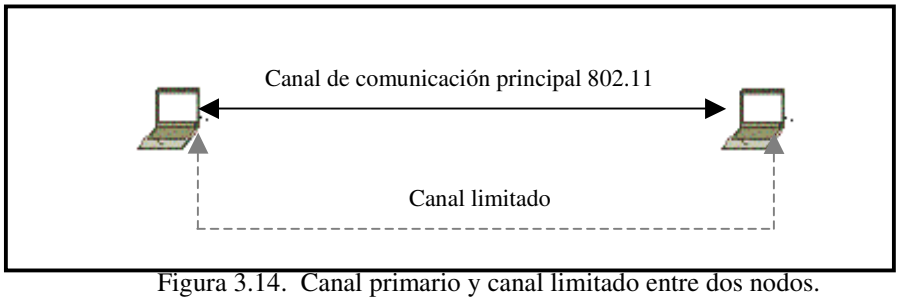


Figura 3.14. Canal primario y canal limitado entre dos nodos.

El siguiente ejemplo ilustra el concepto de identificación demostrativa. Un usuario con una PDA que desea conectarse a una impresora determinada, puede usar un canal infrarrojo de ésta para identificar a la impresora de la cual va requerir el servicio, bastará con que apunte el dispositivo infrarrojo de la PDA hacia el de la impresora para iniciar el proceso de autenticación y posteriormente, si fue exitoso, acceder al servicio.

La solución requiere que todos los nodos tengan el mismo dispositivo para el canal dedicado, por ejemplo: dispositivo infrarrojo, audio, etc. Y que permita el intercambio de llaves en MANETS.

Cabe señalar que de acuerdo a las características que tienen las comunicaciones con infrarrojo y que requieren una distancia establecida, el usuario puede estar seguro que los datos de autenticación intercambiados son originados de la impresora o del dispositivo del cual se requerirá un servicio.

Una vez que se realizo el proceso de intercambio de pre-autenticación, se pueden utilizar algoritmos de llave pública, como Diffi-Hellman, en el canal principal (802.11) para establecer una llave común que se usará para proteger la comunicación entre los nodos. Utilizando identificación demostrativa y pre-autenticación los autores [17] describen protocolos para intercambio de llaves.

3.6.1 INTERCAMBIO DE LLAVES

Para el intercambio de llaves se presentan dos protocolos diferentes. El primero se denomina de dos-partes, el cuál requiere que ambas partes sean capaces de realizar cálculos criptográficos de

[41] Proyecto que surgió en 1994 para definir un estándar para redes de área personal. Para mayores detalles consultar la página http://www.bluetooth.com.

llave pública. El segundo requiere que solo una de las partes involucradas, sea capaz de realizar operaciones criptográficas.

En el protocolo de dos-partes, los nodos *A* y *B* deben ser capaces de realizar criptografía de llave pública. Los nodos para intercambiar una llave secreta de encripción deben realizar el siguiente procedimiento: [18]

1. El nodo *A* utiliza el canal limitado y envía la firma digital de su llave pública $Hash(pk_A)$ al nodo *B* y éste responde enviando la firma digital de su llave pública $Hash(pk_B)$ al nodo *A*.

2. En el canal principal 802.11 los nodos intercambian sus llaves públicas. Las llaves recibidas pk_A^* y pk_B^* son autenticadas verificando la firma digital de las llaves recibidas: $Hash(pk_A^*) = Hash(pk_A)$ y $Hash(pk_B^*) = Hash(pk_B)$.

3. Si la verificación es correcta, se puede utilizar cualquier protocolo de llave pública como Diffie-Hellman o RSA.

En el caso que solo uno de los nodos sea capaz de ejecutar criptografía de llave pública el siguiente procedimiento se lleva a cabo:

1. Utilizando el canal limitado el nodo *A* (que es capaz de realizar criptografía de llave pública) envía al nodo *B* la firma digital de su llave pública $Hash(pk_A)$ y el nodo *B* envía al nodo *A* la firma digital de $Hash(s_B)$, donde s_B es un secreto generado por *B*.

2. En el canal principal 802.11 el nodo *A* envía su llave pública pk_A al nodo *B*.

3. El nodo *B* autentica la llave recibida pk_A^* verificando la firma digital de está conforme a: $Hash(pk_A^*) = Hash(pk_A)$

4. Si la verificación es correcta, el nodo *B* encripta s_B con la llave pública pk_A de A y le envía $E_{pk_A}(s_B)$.

5. El nodo A decripta el mensaje recibido $E_{pk_A}(s_B^*)$ con su llave privada y verifica la firma digital de s_B empleando: $Hash(s_B^*) = Hash(s_B)$. Si la verificación es correcta los dos nodos son autenticados y el secreto compartido s_B puede usarse para generar una llave de encripción compartida.

El protocolo asume que el algoritmo de llave pública que se usa tiene una función de encripción apropiada o barata. RSA es un ejemplo de este algoritmo si *e* es elegido en forma adecuada.

4. ANÁLISIS DE LAS SOLUCIONES PRESENTADAS

En el capítulo dos se comentó que las soluciones tradicionales para administración de llaves, presentan una serie de requerimientos que las hacen inadecuadas para MANETS. El uso de un tercer elemento confiable, que tenga como función proporcionar los servicios de una CA, implica que una infraestructura organizacional o administrativa este presente, en algunas MANETS esto no es posible, pero en otras, como en la solución que propone Zhou [5], se puede distribuir en forma parcial la autoridad certificadora.

Es necesario modificar y adaptar las soluciones existentes para MANETS, las soluciones presentadas cumplen con los requisitos para establecer mecanismos para la administración de llaves en MANETS. En este análisis se retoman cada una ellas y se señalan las ventajas y desventajas que tienen desde una perspectiva de seguridad.

Cabe señalar que el análisis se efectúa desde una óptica funcional, con un enfoque desde un punto de vista de la seguridad de sistemas y fundamentado en conceptos criptográficos. No se realizo un análisis formal de los protocolos que se presentaron en este trabajo, basado en lógica de BAN o algún sistema experto, porque los protocolos ya han sido probados bajo ésos modelos y no sería novedoso rehacer un análisis de algo que ya se ha realizado. Es por ello que éste análisis se realiza bajo otra óptica.

4.1 ANÁLISIS DE LA SOLUCIÓN AUTORIDAD CERTIFICADORA PARCIALMENTE DISTRIBUIDA

Está solución se caracteriza porque requiere que una infraestructura organizacional/administrativa se encuentre disponible. Desde un punto de vista funcional la solución presenta fallas, en particular la del mecanismo de revocación de certificados, porque cualquier solución que se base en el uso de certificados, debe considerar el riesgo de que un atacante pueda comprometer a las MANETS.

La solución requiere que un nodo servidor almacene todos los certificados emitidos y dé un mecanismo de sincronización, que propague la emisión de un nuevo certificado a todos los nodos de la red. También se considera el caso cuando las MANETS son segmentadas y posteriormente reensambladas. Uno de los problemas que se puede llegar a presentar en el proceso de reensamble es la sincronización entre los servidores, este problema se ejemplifica a continuación:

En la figura 4.1 todos los nodos de la red están interconectados y mantienen el repositorio de certificados sincronizado sin ningún problema.

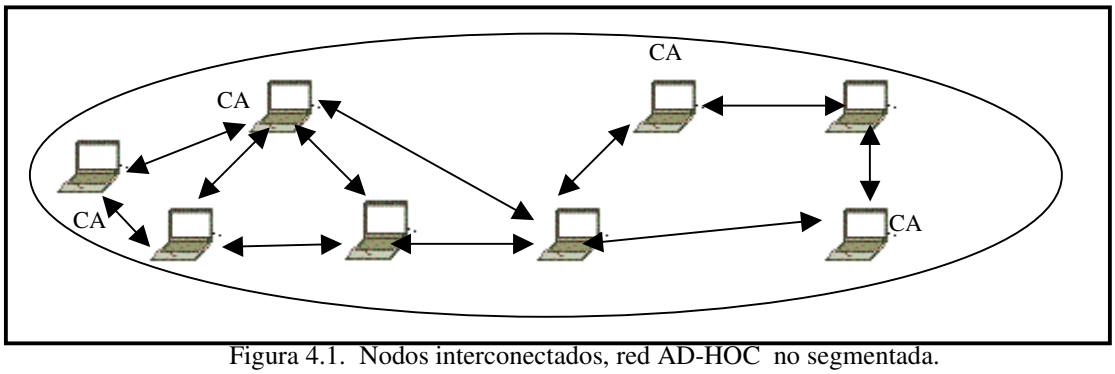


Figura 4.1. Nodos interconectados, red AD-HOC no segmentada.

Posteriormente, la red esta segmentada en dos MANETS(red 1,red 2) y en cada una de ellas se renuevan sus certificados, ver figura 4.2.

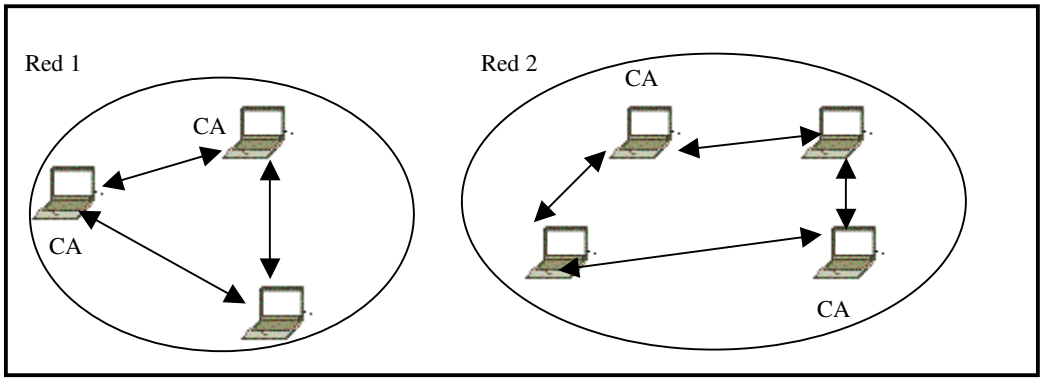


Figura 4.2 Red Segmentada.

Cuando las dos redes segmentadas (red 1 y red 2) son reensambladas, el repositorio de certificados de los nodos servidores será inconsistente y requerirán de un proceso de sincronización para tener consistencia en los certificados firmados por la llave sk_{CA} de las autoridades certificadoras. Ver figura 4.3.

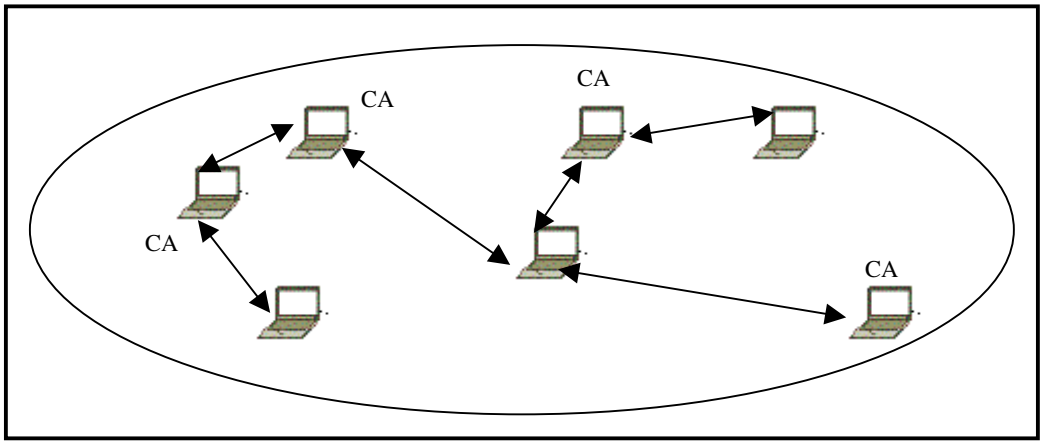


Figura 4.3. Red reensamblada.

La disponibilidad del servicio de la CA, depende de los parámetros de los secretos compartidos *(k,n),* entre más grande sea el parámetro *k* mayor será la seguridad que se logrará en las MANETS y se garantizará la disponibilidad del servicio. Para un adversario que intente derivar *k* (si se elige *k* como un valor grande) representará un esfuerzo computacional alto encontrarlo.

Otro problema que ocurre en este escenario, es cuando un nodo servidor realiza el proceso de actualización compartida, los nodos servidores deben estar sincronizados, de modo que tengan una visión consistente del certificado de la CA firmado con la llave $sk_{CA.}$ De lo contrario el proceso de actualización presentará fallas.

Esta solución es vulnerable a un ataque de negación de servicio, porque el nodo mezclador no verifica la validez de los certificados y si éste proviene de un nodo malicioso puede comprometerse la seguridad de la MANET, el ataque puede ser prevenido si el nodo mezclador verifica la validez del certificado antes de aceptarlo.

Respecto al manejo de direcciones, la solución da a conocerla forma en que un nodo cliente localiza a un nodo servidor. Cuando un nodo se une a la MANET, esta interesado en localizar la llave *k* del nodo servidor; el servicio de la CA puede tomar una dirección *multicast* y entonces un nodo cliente que requiera el servicio de la CA puede enviar la solicitud a la dirección *multicast* de está. El nodo servidor que escoge atender la solicitud responde con su dirección *unicast,* otra opción en el caso que las MANETS no soporten tráfico *multitast,* es que el cliente haga la solicitud vía *broadcast,* con el inconveniente de que se generaría tráfico excesivo en las MANETS. El manejo de direcciones no representa un problema en está solución en el protocolo 802.11 ya que cuenta con los campos adecuados para manejar más de una dirección, el inconveniente sería si se envían solicitudes vía *broadcast* el tráfico de la red podría aumentar en forma considerable.

4.2 ANÁLISIS DE LA SOLUCIÓN AUTORIDAD CERTIFICADORA COMPLETAMENTE DISTRIBUIDA

Como la solución anterior, está requiere de una autoridad administrativa para los servicios de registro e inicialización, pero a diferencia de la solución de *Autoridad Certificadora Parcialmente Distribuida* [5] todos los nodos en las MANETS tienen una *k-parte* de la llave privada sk_{CA} de la CA.

A diferencia de la solución *Parcialmente Distribuida [5]* está contiene un mecanismo para revocar certificados. La mayor ventaja que ofrece está solución es la disponibilidad, porque los servicios de la autoridad certificadora están distribuidos entre los nodos de la red.

En está solución, el manejo de emisión de certificados es parte del servicio de la CA y todos los nodos de la MANET pueden proporcionar este servicio. Si un nodo solicita el servicio de emisión de certificados, el requisito es que éste tenga vecinos (de un solo salto) que le proporcionen el servicio.

El problema de direccionamiento, que se presenta en la solución *Parcialmente Distribuida [5]*, no se presenta aquí, porque las solicitudes de emisión, revocación y actualización siempre se realizan hacia los nodos vecinos (de un solo salto), por lo que las tablas de ruteo no tienen que actualizarse en forma constante.

El tráfico de la red es limitado y con dispositivos 802.11, no presenta problemas para el manejo de criptografía de llave pública ni para el envío de mensajes entre nodos.

Empleando los mecanismos de inicialización compartida y de actualización compartida se obtiene una alta disponibilidad de los servicios de la red.

En cuanto al manejo de secretos compartidos *(k,n),* en está solución las *k-partes* están expuestas, debido a que cada nodo tiene una *k-parte* del secreto compartido (a diferencia de la solución de *Autoridad Certificadora Parcialmente Distribuida [5]*, en donde un nodo especializado se encarga del manejo de las *k-partes*), a que un adversario, en la fase de actualización, pueda comprometer a un nodo y obtenga la *k-parte* permitiéndole reconstruir el secreto compartido, vulnerando la seguridad de la red.

Una solución a lo anterior es que se elija un número grande para el parámetro *k* (del esquema de secretos compartidos *(k,n)*), para aumentar la complejidad de que el adversario pueda reconstruir el secreto compartido.

Por otra parte, con el proceso de actualización de certificados, la solución ofrece un mecanismo de sincronización para redes segmentadas. El mecanismo de revocación de certificados, se basa en el supuesto de que todos los nodos monitorean el comportamiento de sus vecinos cercanos (vecinos que están a un solo salto) y que mantienen sus propias listas de revocación de certificados.

4.3 ANÁLISIS DE LA SOLUCIÓN AUTOEMISIÓN DE CERTIFICADOS

La mayor ventaja que presenta esta solución, es que no requiere de ningún servidor especializado, como en la solución parcialmente distribuida o autoridad administrativa para el manejo de sus certificados. Pero la desventaja es que no ofrece ningún mecanismo de revocación ni de actualización de certificados. Al igual que PGP la solución presenta problemas durante las fases iniciales de emisión de certificados [19].

En la figura 4.4 se ejemplifica el proceso de *Autoemisión de Certificados* [12] y como se resuelve el problema de la propagación de certificados. La imagen muestra que existe una "cadena amistosa" que conecta Alicia y a Beto, pero hasta el momento solo Capy y Oscar han emitido certificados y no existe un cadena de certificados entre Alicia y Beto, hasta que los otros usuarios envíen su certificados a sus amigos Alicia y Beto podrán comunicarse de una manera segura.

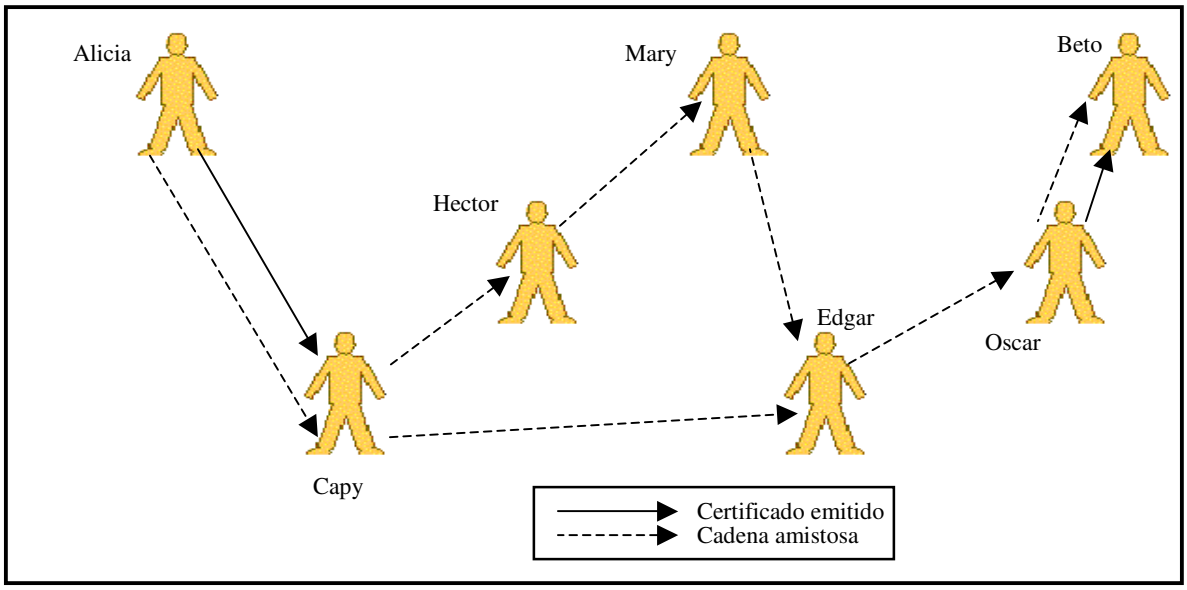


Figura 4.4. Ejemplo de propagación de certificados con PGP.

La solución asume que todos los usuarios son "introductores o presentadores" confiables. Un "introductor" es un usuario que ingresa usuarios en forma confiable, por ejemplo en la figura 4.4 Capy puede emitir un certificado a Héctor, pero no confía en ningún certificado que llegase a emitir Héctor, pero si confía en un certificado emitido por Edgar ya que él es un "introductor confiable". Si Capy confía en los certificados que emiten otros usuarios, a los cuáles Héctor emitió certificados (por ejemplo Oscar) entonces Héctor es conocido como un "meta-introductor".

El algoritmo que proponen "Shortcut Hunter" para la selección de certificados, describe que cada usuario puede almacenar un número limitado de certificados [12], éste ha sido probado en PGP, un ejemplo de ello, es la base de datos de certificados disponibles en internet. En MANETS adquiere una mayor complejidad, porque no existe un conocimiento global de todos los certificados que están disponibles y en la solución no proponen algún proceso para actualizar la base de datos de certificados ni de revocación. Está es una de las principales desventajas de la solución.

Otro problema que se observa, es en la cadena de certificados, si la cadena es muy grande, la búsqueda de un determinado certificado puede ser muy lenta, y se puede llegar a presentar el problema de que un adversario comprometa a un nodo e inserte un certificado no válido. En el caso que se utilice un dispositivo 802.11, se puede exceder el tiempo de vida de la solicitud. Para prevenir esto, se sugiere, limitar el tamaño de la cadena, para prevenir el impacto de que un nodo sea comprometido y evitar que se exceda el tiempo de vida de la solicitud en 802.11, es viable utilizar un mecanismo de revocación de certificados.

Por último hay que destacar que al no emplear ningún servidor dedicado, que funja como autoridad certificadora, los nodos de la red, no dependen de ninguna tercera entidad y ellos pueden de una forma libre, asociarse en las MANETS y compartir su llave pública con los nodos en los cuales confían.

4.4 ANÁLISIS DE LA SOLUCIÓN PEBBLENETS

Se basa en criptografía simétrica y requiere una infraestructura organizacional/administrativa que inicialice los nodos de la red con una llave de identidad de grupo compartida k_{GI} y un parámetro adicional t_{update}.

El problema que tiene está solución, es que requiere que un nodo mantenga el almacenamiento de certificados centralizado. Si se llegará a comprometer la llave de identidad del grupo k_{GI} todos los nodos necesitan ser re-inicializados con una nueva llave de identidad y no presenta ningún esquema de revocación de llaves.

Aunado a lo anterior, las siguientes desventajas pueden presentarse en esquemas de llave simétrica: [2]

1. En la distribución de llaves, los usuarios tienen que seleccionar llave en secreto antes de empezar a comunicarse.

2. En el manejo de llaves, en una red de n usuarios, cada pareja debe tener su llave secreta particular, por ejemplo:

$$n * \left(\frac{n-1}{2} \right) \text{ llaves.}$$

3. Sin firma digital no hay posibilidad, en general, de firmar digitalmente los mensajes.

En la fase de generación de cluster, se puede llegar a presentar el problema en donde dos nodos tenga el mismo peso y ambos sean candidatos a tener el rol de cluster raíz, los autores no presentan alguna solución a este problema.

Las ventajas que ofrece está solución es que los nodos de las MANETS pueden tener capacidad limitada de procesamiento y/o de almacenamiento y pueden pertenecer a la red. Y la desventaja más notable, radica en la administración de llaves, debido a que se lleva a cabo en forma

centralizada, ésta puede presentar los problemas inherentes a la criptografía simétrica. Se considera adecuada para MANETS planeadas y distribuidas, porque soporta la autenticación de grupo, la desventaja es que no aplica para MANETS punto a punto.

4.5 ANÁLISIS DE LA SOLUCIÓN DE IDENTIFICACIÓN DEMOSTRATIVA

Las soluciones anteriores, necesitaban de una infraestructura organizacional (*Autoridad certificadora completamente distribuida [6] o parcialmente distribuida [5], Pebblenets [17]*) o algún tipo de distribución basada en grupos de nodos (*Autoemisión de Certificados [12]*). En está solución no se requiere de ninguna de estás, debido a que esta orientada a MANETS espontáneas, dinámicas y con gran movilidad. El requisito que deben tener los nodos en las MANETS, es que tengan un dispositivo adicional al 802.11 para garantizar la autenticación. Este dispositivo adicional puede ser infrarrojo o *bluetooth.* El objetivo de este dispositivo es realizar una pre-autenticación entre los nodos y que los datos de pre-autenticación viajen en el canal limitado. En la solución, los datos que son intercambiados entre los nodos en el canal limitado, son la firma digital de la llave pública de los nodos.

La desventaja de está solución, radica en que si un nodo solo cuenta con el dispositivo 802.11 no puede llevar a cabo el intercambio de pre-autenticación. Y muchas veces el dispositivo adicional como *bluetooth* puede ser costoso. No cuenta con ningún mecanismo para renovación ni de revocación de llaves.

El alcance máximo del dispositivo adicional en el caso de infrarrojo es hasta 5 metros y el rayo debe ser dirigido, por lo tanto si el rayo es interferido por objetos puede sufrir interferencias. En *bluetooth* el rango de operación es de 10 metros y la señal es omnidireccional, la frecuencia que utiliza es 2.4-2.4835 Ghz, y es la misma que utiliza 802.11, que no representa un problema entre 802.11 y *bluetooth* debido a que utiliza saltos de frecuencia (FH) [42] y con ello evita la interferencia de otras señales saltando a una nueva frecuencia antes de enviar o transmitir un paquete. El protocolo 802.11 también utiliza FH así como DSSS[43] y OFDM[44] como se presentó en el capítulo 1.

El dispositivo 802.11 se emplea para el intercambio de datos en forma segura y el dispositivo adicional para garantizar la autenticación. La desventaja principal oscila que si no se cuenta con el dispositivo adicional al del 802.11, la autenticación no se llevaría a cabo, pero la mayoría de los fabricantes incorporan dichos dispositivos de fábrica. La solución es una buena opción para MANETS espontáneas, por ejemplo, cuando un usuario dentro de un aeropuerto desea acceder algún servicio de red, como consulta de correo o impresión, el diseño de está solución se considera adecuado porque puede garantizar que el usuario que emplea dichos servicios este autenticado en la MANET, cabe señalar que la solución además de aplicarse en computadoras portátiles o PDA's, también puede ser aplicada en otros dispositivos como impresoras, proyectores, scanners, para ello se necesita que éstos cuenten con el dispositivo adicional.

---

[42] Por sus siglas en inglés Frecuency hopping
[43] Ídem Direct Sequence.
[44] Ídem Orthogonal Frequency Division Multiplexing

4.6 COMPARATIVO DE SOLUCIONES

Las soluciones presentadas en este trabajo difieren en forma significativa en requerimientos, funcionalidad y complejidad para su implementación. A continuación se realiza un comparativo entre las distintas soluciones.

Las soluciones de *Autoridad Certificadora Parcialmente Distribuida* [5] y *Completamente Distribuida [6]* son similares porque utilizan un esquema de secretos compartidos, para distribución de una llave privada. Difieren en la forma en que administran los certificados, la *Autoridad Certificadora Parcialmente Distribuida* [5] es similar a la forma de administrar certificados en *Pebblentes* [17]. Emplean un esquema de criptografía de llave pública, al igual que las demás soluciones, excepto la de *Pebblentes [17]*. Son adecuadas para MANETS distribuidas con gran amplitud, como MANETS militares en el campo de batalla o para desastres naturales, en las cuáles los nodos están definidos y clasificados. Si los nodos de las MANETS en está solución cuentan con dispositivos 802.11, éstos no presentan problemas para el intercambio de mensajes ni para el direccionamiento, porque el diseño del protocolo cuenta con los campos requeridos para el manejo de direcciones y de mensajes. Difieren de las otras soluciones en el esquema de distribución de llaves.

La solución de *Pebblentes* [17] es adecuada para MANETS en donde los nodos estén distribuidos y que alguno de los nodos no tenga la capacidad o la necesidad de emplear criptografía de llave pública. Si los nodos en estas MANETS cuentan con dispositivos 802.11 pueden realizar criptografía de llave pública sin problema, pero el diseño de la solución contempla la posibilidad de que alguno de los nodos no cuente con dispositivos que sean capaces de efectuar cálculos criptográficos, por lo tanto si alguno de los nodos no cuenta con el dispositivo 802.11, se pueden presentar problemas en el intercambio de mensajes entre ellos, debido a que se completaría el ciclo de comunicación entre los nodos. Es similar a la solución de *Autoridad Certificadora Parcialmente Distribuida* [5] en la administración de llaves, porque requieren de una autoridad certificadora para que realice las funciones de administración y distribución de llaves. Difiere de todas las soluciones en el esquema criptográfico que utiliza, que es de llave simétrica.

La solución de *Identificación Demostrativa* [18] está orientada a MANETS espontáneas, dinámicas y con gran movilidad. La desventaja es que los nodos de las MANETS requieren de un dispositivo adicional al 802.11 para efectuar los procesos de autenticación entre ellos. El dispositivo 802.11 no presenta problemas para el intercambio de mensajes cifrados en está solución. Se considera que es la mejor solución para MANETS espontáneas y con temporalidad mínima. Es similar a la solución de *Autoemisión de Certificados [12]* en la administración de llaves, porque no requieren de ninguna estructura administrativa, y difiere de todas las soluciones, en que requiere de un dispositivo adicional para entablar el proceso de autenticación. Utiliza un esquema de criptografía de llave pública como todas las soluciones, a diferencia de *Pebblentes. [17]*

Por último la solución de *Autoemisión de Certificados* [12], es la que se adecua a la naturaleza de las MANETS, porque no depende de ninguna otra infraestructura o autoridad fija para emitir y compartir sus certificados. Difiere en la forma que administran los certificados y las soluciones de *Autoridad Certificadora Parcialmente* [5], *Autoridad Completamente Distribuida [6] y*

Pebblenets [17], pero es similar al que emplea *Identificación Demostrativa* [18]. El esquema de criptografía que utiliza, es de llave pública como todas las soluciones, excepto la de *Pebblentes* [17]. En cuanto a la infraestructura, se considera adecuada para MANETS distribuidas y no es necesario que exista cierto conocimiento entre los nodos, porque cuenta con los mecanismos para realizar el intercambio de certificados, aún si no ha existido cierta relación previa entre los nodos.

La solución presentaba dos grandes problemas que eran el mecanismo de revocación y actualización de certificados, y con la propuesta de solución (ver 5.1 y 5.2) han quedado mitigados. Si los nodos en las MANETS cuentan con dispositivos 802.11, los procesos de encripción y de intercambio de mensajes pueden llevarse a cabo sin ningún problema.

En la tabla 4.1 se presenta un resumen de las diferencias y similitudes de las soluciones en cuanto a su tipo, CA, alcance y temporalidad y si presenta problemas con 802.11.

Tabla 4.1. Comparativo entre las soluciones.

Solución	**Tipo de MANET**	**CA**	**Alcance**	**Temporalidad de la MANET**	**Problemas con 802.11**
CA parcialmente distribuida	Planeada	Si	Distribuido	Largo término	No
CA completamente distribuida	Planeada	Si	Distribuido	Largo término	No
Autoemisión de certificados	Espontánea	No	Distribuido	Largo término	No
Pebblenets	Planeada	Si	Distribuido	Largo término	Si
Identificación demostrativa	Espontánea	No	Local	Corto término	No

5. PACDRE: PROTOCOLO DE AUTOEMISIÓN DE CERTIFICADOS CON DOS REPOSITORIOS

De las soluciones que se presentaron y conforme al análisis que se realizó, se puede concluir que existen dos formas de realizar el manejo de certificados y llaves en MANETS: la primera, es aquella donde la distribución y manejo de las llaves las realiza una autoridad centralizada [5] o una autoridad semi-centralizada [6] y la segunda, es donde no está presente ninguna autoridad o entidad centralizada como la solución de *Autoemisión de Certificados [12]*, la cual le da a cada nodo autosuficiencia en el manejo, almacenamiento, creación y distribución de llaves. Además se basa en un esquema de encripción de llave pública y cumple con los requerimientos de independencia y autosuficiencia en el manejo de llaves para MANETS.

La solución de *Autoemisión de Certificados [12]* será retomada, respecto al uso de PGP, del esquema de administración de llave pública y a la distribución de certificados. Pero no se utilizará el esquema de revocación de certificados ni de actualización. Con PACDRE se propone una solución a los dos problemas que se identificaron: el primero es el proceso de actualización y el segundo la revocación de certificados de acuerdo a una fecha determinada. (ver seccion 4.3).

El problema principal de cualquier sistema que basa la seguridad de las llaves en algoritmos de llave pública, es lograr que cada llave pública de los usuarios se encuentre disponible para cualquier usuario dentro del sistema y además pueda verificar la autenticidad de está. La solución al problema de verificación, es el uso de certificados con llave pública. La implementación de está solución se agudiza en MANETS y aumenta la dificultad para solventarlo, por la ausencia de servidores centralizados. [21]

Para resolver el problema del proceso de autenticación de las llaves, se emplea el proceso que se denomina “cadena de certificados” [12]. Este proceso consiste en que si un nodo desea obtener un certificado de otro, deberá realizar una búsqueda en su repositorio y si no lo encuentra deberá realizar una solicitud a su vecino (de un solo salto) para que también lo busque en su repositorio local y así sucesivamente. Un nodo puede realizar solicitudes de búsqueda hasta por un máximo de seis nodos, que puede extenderse por toda la MANET, debido a que cada nodo a su vez, puede realizar dicha solicitud y buscar en el repositorio de sus nodos vecinos y de esta forma ampliar la búsqueda del certificado por toda la MANET.

El problema que se deriva de este proceso, es que los nodos no tienen una vista consistente de los certificados en su repositorio, y no pueden diferenciar que certificados ya no son válidos, porque expiraron conforme a una fecha determinada, y cuales no lo son porque han sido revocados.

Para solventar este problema, se propone el uso de *PACDRE,* que añade un repositorio adicional para el nodo, el objetivo de que cuente con dos repositorios es que: Uno que funja como repositorio de certificados actualizados y otro que actué como repositorio de certificados vencidos.

El repositorio de certificados vencidos, se utilizará solo para almacenar certificados que hayan expirado conforme a una fecha determinada. El objetivo de tenerlos almacenados en un repositorio independiente, es que los certificados son renovados muy seguido por sus emisores y muy pocas veces son revocados. Por lo tanto, resulta eficaz almacenarlos por un tiempo. En el caso que sean actualizados por sus emisores, se deben transferir al repositorio de certificados actualizados, el cuál almacena todos los certificados válidos y que pueden ser intercambiados entre los nodos de la MANET. Además de eficientar la administración de certificados, la contribución ayudará a aumentar la capacidad de respuesta en los nodos al reducir el espacio de búsqueda a un solo repositorio que solo tendrá certificados válidos y actualizados.

PACDRE es una modificación a la propuesta de PGP [12,14], debido a que añade un repositorio adicional. Los mecanismos de actualización, renovación y revocación de certificados, se realizaran con el lenguaje SQL [45], que es el estándar para el manejo de base de datos.

5.1 MARCO DE TRABAJO

A continuación se detallan las caracteristicas de los nodos, el protocolo, el esquema de consumo y los procedimientos de almacenado.

Nodos:

1. Laptop: Procesador Pentium a 1.2 MHz, Memoria RAM de 800 MB, Tarjeta Inalambrica PCMCIA 802.11, Disco Duro con espacio disponible de 500 MB, Bateria cargada.

2. Software: S.O. WindowsX y PGP.

Protocolo:

1. Se sugiere un protocolo híbrido como ZRP debido a que combina las características de los protocolos proactivos de manera local y reactivos en forma global. Este protocolo divide la MANET en zonas, los nodos que se encuentren dentro de una zona especifica emplean

[45] Por sus siglas en inglés Structured Query Language

protocolos proactivos y para todos aquellos nodos que se encuentren fuera de ésa zona emplea protocolos proactivos.

Consumos:

1. El factor de consumo de energía en los nodos, es un factor que debe tomarse en cuenta para los protocolos de ruteo en las MANETS. La limitante de la energía en los nodos debe considerarse en el diseño del protocolo. El uso de metricas de ruteo que contemplen la cantidad de energía que disponen los nodos para trazar nuevas rutas es un factor que ayuda a conservar el tiempo de vida de las baterías en las MANETS. *S.Sing et al* [40] propusierón un conjunto de metricas de ruteo, que ayudan a conservar el tiempo de vida de las baterías de los nodos en las MANETS. Estas metricas están orientadas a minimizar el consumo de energía por paquete cuando éste es enviado desde el nodo fuente al nodo destino. El cálculo de la métrica se conforma de la suma de la energía requerida por cada nodo intermedio (de un solo salto) en función de la distancia hasta alcanzar el nodo destino. [40]. Si se toma en cuenta está métrica en el calculo de rutas en el diseño del protocolo se podrá eficientar la vida útil de las baterías de los nodos en las MANETS.

2. El protocolo de ruteo que se eligio en éste trabajo es ZRP, para ello tiene que ser modificado para implementar las métricas de consumo sugeridas por *S.Sing,* para optimizar el consumo de batería de los nodos de las MANETS, la modificación al protocolo queda fuera del alcance de éste trabajo.

Procedimientos de Almacenado:

1. Los procedimientos de almacenado serán disparados a través de un proceso de calendarización de tareas conocido como "*cron*". Se descarta el uso de un *Trigeer*, que es un procedimiento de base de datos que se ejecuta conforme a una acción determinada. Un *Trigeer* a diferencia del cron no puede ser programado para ejecutarse conforme a una calendarización y para efectos de este trabajo requerimos que dicho procedimiento almacenado se ejecute conforme a una calendarización sin la intervención del usuario.

5.2 MECANISMO PARA MANTENER EL REPOSITORIO DE CERTIFICADOS ACTUALIZADOS

El objetivo de desarrollar un mecanismo para mantener el repositorio de certificados actualizados, es la de crear un proceso automatizado, para que el usuario cuente con certificados vigentes en su repositorio.

La creación de un repositorio adicional, modifica a la solución de PGP [14], porque en está solo se considera un repositorio para cada usuario, (ver figura 5.1).

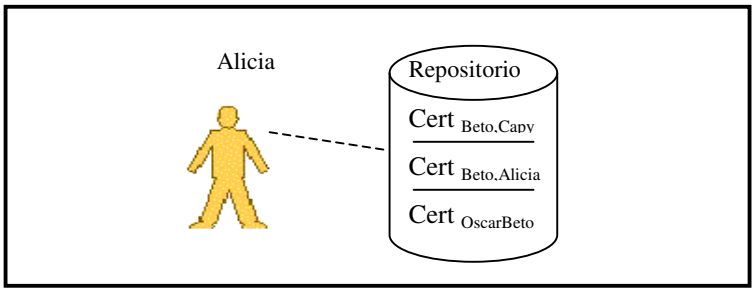


Figura 5.1. Usuario con su repositorio.

Cuando el usuario genera sus llaves pública y privada, el primer paso que se lleva a cabo en PACDRE es la creación de dos repositorios (ver figura 5.2), a los cuáles les creará un identificador único, por ejemplo: Repositorio A y Repositorio V. Este proceso es el mismo que realiza PGP [12,14] pero con PACDRE se añade un repositorio adicional.

Cabe señalar que el manejo de los datos en un repositorio, se puede llevar a cabo de la misma forma que el manejo de una base de datos. Para ello se emplea SQL, que es el lenguaje estándar ANSI para manipulación de base de datos. El lenguaje contiene un conjunto de instrucciones, para particionar y combinar relaciones entre repositorios de bases de datos. Por lo tanto, los repositorios de las llaves pueden ser administrados empleando instrucciones del lenguaje SQL.

Las principales ventajas que proporciona el uso de SQL son: [38]

1. Eficiencia.

2. Facilidad para su aprendizaje y uso.

3. Completa funcionalidad, permite definir, recuperar y manipular datos en las tablas de las bases de datos.

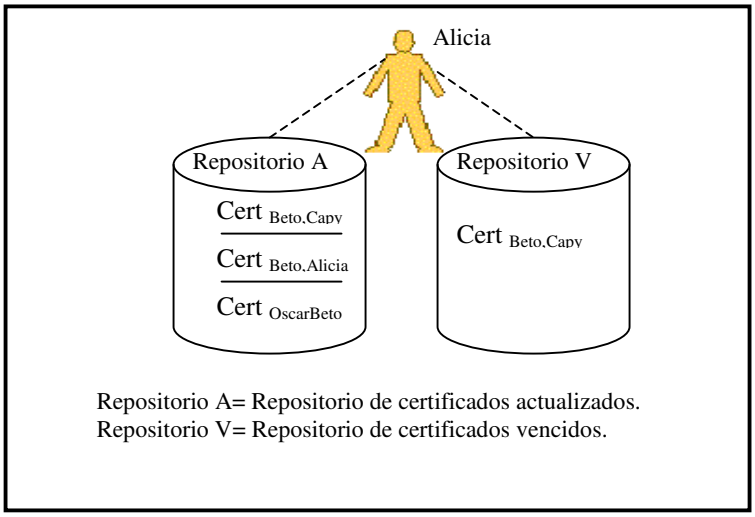


Figura 5.2. Usuario con dos repositorios.

Para el proceso de actualización de los repositorios, se hace uso de los procedimientos de almacenado de SQL (*store procedures* [38]) denominado en nuestra solución como *proc_actualiza*. El objetivo de este procedimiento es permitir la actualización de certificados de forma autónoma e independiente. El procedimiento se ejecuta en forma periódica y se encuentra en el repositorio A. La ejecución periódica se lleva a cabo a través de un proceso de calendarización de tareas conocido como "*cron*". Cada vez que el usuario necesite certificar la autenticidad de una llave, se llama a *proc_acualiza*.

En la sección 1.3.4 se observo que un certificado, bajo el estándar X.509, está compuesto por los siguientes campos:

- Nombre del usuario y otra información como su e-mail.
- Llave pública del usuario.
- Nombre del emisor (CA).
- Número de serie.
- Período de validez.

El campo período de validez se emplea para verificar que la fecha de vencimiento de un certificado, no se haya cumplido.

El procedimiento *proc_actualiza*, está conformado por las siguientes instrucciones:

1. Para seleccionar el período de validez de un certificado se realiza, lo siguiente:

select periodo_validez from repositorio_actualizado where periodo_validez <= fecha_actual

Si el resultado de la sentencia anterior es *null*, entonces todos los certificados del repositorio están actualizados y el procedimiento termina, en caso contrario continua con el paso 2.

2. La sentencia anterior, nos arroja como resultado aquellos certificados cuyo período de validez es menor o igual que la fecha actual. Con lo anterior se pueden mover los certificados seleccionados al repositorio de certificados vencidos.
   Para ello se utiliza la siguiente instrucción *SQL*:

insert into repositorio_vencidos (nombre_usuario,llave_pública,nombre_emisor ,número_serie,periodo_validez) select periodo_validez from repositorio_actualizado where periodo_validez <= fecha_actual

3. Una vez que se han insertado en el repositorio de certificados vencidos, los registros seleccionados se deberán borrar del repositorio actualizado, para ello se emplea la siguiente instrucción SQL:

delete from repositorio_actualizado where periodo_validez <= fecha_actual

El procedimiento *proc_actualiza* finaliza, cuando se hayan borrado los registros del repositorio de certificados actualizados.

La figura 5.3 ejemplifica el proceso de actualización de los repositorios. Primero se muestra el procedimiento almacenado que se ejecuta en el repositorio A, que contiene los certificados actualizados. Posteriormente el procedimiento, con la instrucción de SQL *select*, detecta que el período de validez del certificado $Cert_{Oscar,Jerry}$ venció, y por lo tanto debe ser trasladado al repositorio V, que contiene a los certificados vencidos. Para ello se ejecuta la instrucción SQL *insert* que añade el certificado $Cert_{Oscar,Jerry}$ al repositorio V, por último se elimina el certificado $Cert_{Oscar,Jerry}$ del repositorio A y finaliza el procedimiento almacenado.

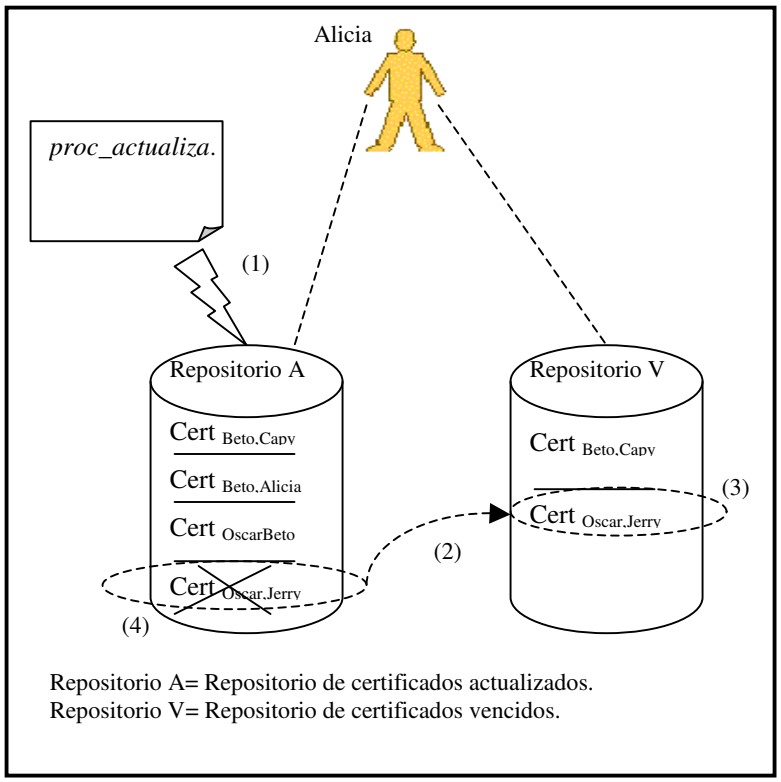


Figura 5.3. Proceso de actualización de certificados vencidos.

La figura 5.4 muestra el diagrama de flujo del procedimiento de actualización.

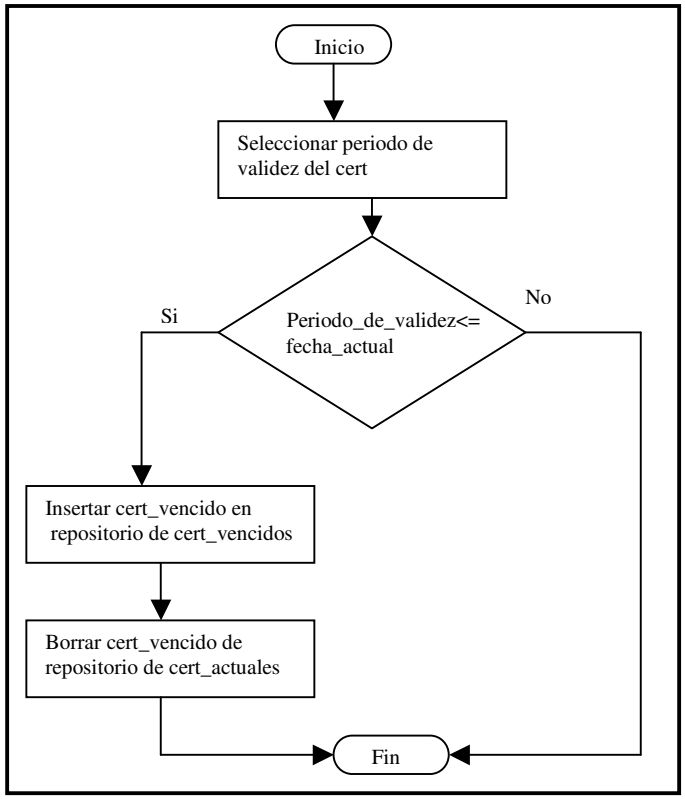


Figura 5.4. Diagrama de flujo del proceso de actualización de certificados.

5.3 MECANISMO PARA REVOCACIÓN Y RENOVACIÓN DE CERTIFICADOS

El siguiente problema que se presenta, es el referente al proceso de revocación de llaves. En PGP [12,14] existe la revocación explícita, en la cuál, un usuario revoca un certificado de forma manual de su repositorio. En MANETS esto también es válido, pero adicionalmente se propone crear un mecanismo, para que sea en forma implícita. De esta forma, el usuario no tendrá que dedicar tiempo en revocar certificados vencidos en forma manual.

Para ello es necesario contar con los dos repositorios que se mencionaron (repositorio de certificados actualizados y repositorio de certificados vencidos). Se va a contar con un proceso que verifique el repositorio de certificados vencidos y comprobar si el certificado del usuario tiene una antigüedad mayor a 1 año, si esto es cierto, entonces será revocado del repositorio. Cabe señalar que la antigüedad del certificado, para efectos de está propuesta, se tomará de 1 año, pero puede ser modificada.

Este repositorio de certificados vencidos, permite que si un usuario emisor renueva su certificado y envía un mensaje al usuario receptor en la MANET, el certificado que se encuentra en el

repositorio de certificados vencidos del usuario receptor, se pueda mover al repositorio de certificados actualizados, por medio de un procedimiento almacenado.

Para llevar a cabo la revocación de certificados, se creó un procedimiento denominado *proc_revoc,* el cuál se basa en el campo de período de validez del repositorio de certificados vencidos para verificar las fechas de vencimiento del certificado. El procedimiento almacenado está conformado por las siguientes instrucciones:

1. Para seleccionar el campo de período de validez, se empleará la siguiente sentencia de SQL:

   select periodo_validez from repositorio_vencidos where (periodo_validez-fecha_actual)>1

   Si el resultado de la sentencia anterior es *null*, entonces todos los certificados del repositorio no pueden ser revocados en forma permanente y el procedimiento termina,
   en caso contrario continua con el paso 2.

2. Con la sentencia *SQL* anterior, el resultado arroja aquellos certificados cuyo período de validez es mayor a un año. Con lo anterior se pueden eliminar los certificados seleccionados del repositorio de certificados vencidos. Para ello se utiliza la siguiente instrucción *SQL*:

   delete from repositorio_vencidos where(select periodo_validez from repositorio_actualizado where(periodo_validez-fecha_actual)>1)

Una vez que se ejecuta la sentencia anterior, el procedimiento *proc_revoc* finaliza.

La figura 5.5 ejemplifica el proceso de revocación en el repositorio de certificados vencidos.

Cuando se ejecuta el procedimiento almacenado en el repositorio V, seleccionan aquellos certificados cuyo período válido sea mayor a un año, utilizando la instrucción SQL *select*. Después el procedimiento detecta que el período válido del certificado $Cert_{Oscar,Jerry}$ ha vencido y debe ser revocado, y selecciona todos los certificados que son candidatos a ser revocados. Una vez que ha seleccionado todos los certificados no válidos, el procedimiento los elimina en forma definitiva del repositorio, para ello emplea la instrucción SQL *delete*.

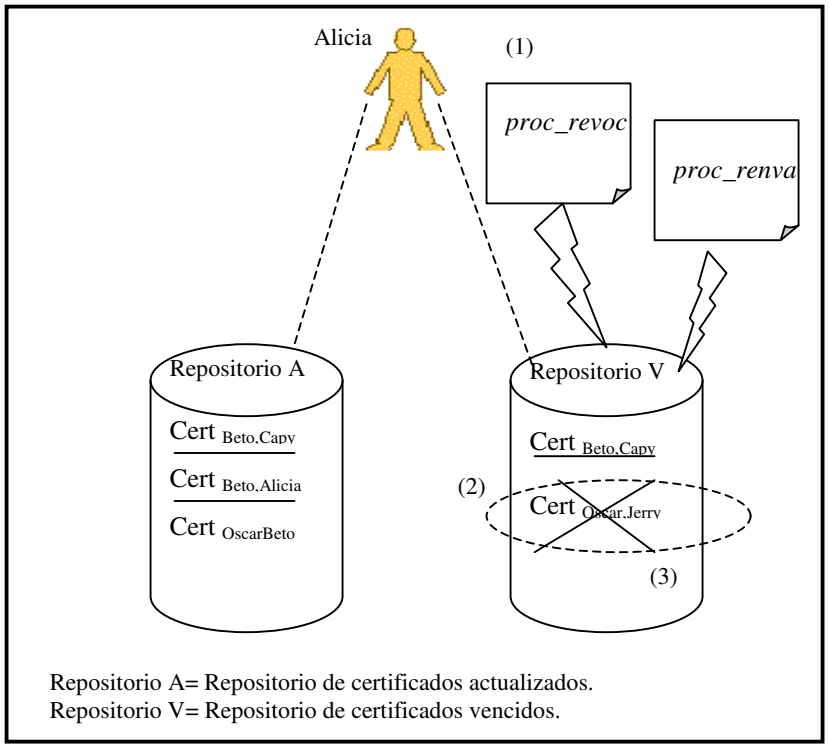


Figura 5.5. Proceso de revocación de certificados vencidos.

La figura 5.6 muestra el diagrama de flujo del procedimiento de revocación.

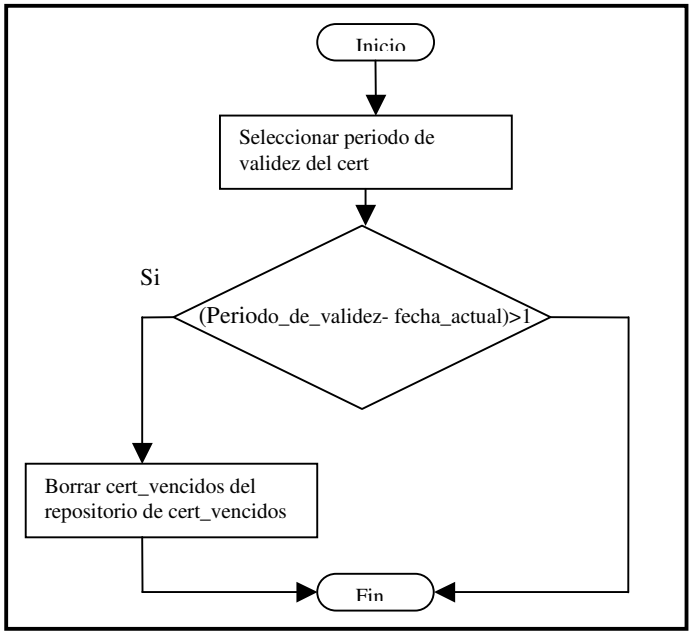


Figura 5.6. Diagrama de flujo del proceso de revocación de certificados vencidos.

Si un usuario recibe una notificación de renovación del certificado de parte del usuario emisor, se ejecuta el procedimiento *proc_renva* llevarán a cabo los siguientes pasos para renovar un certificado.

1. Si llega una petición de renovación del certificado, se verifica que el certificado se encuentre en el repositorio de certificados vencidos, para ello se ejecuta la siguiente sentencia SQL:

   select nombre_emisor from repositorio_vencido where nombre_emisor=nombre_emisor_certificado_actualizado

   Si el resultado de la consulta anterior es null, entonces finaliza *proc_renva*. De lo contrario continua con el paso 2.

2. Si el resultado no es nulo, entonces se inserta el certificado vencido en el repositorio de certificados actualizados y posteriormente se borra del repositorio de certificados vencidos, la siguiente sentencia de SQL se efectúa para tal efecto:

insert into repositorio_actualizado (nombre_usuario,llave_pública,nombre_emisor ,número_serie,periodo_validez) select nombre_emisor from repositorio_vencido where nombre_emisor=nombre_emisor_certificado_actualizado

3. Por último se borra del repositorio de certificados vencidos, ejecutando la siguiente instrucción SQL:

delete from repositorio_vencidos where(select from repositorio_vencido where nombre_emisor=nombre_emisor_certificado_actualizado)

En la figura 5.7 se ejemplifica el proceso de renovación, el usuario Beto desea renovar su certificado *Cert* $_{Beto,Capy}$. Lo primero que realiza Beto es enviarle un mensaje al usuario Alicia (1). Por medio del procedimiento *proc_renva* (2), verifica en su repositorio V de certificados vencidos si tiene el certificado de Beto (3). Para ello emplea la instrucción SQL *select*. Como el certificado si existe en su repositorio, entonces realiza el proceso de renovación, moviendo *Cert* $_{Beto,Capy}$ al repositorio de certificados actualizados (4), ejecutando la instrucción SQL *insert.* Por último lo borra del repositorio de certificados vencidos, para ello se usa la instrucción SQL *delete*.

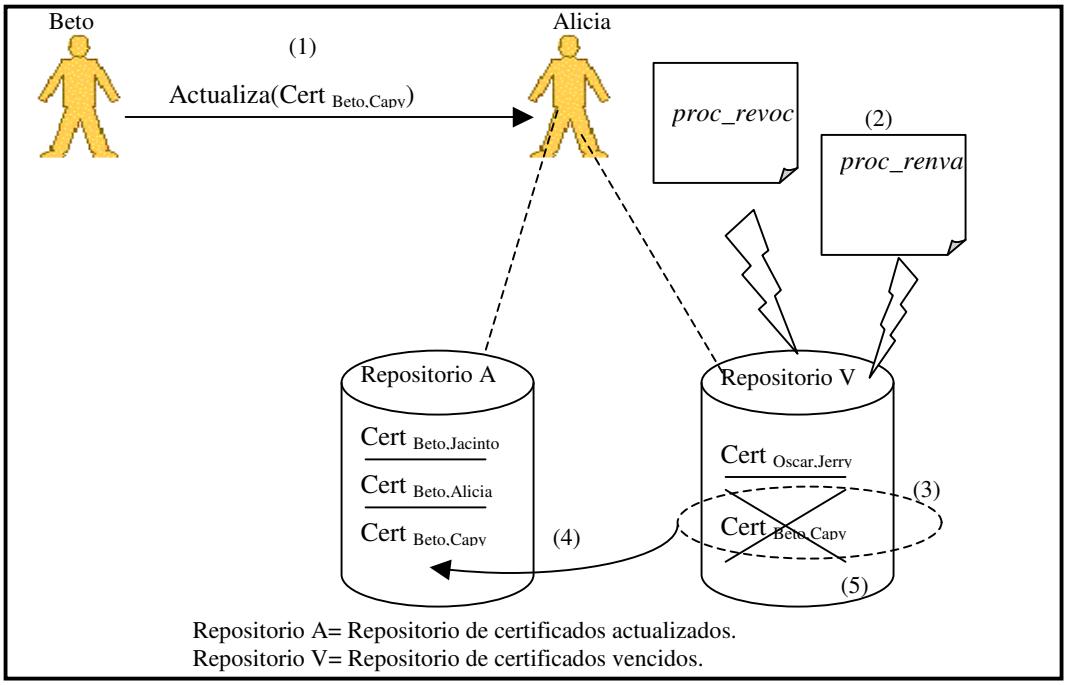


Figura 5.7. Proceso de renovación de certificados.

Cada uno de los procedimientos que se mencionaron, se ejecutan en forma independiente y tienen la ventaja de que el usuario no almacena certificados vencidos, simplificando la administración de los certificados en las MANETS. La figura 5.8 muestra el diagrama de flujo del procedimiento de revocación.

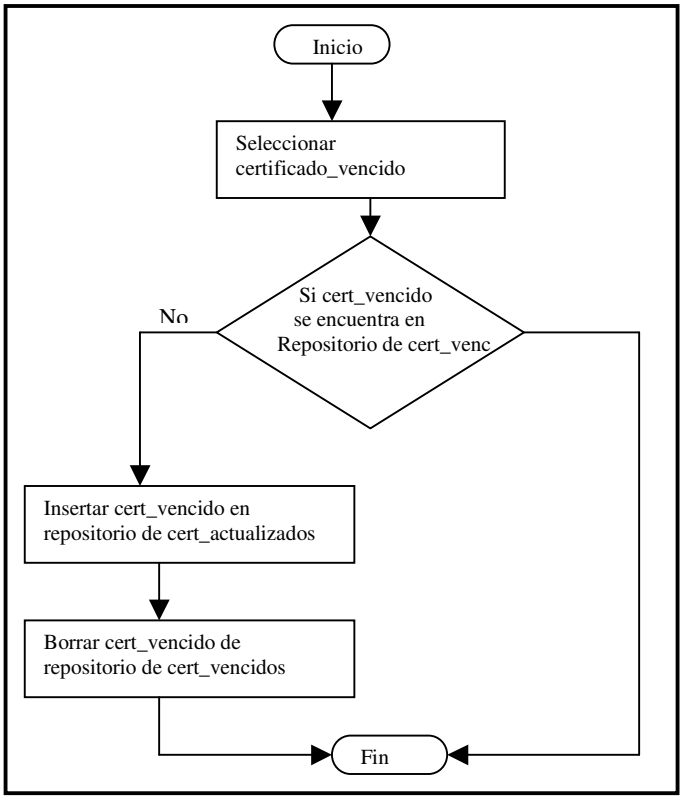


Figura 5.8. Diagrama de flujo del proceso de renovación de certificados vencidos.

Una de las ventajas de nuestra propuesta es que el protocolo 802.11 no se ve afectado ya que no existe una sobrecarga del intercambio de mensajes, debido a que el proceso de actualización se lleva a cabo en cada nodo de forma independiente.

En cuanto a la división de dos repositorios *Hubaux* [20] propone la división de éstos, pero los procesos de revocación y actualización, los lleva a cabo conforme a un intercambio de mensajes. La desventaja que se observa en la solución de *Hubaux* [20], es la sobrecarga de mensajes debido a que se tienen que estar enviando en forma constante, con el objeto de verificar el estado que guardan los repositorios y el de los nodos vecinos.

En está propuesta de solución, esto no ocurre porque el envío de mensajes no se incrementa porque las búsquedas y actualizaciones de los certificados se realizan en forma local.

5.4 COMPARATIVO DE PACDRE CON RESPECTO A LAS SOLUCIONES PRESENTADAS

Diferentes soluciones se han presentado para el problema de administración de llaves en MANETS. PACDRE resuelve el problema de renovación, revocación y actualización de certificados como una mejora para la solución PGP. [12,14]

A continuación se realiza un comparativo de las soluciones presentadas y nuestra solución. Los factores de comparación son: tipo de CA que utilizan (centralizada, semicentralizada o descentralizada), esquema de revocación, renovación de certificados y verificación de certificados vencidos.

La solución de *Autoridad Certificadora Parcialmente Distribuida* [5] no emplea un esquema de revocación ni de verificación de certificados, pero si cuenta con un mecanismo para renovación de certificados. La administración de certificados es centralizada. Por otro lado nuesta solución PACDRE no centraliza la administración de certificados, debido a que cada nodo es responsable de los certificados que resguarda y que emite. Como nuestra solución tiene un mecanismo de renovación de certificados pero no cuenta con ninguno para revocar ni verificar certificados.

La solución *Completamente Distribuida [6]* utiliza un mecanismo para renovar, y revocar certificados, pero no puede verificar que certificados vencidos. La administración de certificados es semi-centralizada. Como se explicó anteriormente PACDRE no centraliza la administración de certificados, pero si cuenta con un mecanismo como nuestra solución para renovar y revocar certificados.

La solución de *Pebblenets* [17] no puede revocar ni verificar certificados, pero puede efectuar el proceso de renovación de certificados. La administración de certificados es descentralizada. Nuestra solución cuenta con los mecanismos para verificar y revocar certificados y al igual que *Pebblenets* [17] la administación de certificados en PACDRE no es centralizada.

La solución de *Identificación Demostrativa* [18] no cuenta con ningún mecanismo para realizar el proceso de renovación, revocación ni verificación de certificados. La administración de certificados no es centralizada. PACDRE no centraliza la administración de certificados y cuenta con los mecanismos para renovar, revocar y actualizar certificados.

Tabla 5.1. Comparativo entre las soluciones que emplean revocación de certificados.

Solución	**Revocación de llaves o certificados**	**Renovación de Certificados o llaves**	**CA centralizada**	**Verificación de certificados o llaves vencidas**
CA parcialmente distribuida	No	Si	Si	No
CA completamente distribuida	Si	Si	Semi-centralizada	No
PACDRE	Si	Si	No	Si
Pebblenets	No	Si	Sí	No
Identificación demostrativa	No	No	No	Si

De la tabla 5.1 se puede apreciar que PACDRE es la única solución que contempla que el manejo de los certificados en MANETS se pueda llevar a cabo sin que el usuario se preocupe por la administración de éstos.

PACDRE además de fortalecer la seguridad en las MANETS, no representa una sobrecarga de mensajes adicional al protocolo de comunicación 802.11. A diferencia de otras soluciones, que contemplan que los procesos de administración de certificados intercambien un gran número de mensajes, PACDRE reduce el intercambio de mensajes debido a que las búsquedas siempre se realizan en forma local en los repositorios de los nodos.