Xarxes emergents

MANET, WSN, DTN

Helena Rifà Pous
Sergi Robles Martínez
Joan Borrell Viader

PID_00195516

UOC
Universitat Oberta de Catalunya

Índex

Introducció 5

Objectius 6

1. Xarxes emergents 7
1.1. Descripció de les xarxes 7
1.2. Vulnerabilitats 8

2. Xarxes ad hoc 12
2.1. Descobriment dels nodes veïns 12
2.1.1. TESLA 14
2.2. Encaminament 15
2.2.1. DSR 16
2.2.2. AODV 22
2.2.3. OLSR 26
2.3. Privadesa 29
2.3.1. Anonimat dels subjectes 29
2.3.2. Desvinculació de missatges 31
2.3.3. Indetectabilitat 33

3. Xarxes de sensors 35
3.1. Gestió de claus 35
3.1.1. Establiment de claus a partir d'una clau mestra de curt termini 37
3.1.2. Piscines de claus 38
3.1.3. Predistribució aleatòria de claus basada en polinomis 41
3.1.4. Predistribució de claus basada en matrius 43
3.2. Agregació de dades 45
3.2.1. Arbre de Merkle 46

4. Protocols tolerants a retards i a interrupcions 48
4.1. Arquitectura DTN proposada pel DTNRG 49
4.2. Arquitectura col·lapsada Haggle 51
4.3. Arquitectura DTN amb missatges actius 52
4.3.1. Missatges actius 53
4.3.2. Un exemple pràctic: PROSES 55
4.4. Investigacions en marxa i problemes oberts 56

5. Problemes de seguretat en protocols DTN 58
5.1. Gestió de claus 58
5.1.1. Solucions basades en IBC 59
5.1.2. Solucions basades en SPKI/SDSI 60
5.1.3. Revocació de claus 61
5.2. Encaminament 61
5.2.1. No-repudi 62
5.3. Control d'accés en els nodes DTN 64

Resum 66

Activitats 67

Exercicis d'autoavaluació 67

Solucionari 68

Glossari 69

Bibliografia 70

Introducció

En aquest mòdul didàctic presentem alguns dels principals problemes de seguretat al voltant de les noves xarxes que estan sorgint en els últims anys. Es tracta principalment de xarxes sense fils, distribuïdes i fins i tot autoorganitzades, que estan formades per dispositius mòbils i amb limitacions de recursos molt més severes que les computadores que s'utilitzen en les xarxes tradicionals.

Comencem el primer apartat introduint què entenem per xarxes emergents i quines són les noves vulnerabilitats de seguretat d'aquest tipus de xarxes.

Tot seguit, en el segon apartat, ens centrem en les xarxes ad hoc i descrivim els mecanismes emprats per a protegir les operacions de construcció de la xarxa. Veiem com es pot implementar un procés de descobriment de veïns que en garanteixi la identitat, i com es pot establir la informació d'encaminament (taules d'encaminament o *caches*) correcta per a poder enviar paquets a través de la xarxa. Finalment, en aquest apartat també descrivim quins són els riscos en la privacitat que tenen aquestes xarxes i com podem minimitzar-los.

En el tercer apartat estudiem les característiques de les xarxes de sensors i veiem com es poden resoldre dos dels seus principals problemes de seguretat: la gestió de claus i l'agregació de dades. Les limitacions dels sensors fan que totes les solucions hagin de basar-se en operacions criptogràfiques molt lleugeres, com funcions resum (*hash*).

Acabem el mòdul presentant els protocols tolerants a retards i a interrupcions (*Delay- and Disruption-Tolerant Networking*, DTN), destinats a les situacions en què, o bé no hi ha un canal continu entre les parts que es comuniquen, o bé els retards en les comunicacions són molt elevats. En l'enfocament DTN les comunicacions es fan exclusivament a nivell d'aplicació, entre nodes adjacents quan aquests tenen l'oportunitat de comunicar-se, i sense necessitat de connectivitat contínua d'extrem a extrem. Aquest mètode permet les comunicacions fins i tot en les situacions més adverses, però a costa de perdre la interactivitat, ja que cada pas d'una comunicació pot prendre un temps no limitat. Els principals problemes de seguretat que tenen plantejats els protocols DTN són la gestió de les claus criptogràfiques i l'encaminament de la informació.

Objectius

Els materials didàctics d'aquest mòdul han de permetre que l'estudiant assoleixi els objectius següents:

1. Entendre els problemes de seguretat de les xarxes emergents.
2. Comprendre les característiques dels algorismes de seguretat que s'utilitzen en entorns on el costos computacionals, d'energia i de transmissió de dades a través de la xarxa són realment importants.
3. Conèixer el funcionament de diversos mecanismes i protocols de seguretat dissenyats per a xarxes ad hoc, de sensors i protocols tolerants a retards i interrupcions.

1. Xarxes emergents

En aquest apartat resumirem les característiques bàsiques de les xarxes emergents i descriurem quines són les vulnerabilitats a què són susceptibles aquestes xarxes atesa la seva naturalesa bàsica de funcionament.

1.1. Descripció de les xarxes

La infraestructura d'una xarxa tradicional, és a dir, el conjunt d'elements que conforma la xarxa, excepte els terminals finals, és propietat d'una empresa o operadora que dóna servei de connexió als seus usuaris. Els usuaris que fan ús de la xarxa confien en l'operadora i assumeixen que aquesta els donarà un bon servei de transmissió i recepció de paquets.

En els últims anys la tecnologia sense fils ha baixat molt de preu, cosa que ha facilitat el desenvolupament de sistemes basats en aquesta tecnologia i l'adquisició d'equipament per part dels usuaris. L'increment de terminals sense fils ha fet sorgir les xarxes sense fils basades en una topologia en malla*. En una topologia en malla cada node està connectat a un dels altres nodes o a més d'un. La infraestructura d'aquestes xarxes pot ser descentralitzada, sense servidor central ni el suport d'una operadora, o centralitzada.

* En anglès, *Wireless Mesh Networks* (WMN).

En el cas de xarxes WMN descentralitzades els nodes utilitzen la implementació del mode ad hoc de l'estàndard IEEE 802.11. És per aquest motiu que també s'anomenen **xarxes ad hoc**.

En les xarxes ad hoc la comunicació es basa en la cooperació d'un gran nombre de dispositius individuals sense fils que permeten que un missatge acabi arribant, salt a salt, d'un punt a un altre de la xarxa. Cada usuari ha de tenir unes capacitats d'encaminament: una mena de cursa de relleus en la qual la densitat d'usuaris aconsegueixi que el relleu representi un esforç menor.

MANET

Les MANET (Mobile Ad-hoc Network) són xarxes ad hoc generades per l'autoconfiguració de nodes encaminadors que formen una topologia arbitrària. Com que els nodes es poden moure lliurement es considera que la topologia de xarxa canvia ràpidament i de manera impredictible.

Les xarxes WMN centralitzades corresponen a una barreja de topologies ad hoc i infraestructura (o mode amb punt d'accés). Es tracta de xarxes amb infraestructura que, a més a més, permeten la unió de dispositius que estan fora del rang de cobertura dels punts d'accés, però dins del rang de cobertura d'algun punt d'accés de trànsit*. Els TAP són nodes especials controlats per l'operadora que permeten la retransmissió dels paquets dels clients més enllà del seu radi de cobertura.

* En anglès, *Transit Access Points* (TAP).

Un dels altres tipus de xarxes que s'han estès en els últims anys són les xarxes de sensors*. Es tracta de xarxes sense fils formades per dispositius autònoms que poden treballar de manera cooperativa per monitoritzar les condicions físiques o ambientals (per exemple, temperatura, so, vibració, pressió, moviment, pol·lució) d'una zona.

*** En anglès, *Wireless Sensor Network* (WSN).**

Les **xarxes de sensors** es poden considerar un subconjunt molt especialitzat de les WMN en què els nodes tenen recursos molt limitats (són petits nodes sensors) i estan orientats a una tasca molt específica de monitorització.

En la literatura de les WSN s'utilitza generalment una nomenclatura diferent de la utilitzada en WMN, en la qual els TAP es denominen *estacions base* o *supernodes*. Les estacions base són, per tant, nodes de la WSN amb capacitat computacional ampliada i més recursos quant a bateria i comunicacions.

Els protocols DTN* han sorgit recentment per donar una solució de comunicació en situacions en què o bé no es pot garantir una connectivitat d'extrem a extrem o bé els retards en les comunicacions són tan elevats que els protocols actuals d'Internet no són aplicables.

*** De l'anglès, *Delay- and Disruption-Tolerant Network.***

Les situacions a què s'apliquen els protocols DTN són molt variades, des de comunicacions interplanetàries fins a xarxes WSN (o MANET) amb connectivitat intermitent o nul·la, ja sigui per l'elevada mobilitat dels seus nodes, per la inexistència d'una infraestructura de comunicació o per l'excessiva distància que en separa els nodes.

Temporitzadors

Recordem que el protocol TCP té diversos temporitzadors per a controlar la transmissió de la informació, amb valors màxims de pocs minuts, i que assumeix una taxa d'error molt baixa.

1.2. Vulnerabilitats

Les característiques més rellevants de les xarxes emergents són el fet que es basen en tecnologies de comunicació sense fils (normalment ràdio, encara que també es considera l'ús d'infrarojos o ultrasons), i que la cooperació dels usuaris és un dels punts clau per a fer-les atractives i viables.

Les xarxes sense fils faciliten l'accés al medi de l'atacant i, per tant, són especialment vulnerables al següent:

- **Escoltes no autoritzades (*eavesdropping*)**. Col·locant una antena de recepció en el lloc adequat, un atacant pot escoltar la informació que s'envia o rep pel medi ràdio. L'atac d'escoltes no autoritzades es classifica en la categoria d'atacs passius, que consisteixen a escoltar el medi i analitzar les

dades capturades sense interactuar amb el medi. Els atacs passius es poden evitar, i se sol fer-ho, mitjançant el xifrat de la informació intercanviada.

- **Alteració de les dades**. Es tracta d'un atac actiu en que una entitat maliciosa modifica el contingut dels missatges que s'intercanvien dues entitats o més. Normalment, aquests atacs es realitzen mitjançant l'atac de l'home a mig camí *(Man-In-The-Middle*, MITM), en el qual l'atacant es converteix en una passarel·la entre dues entitats i, per tant, té control absolut sobre el trànsit entre elles.

- **Atacs de suplantació d'identitat (*spoofing*)**. Un atacant es fa passar per un altre node per poder tenir accés a zones restringides o limitades a què només pot accedir un node autoritzat.

- **Atacs per múltiples falses identitats (*sybil*)**. En aquest cas un atacant agafa de manera deliberada múltiples identitats legítimes de la xarxa (roba les dades d'altres usuaris o en genera de noves) amb l'objectiu de tenir una gran influència en la xarxa o grup, per exemple, en serveis en què s'usen votacions, reputacions, etc.

- **Negació del servei (*DoS*)**. Aquest tipus d'atac passa en l'àmbit físic, quan un node maliciós envia indiscriminadament missatges que consumeixen l'amplada de banda disponible de la xarxa, i aconsegueix la indisponibilitat temporal o permanent d'un servei. En xarxes sense fils hi ha bàsicament dos tipus d'atacs DoS:

– **Bombes electròniques**, en les quals l'atacant genera un senyal de gran potència pel mateix mitjà de comunicació, de manera que impedeix que hi hagi intercanvi de missatges. Aquest tipus d'atac és difícil d'evitar, encara que es pot pal·liar una mica utilitzant tècniques d'espectre eixamplat* i salt de freqüències**.

*** En anglès, *spread spectrum*.**
**** En anglès, *frequency hopping*.**

– **Inundació de dades**, en què els nodes compromesos envien grans quantitats d'informació a altres nodes a fi de saturar-los. En aquest cas, els atacs es poden evitar mitjançant sistemes de detecció d'intrusos*** i el control posterior d'accés que inhabiliti (deixi fora del grup segur) el node que inunda/ataca.

***** En anglès, *Intrussion Detection System* (IDS).**

D'altra banda, el fet que els nodes de les xarxes sense fils puguin ser dispositius petits i mòbils també introdueix certes vulnerabilitats en el sistema:

- **Atacs físics**. Els usuaris maliciosos tenen accés físic als nodes de la xarxa i poden efectuar atacs, com ara destruir nodes, capturar nodes i substreure'n informació sensible o comprometre'n les claus criptogràfiques per a transmetre informació falsa a la xarxa.

- **Localització**. En molts casos, per a garantir la mobilitat d'un dispositiu se n'ha de traçar la ubicació, cosa que es pot considerar com un atac a la privacitat.

- **Limitació de recursos**. Generalment els dispositius mòbils són petits i, per tant, de recursos més limitats de potència, emmagatzematge i bateria. De les tres limitacions, la de bateria és la més representativa, ja que el seu progrés tecnològic és molt més lent que el de processadors i memòries. Normalment les limitacions de bateria porten a reduir el nombre d'operacions computacionals que realitza el dispositiu sense fils, cosa que sol derivar en una pobra implementació de protocols de seguretat.

I finalment, les xarxes que es basen en la cooperació dels seus nodes es poden veure perjudicades per l'existència de nodes no col·laboradors. Els nodes que no col·laboren correctament amb la xarxa amb l'objectiu de malmetre-la o perjudicar-ne els integrants són nodes maliciosos. Els que no col·laboren per tal d'estalviar recursos i alhora fan ús dels serveis de la xarxa a expenses dels altres usuaris són nodes egoistes.

Maliciosos i egoistes

Utilitzant la terminologia de nodes maliciosos i egoistes podríem definir que un dissenyador de virus és maliciós, mentre que un emissor de correu brossa (*spammer*) és egoista.

Els nodes maliciosos i egoistes poden atacar els protocols col·laboratius de la xarxa, en particular els protocols d'encaminament. En xarxes descentralitzades on tots els usuaris col·laboren en l'encaminament de paquets, els atacs a l'encaminament degraden el servei de manera indirecta per inhabilitació de les comunicacions:

- **Disrupció de rutes**. Un atacant evita que una ruta entre dos nodes sigui descoberta. L'objectiu és degradar la qualitat de servei de la xarxa.

- **Desviació de rutes**. En aquest cas l'adversari no evita que es creï una ruta, però aconsegueix que s'estableixi de manera diferent a la que marca el protocol (no es creen les rutes més curtes, ràpides, fiables, etc.). L'objectiu és incrementar el control de l'adversari sobre algunes víctimes, i poder escoltar o fins i tot modificar els paquets de dades que envia.

Hi ha diferents maneres d'implementar aquests atacs, però en podem destacar les següents:

- **Atacs *blackhole/sinkhole***. En aquest cas un node atacant es presenta com una bona opció per a l'encaminament i es converteix en un node atractiu pel qual passen gran part dels paquets de la xarxa. Un cop el node ha aconseguit atraure el tràfic de la xarxa elimina els paquets, amb la qual cosa provoca una disrupció de rutes (atac *blackhole*) o utilitza el tràfic per a fins maliciosos (atac *sinkhole*).

- **Atacs *warmhole***. En aquest cas un node atacant crea un túnel ocult entre dues parts diferents d'una xarxa descentralitzada i hi encamina certs missatges de control d'una zona a una altra amb l'objectiu de distorsionar els mecanismes d'encaminament.

- **Atacs al control d'errors**. En aquest cas un atacant envia missatges d'error falsos que invalidin l'establiment correcte d'una ruta.

- **Atacs de re-injecció de paquets**. En aquest cas un node envia paquets d'encaminament antics amb informació que en el seu moment era correcta però que ara ja ha quedat desfasada i, per tant, és invàlida.

- **Atacs de generació i modificació de paquets**. En aquest cas un node crea o modifica paquets amb informació d'encaminament falsa.

En els apartats vinents veurem com aquestes vulnerabilitats afecten el bon funcionament de les xarxes ad hoc i de sensors.

Per al cas dels protocols DTN, a part de les vulnerabilitats comentades, cal tenir en compte algunes altres especificitats, com ara el fet que perd sentit qualsevol infraestructura centralitzada per a la gestió de les claus criptogràfiques, atesa la impossibilitat d'interactuar de manera continuada amb el centre de la infraestructura. De manera relacionada, qualsevol protocol clàssic basat en una negociació de diversos passos* deixa de poder utilitzar-se en aquests entorns.

Vegeu també

Els problemes dels protocols DTN s'estudien en l'apartat 5 d'aquest mòdul.

*** Com el protocol TLS (11).**

2. Xarxes ad hoc

La característica més rellevant de les xarxes ad hoc és el fet que els nodes de la xarxa tenen dos rols: actuar com a terminals finals i realitzar les funcions d'encaminament.

Atès que els nodes són terminals personals, els usuaris en poden manipular el funcionament per tal que actuïn únicament buscant el benefici del propietari i perjudicant, si escau, la resta de la xarxa. D'altra banda, com que es tracta de terminals sense fils, mòbils i relativament petits, tots els nodes són vulnerables a atacs d'usuaris maliciosos.

En aquest apartat estudiarem els problemes de seguretat de les xarxes ad hoc en les operacions que li són més pròpies i característiques, és a dir, en les operacions de gestió de l'encaminament de la xarxa a partir de terminals finals. En primer lloc, veurem com podem assegurar d'una manera eficient que els veïns d'un node són qui diuen ser, després analitzarem el funcionament dels protocols d'encaminament i finalment veurem quins són els riscos de privacitat d'aquestes xarxes i com podem minimitzar-los.

2.1. Descobriment dels nodes veïns

Un dels principals problemes de treballar amb xarxes dinàmiques en les quals els nodes canvien eventualment d'estat (actius/inactius), posició i condicions d'entorn, és la complexitat que tenen els nodes origen per a trobar els nodes o serveis destinacions amb qui es volen connectar.

El primer pas per a construir rutes entre nodes remots és conèixer quins són els terminal veïns, és a dir, aquells amb els quals hi ha connexió directa.

El descobriment de veïns es pot fer a través de protocols molt simples en què un node envia una petició de descobriment en mode difusió (*broadcast*), i tots els nodes que reben aquest paquet responen. Les respostes que rep el node emissor corresponen als nodes que estan en el seu rang de cobertura i, per tant, amb qui hi ha connexió directa.

Els protocols de descobriment de veïns sovint s'anomenen *"Hello protocols"*, i els missatges enviats són *"Hello messages"*.

Un dels requisits desitjables dels protocols de descobriment de veïns és la seva capacitat d'enviar missatges en difusió que puguin ser autenticats. És a dir, un emissor ha de ser capaç d'enviar un missatge i tots els receptors n'han de poder verificar l'autenticitat, alhora que s'ha d'impedir que aquests puguin utilitzar la informació rebuda per fer atacs.

La solució tradicional d'usar signatures digitals per a l'autenticació de missatges no és aplicable en aquests entorns, atès el gran volum de missatges *Hello* que es gestionen en una xarxa ad hoc, i les característiques limitades dels dispositius que solen formar part d'aquestes xarxes (dispositius empotrats, mòbils, etc.). És per aquest motiu que es necessita trobar alternatives que no es basin en operacions criptogràfiques complexes com les de clau pública.

Una de les propostes més destacades per a l'autenticació d'una seqüència de missatges en difusió és el protocol TESLA. La robustesa de TESLA es basa en dos elements:

Lectura complementària

El protocol TESLA està definit en una RFC: **A. Perrig; D. Song; R. Canetti; J. D. Tygar; B. Briscoe** (2005). "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction". *RFC 4082* (*informational*) (juny).

1) Cadenes resum (*hash*). Una cadena resum és una seqüència de valors tal que cada element és el resultat d'aplicar una funció resum a l'element immediatament anterior. Una de les seves propietats principals és la unidireccionalitat, ja que els seus elements es poden calcular molt fàcilment en una direcció, però no en la direcció contrària.

2) Signatures MAC. Una signatura MAC té dues entrades -un missatge i una clau secreta- i produeix una sortida que permet verificar la integritat i l'autenticitat del missatge. Qualsevol canvi en el missatge o en la clau secreta resultarà en la generació d'un resultat diferent.

Construcció i ús de les cadenes resum

Una cadena resum es construeix de darrere cap endavant. Inicialment es determina la longitud N de la cadena i s'escull un valor aleatori que anomenarem v_N. Aleshores, es poden calcular la resta de valors de la cadena aplicant una funció resum de manera recursiva: $v_i = h(v_{i+1}) = h^{(N-i)}(v_N)$.

L'ús dels valors de la cadena resum es fa en ordre creixent, des de v_0 fins a v_N (en l'ordre contrari en què s'han generat els valors). Observeu que, donat v_i, és computacionalment eficient calcular $v_j = h^{(i-j)}(v_i)$ per a qualsevol $j > i$, però no és possible calcular v_k amb $k > i$. Així doncs, l'ús i la revelació d'un valor v_i de la cadena no desvetlla cap informació sobre els valors següents de la cadena.

Exemple

Un dels primers protocols que usa les cadenes resum és l'esquema d'autenticació per contrasenya de Lamport. Considerem un escenari en què un usuari es vol logar en un ordinador remot. Per fer-ho ha d'enviar la constrasenya a través d'un canal insegur. Per impedir que un intrús intercepti la contrasenya i la pugui utilitzar més tard, l'usuari té un joc de claus $\{x_0,x_1,...,X_{1000}\}$ emmagatzemat a l'equip; cada vegada que utilitza una contrasenya, aquesta esdevé invàlida per a ser usada mai més.

Per a minimitzar els requisits d'emmagatzematge de l'usuari i l'equip remot, es proposa que el valor x_{1000} sigui escollit per l'usuari, i que els altres valors es defineixin com a

$x_i = f(x_{i+1})$ per alguna funció f resum resistent a les preimatges. Per a iniciar el sistema només és necessari emmagatzemar x_{1000} i l'índex de la contrasenya que s'utilitzarà en la connexió següent.

A continuació veurem amb més detall el funcionament del protocol TESLA.

2.1.1. TESLA

L'autenticació de missatges en multicast o *broadcast* requereix una font d'asimetria, de manera que els receptors puguin verificar l'autenticitat dels missatges que reben però no puguin utilitzar aquesta informació per a enviar ells mateixos dades suplantant la font d'informació autèntica. TESLA utilitza el temps per a crear aquesta asimetria. Tots els participants en una comunicació han d'estar lleugerament sincronitzats (dins d'un cert marge de tolerància, totes les parts s'han de posar d'acord en el temps actual). Aleshores, el protocol TESLA opera de la manera següent:

Cadena resum

La cadena resum s'assigna a un segment de temps en l'ordre invers a la seva generació, de manera que qualsevol valor d'un interval pot ser utilitzat per a obtenir els valors dels intervals de temps anteriors però no dóna cap informació dels valors dels intervals posteriors.

1) L'emissor divideix un segment de temps en intervals uniformes.

2) L'emissor construeix una cadena resum i assigna els valors de la cadena $v_0,v_1,...,v_N$ als intervals de temps de manera seqüencial (un valor per a cada interval de temps). Durant la inicialització del sistema, l'emissor envia el valor v_0 a través d'un sistema d'autenticació forta tradicional, com pot ser una firma digital.

3) Quan l'emissor vol enviar un paquet, calcula el MAC dels seus continguts utilitzant com a clau criptogràfica el valor v_i associat a l'interval de temps en què està operant. Després envia el paquet de dades, el MAC, i una referència a l'interval de temps en el qual ha estat operant.

4) L'emissor fa públic el valor v_i utilitzat en l'interval t_i després d'un marge predefinit de temps (per exemple, la clau usada en l'interval de temps t_i es revela en el temps t_{i+2}).

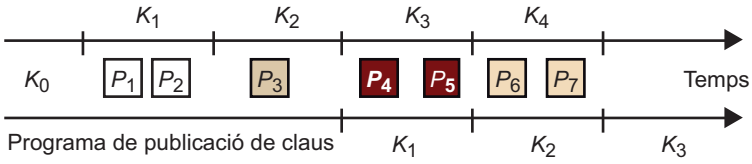
5) Quan un receptor rep un missatge de dades comprova que la clau utilitzada per a calcular el MAC encara és secreta (això ho pot comprovar perquè té el rellotge sincronitzat amb l'emissor i sap en quins intervals de temps es publiquen les claus). Si la clau encara és secreta, tan aviat com la rep comprova que la clau és autèntica, és a dir, que realment pertany a l'emissor que diu. Això ho pot fer aplicant recursivament una funció resum sobre la clau v_i rebuda fins a trobar el valor v_0 que ha estat autenticat per l'usuari a través d'una firma digital, o fins a trobar un altre valor v_j, amb $j < i$, que ja hagi estat verificat prèviament pel receptor.

Un cop es publica la clau v_i, un adversari que monitoritzés la comunicació podria usar la clau per a fabricar missatges falsos. Tanmateix, aquests missatges no serien acceptats pel sistema perquè els usuaris coneixen que en l'interval de temps actual t_j l'ús de la clau v_i, $i < j$, ja ha caducat.

La figura 1 mostra un exemple de com s'usen les claus de TESLA per a firmar diferents paquets, i com es revelen al cap de cert temps. Els paquets P_1 i P_2 s'autentiquen utilitzant K_1. La clau K_1 no es revela fins al cap de dos intervals de temps, i quan es revela, aquesta clau ja no és vàlida, la que és vàlida és la clau K_3.

Hi ha una versió més lleugera del protocol TESLA, anomenada μ-TESLA, adequada per a xarxes de sensors.

Figura 1. Autenticació TESLA



El protocol TESLA és apropiat per a xarxes de comunicació sense fils que tenen una fiabilitat menor que les xarxes cablejades. Observeu que si durant algun interval de temps es perden les claus i no es publiquen, el receptor podrà recuperar les claus de la cadena a partir d'algun valor que rebi posteriorment, i a partir d'aquí podrà verificar l'autenticitat de tots els paquets rebuts.

2.2. Encaminament

Una de les característiques particulars de les xarxes ad hoc és la seva naturalesa multisalt. Els protocols d'encaminament multisalt es poden classificar en tres categories:

- els que es basen en la topologia de la xarxa,
- els que es basen en la posició dels nodes i
- els que tenen una aproximació híbrida entre les dues anteriors.

Els primers tenen els mateixos principis que els protocols tradicionals d'Internet (crear i mantenir taules d'encaminament segons la distribució dels nodes, distribuir la informació d'enllaç, etc.), mentre que els segons utilitzen la localització física dels nodes per a encaminar paquets a la seva destinació. L'avantatge dels protocols basats en la posició és que no necessiten mantenir informació d'encaminament o descobrir rutes explícitament. Els nodes coneixen la seva localització a través d'un servei de posicionament (per exemple, GPS) i obtenen la posició dels altres nodes a través d'un servei de localització. Quan volen enviar un paquet, la font obté la localització de la destinació i inclou aquesta informació a la capçalera del paquet. Després els nodes intermedis prenen decisions d'encaminament basant-se en la seva posició i la localització de la destinació. Tanmateix els protocols basats en la posició tenen el problema que els nodes han de disposar d'un maquinari car, com és el GPS, i a més es vulnera totalment la privacitat dels usuaris d'una comunicació en revelar-se on estan posicionats. És per aquest motiu que els protocols basats en la posició estan relegats a aplicacions molt específiques. La gran majoria de serveis es basen en protocols basats en la topologia i aquests són els que analitzarem en aquest apartat.

El grup de treball en xarxes ad hoc de l'*Internet Engineering Task Force* (IETF), anomenat grup MANET, està desenvolupant diversos protocols d'encaminament basats en la topologia. Aquests protocols estan basats en dos enfocaments diferents:

- **Protocols reactius**. Aquests protocols estableixen les rutes necessàries sota demanda, és a dir, la informació sobre la topologia de la xarxa només s'envia quan és necessari. Quan un node vol comunicar-se amb un altre amb el qual no hi ha una ruta establerta, s'activa el protocol d'establiment per a generar una nova ruta. Els exemples més coneguts d'aquest tipus de protocols són els denominats *Dynamic Source Routing Protocol* (DSR) i *Ad hoc On-Demand Distance Vector* (AODV).

- **Protocols proactius**. Després d'un pas inicial d'establiment, aquests protocols mantenen les rutes actives i "netes" (sense bucles) constantment. Quan un node vol comunicar-se amb un altre ja té a punt la ruta d'enllaç per on enviar paquets. L'exemple més conegut d'aquest model de protocols és l'*Optimized Link State Routing Protocol* (OLSR).

A continuació veurem més detalladament aquests protocols i estudiarem les vulnerabilitats de seguretat que tenen i com es poden solucionar.

2.2.1. DSR

El DSR és un dels primers protocols que es van proposar per a l'encaminament de xarxes ad hoc i és un dels més influents sobre els protocols que han anat sorgint des de llavors.

Lectura complementària

El protocol DSR està definit en una RFC:
D. Johnson; Y. Hu; D. Maltz (2007). "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4". *RFC 4728* (*experimental*) (febrer).

El que defineix el DSR és que es tracta d'un protocol pensat perquè l'encaminament dels paquets l'estableixi la font de dades. Així, cada paquet porta en la seva capçalera una llista dels nodes pels quals ha de passar per arribar a la destinació desitjada. Quan un node rep un paquet, primer verifica si ell n'és el destinatari i si no és així, comprova si està a la llista de transportadors del paquet. Si es tracta d'un node encaminador, retransmet el paquet al proper node de la llista (que ha de ser un veí directe), sinó el paquet es descarta.

Quan s'utilitza encaminament de font, el node origen de la transmissió ha de conèixer la ruta completa fins a la destinació abans de poder-hi enviar cap paquet. En general, els protocols d'encaminament estan formats bàsicament per dos mecanismes:

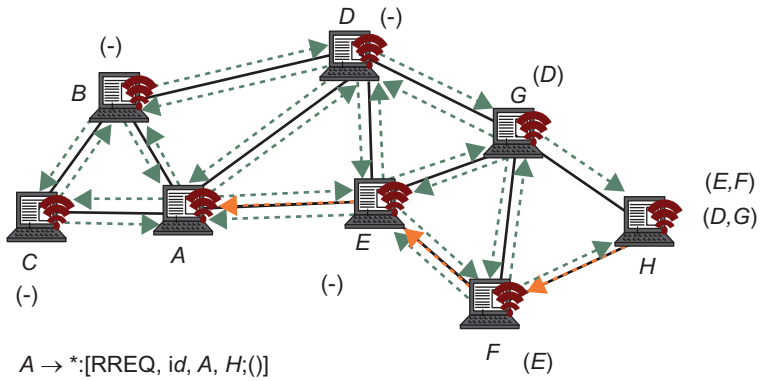
- **Descobriment de rutes**. S'usa quan els nodes no tenen informació de com fer arribar un paquet a la destinació. El mecanisme de descobriment de rutes en DSR es basa en la inundació de tota la xarxa amb un missatge de petició de ruta* i en el retorn de les respostes de ruta**.

*** En anglès, *Route request* (RREQ).**
**** En anglès, *Route reply* (RREP).**

- **Manteniment de rutes**. En aquest cas els nodes de la xarxa ja tenen unes taules d'encaminament amb certa informació de la topologia de la xarxa i com poden encaminar-hi la informació. Aquest mecanisme permet detectar errors de ruta (per exemple, si un enllaç de la ruta ja no funciona perquè un node s'ha mogut o ha sortit de la xarxa).

La figura 2 il·lustra el funcionament del descobriment de rutes en DSR per un node origen *A* i un node destinació *H*. El node origen envia un RREQ amb un identificador de petició (*id*), l'identificador del node origen (*A*), el de la destinació (*H*), i una llista buida de nodes intermedis. Ho envia en *broadcast* a tots els nodes que estan sota la seva cobertura. Cada node que rep una copia de la petició verifica que no l'ha rebut prèviament. Si la petició ja ha estat rebuda, la descarta. En cas contrari, el node afegeix el seu identificador a la llista de nodes pels quals ha passat l'RREQ i reenvia el paquet en *broadcast* als seus veïns. Aquest procediment es repeteix fins que la petició arriba a la destinació *H*.

Figura 2. Descobriment de rutes DSR



$A \rightarrow$ *:[RREQ, id, *A*, *H*;()]
$B \rightarrow$ *:[RREQ, id, *A*, *H*;(*B*)]
$C \rightarrow$ *:[RREQ, id, *A*, *H*;(*C*)]
$D \rightarrow$ *:[RREQ, id, *A*, *H*;(*D*)]
$E \rightarrow$ *:[RREQ, id, *A*, *H*;(*E*,*F*)]
$F \rightarrow$ *:[RREQ, id, *A*, *H*;(*E*,*F*)]
$G \rightarrow$ *:[RREQ, id, *A*, *H*;(*D*,*G*)]

$H \rightarrow A$:[RREP, (camí *F*,*E*); (*E*,*F*)]

Quan *H* rep el paquet, extreu la ruta que l'uneix amb el node *A* de la llista d'identificadors de l'RREQ, la inverteix i la copia en la seva taula d'encaminament. Aleshores genera un RREP copiant la llista d'identificadors de l'RREQ en la resposta. Aleshores, la resposta és enviada per unidestinació (*unicast*) a la font a partir de la ruta que el node ha guardat en la seva taula d'encaminament.

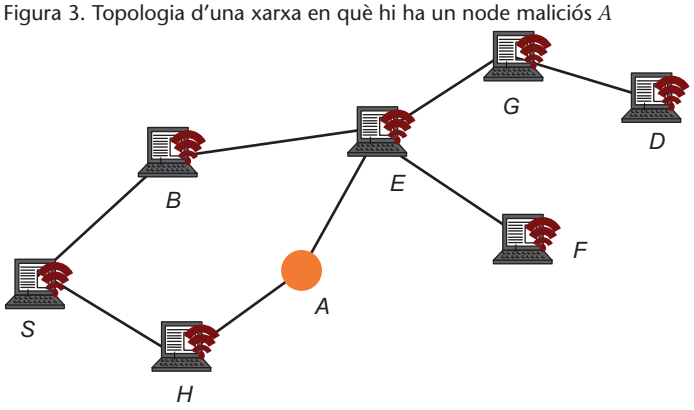
Durant la fase del manteniment de rutes, cada node itermedi d'una comunicació s'ha d'assegurar que el paquet que està enviant realment arriba al salt següent. Això es pot fer requerint que el protocol de la capa d'enllaç enviï un ACK per a cada paquet que s'entregui a un node. També és possible implementar que els nodes escoltin les transmissions dels seus veïns i s'assegurin que el paquet que ells envien és reenviat al seu torn pel node veí.

Vulnerabilitats del DSR

A continuació exposem algunes de les vulnerabilitats més importants del DSR. Utilitzarem la figura 3 per a il·lustrar alguns atacs contra el protocol d'encaminament:

1) Disrupció de rutes. Suposem que un node *S* vol trobar una ruta cap a la destinació *D*. *S* inicia el protocol de descobriment de rutes inundant la xarxa amb un RREQ. Suposem que els nodes tenen la memòria cau buida. El node maliciós *A* pot prevenir el descobriment de la ruta *S*, *H*, *A*, *E*, *G*, *D* eliminant el missatge d'RREQ que prové de *S* o eliminant el missatge de resposta RREP. D'altra banda, el node *A* també pot fer que el node *E* rebi el missatge d'RREQ a partir de *A* abans que a partir de *B*. El node *A* pot, per exemple, mantenir el canal constantment ocupat per evitar que *E* rebi res de *B*. D'aquesta manera l'adversari pot aconseguir que cap de les rutes existents entre *S* i *D* siguin descobertes.

Figura 3. Topologia d'una xarxa en què hi ha un node maliciós *A*



2) Desviament de rutes (atac *sinkhole*). Altra vegada assumim que *S* vol trobar una ruta cap a *D* i inicia el protocol DSR enviant un RREQ. Quan *A* rep la petició de *H*, *A* respon amb una fals RREP que conté la ruta *S*, *H*, *A*, *D*. La ruta falsa és enviada cap a *S*. Com que la ruta falsa és més curta que la descoberta a través del node *E*, *S* decideix usar la ruta *S*, *H*, *A*, *D*. D'aquesta manera l'adversari aconsegueix modificar la ruta natural entre *A* i *D* i escoltar tots els missatges que es transmeten.

3) Creació d'estats d'encaminament incorrectes. En el cas del DSR, crear estats d'encaminament incorrectes significa que l'adversari enganya la font d'un RREQ fent-li acceptar i posar a la memòria cau una ruta inexistent cap a la destinació. Per exemple, quan *S* inicia un descobriment de rutes, *A* pot no reenviar el missatge rebut per *H*. Simplement espera fins que li arriba una altra còpia de la mateixa petició a través del node *E*. Aquesta còpia conté la ruta *S*, *B*, *E*. Després, A genera un missatge RREP que conté la ruta *S*, *B*, *E*, *A*, *D* i l'envia a *E*. La resposta és enviada de tornada cap a *S* via el node *B*, i *S* guarda la ruta *S*, *B*, *E*, *A*, *D*.

Hi ha diversos protocols que afegeixen seguretat al DSR, com ara els següents:

- SRP (on-demand source routing)
- Ariadne (on-demand source routing)
- endairA (on-demand source routing)
- SADSR (security aware adaptive DSR)
- SRDP (secure route discovery protocol)

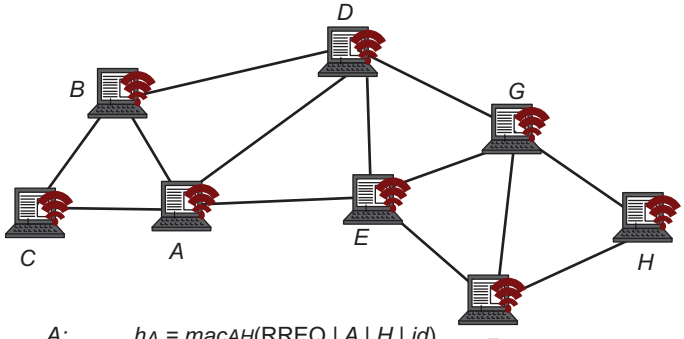
A continuació veurem el funcionament de dos dels més rellevants, l'Ariadne i l'endairA.

Ariadne

L'Ariadne utilitza l'autenticació dels missatges de control RREQ i RREP per evitar-ne la modificació i falsificació. L'autenticació dels missatges es pot fer de tres maneres diferents: a patir de signatures digitals, usant MAC, o a partir del protocol TESLA. A més, l'Ariadne utilitza un mecanisme basat en funcions resum per a evitar la manipulació de la informació per part dels nodes intermedis d'una comunicació durant el procés d'enviament dels RREQ.

L'operació d'Ariadne utilitzant signatures digitals s'il·lustra en la figura 4. En la figura té lloc un protocol de descobriment de rutes entre els nodes *A* i *H* (el mateix que s'il·lustra en la figura 2). La font *A* genera un RREQ i l'envia per *broadcast* a tota la xarxa. Aquest missatge d'RREQ, a més de contenir la informació bàsica del DSR, conté un codi d'autenticació MAC calculat amb una clau compartida entre la font *A* i la destinació *H*. El paquet RREQ va viatjant a través dels diferents nodes de la xarxa. Cada vegada que un node retransmet un paquet RREQ calcula de manera iterativa un resum *hash* sobre l'últim MAC o resum *hash* que ha rebut juntament amb el seu identificador de node. A més, el node que rep un RREQ afegeix el seu identificador a la llista de nodes itermedis i adjunta una signatura de tots els valors del missatge.

Figura 4. Protocol Ariadne amb operacions de firma digital



A: $h_A = mac_{AH}(\text{RREQ} \mid A \mid H \mid id)$
$A \rightarrow *$: [RREQ, A, H, id, h_A, (), ()]

E: $h_E = H(E \mid h_A)$
$E \rightarrow *$: [RREQ, A, H, id, h_E, (E), (sig_E)]

F: $h_F = H(F \mid h_E)$
$F \rightarrow *$: [RREQ, A, H, id, h_F, (E,F), (sig_E, sig_F)]

$H \rightarrow A$: [RREP, H, A, (E,F), (sig_E, sig_F), sig_H]

Quan la destinació rep l'RREQ verifica el valor de resum recalculant el MAC de la font i els valors de resum salt a salt. Després verifica totes les signatures digitals. Si totes aquestes verificacions són exitoses, la destinació genera un RREP i l'envia de tornada a la font a través de la ruta inversa obtinguda en el RREQ. El RREP conté els identificadors del node destinació i del node font, la ruta a seguir, la llista de signatures obtingudes de l'RREQ, i la signatura digital de la destinació sobre tots aquests elements. Cada node intermediari passa la resposta al node següent de la ruta sense fer-hi modificacions. Quan la font rep la resposta verifica la signatura digital dels nodes intermedis (per fer-ho necessita reconstruir les peticions que els nodes intermedis van firmar). Si les verificacions són correctes, la font rep la resposta com a vàlida.

Quan Ariadne fa servir MAC, s'assumeix que cada node intermediari comparteix una clau amb el node destinació. El funcionament és molt similar a l'Ariadne amb signatures digital, però en comptes de signatures, els nodes intermedis calculen MAC de l'RREQ amb la clau que comparteixen amb la destinació (en el cas de l'exemple, el node *H*). Quan l'RREQ arriba a la destinació aquest pot verificar tots els MAC i calcular un altre MAC de tots els camps amb la clau que comparteixen destinació i origen. Observa que, en aquest cas, la font no pot autenticar els nodes intermedis sinó que ha de confiar que la destinació ha executat la seva part del protocol correctament. D'altra banda, els nodes intermedis no poden autenticar ni la font ni la destinació.

Si l'Ariadne es fa servir amb TESLA, els usuaris van afegint MAC al paquet de manera similar a les operacions d'Ariadne amb MAC. L'avantatge és que les claus MAC permeten autenticar els paquets, i són molt més eficients que les signatures digitals. El punt negatiu és el retard que introdueix TESLA, que segons el tipus de serveis pot ser insostenible.

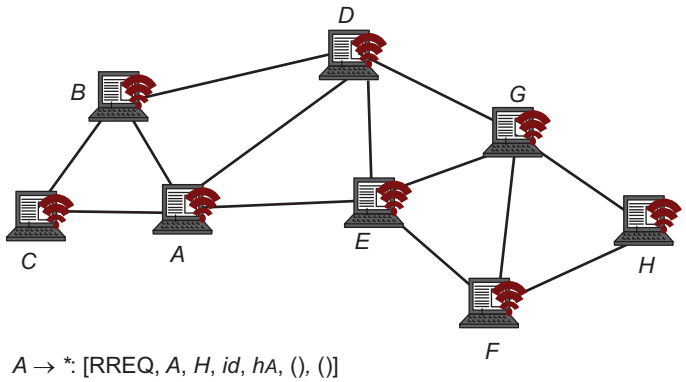
La figura 5 mostra un exemple del protocol Ariadne amb mode d'operació TESLA. Observeu que en el RREP, els nodes intermedis retarden la resposta fins que poden revelar la clau TESLA que han utilitzat per a generar el MAC. Aleshores inclouen la clau TESLA en la resposta. La font pot verificar la firma MAC de la destinació i de tots els nodes intermedis.

endairA

endairA és una altra extensió de seguretat per al protocol DSR que està inspirada en el protocol Ariadne. La diferència principal amb els seu predecessor és que en comptes de signar els missatges d'RREQ, en l'endairA els nodes intermedis signen l'RREP. Això explica el nom endairA, palíndrom d'Ariadne.

El funcionament d'endairA s'il·lustra en la figura 6. La font genera un RREQ que conté els identificadors de la font i de la destinació, i un identificador d'RREQ aleatori. Cada node intermediari que rep un RREQ per primer cop, afegeix el seu identificador a la ruta acumulada fins a aquell moment i retransmet el

Figura 5. Protocol Ariadne amb mode d'operació TESLA

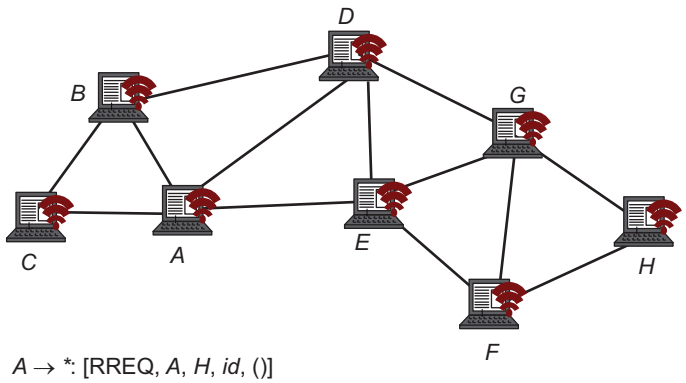


$A \rightarrow *$: [RREQ, A, H, id, h_A, (), ()]
$E \rightarrow *$: [RREQ, A, H, id, h_E, (E), ($mac_{KL,i}$)]
$F \rightarrow *$: [RREQ, A, H, id, h_F, (E,F), ($mac_{KE,i}$, $mac_{KF,i}$)]

$H \rightarrow F$: [RREP, H, A, (E,F), ($mac_{KE,i}$, $mac_{KF,i}$), mac_{HA}, ()]
$F \rightarrow E$: [RREP, H, A, (E,F), ($mac_{KF,i}$, $mac_{KF,i}$), mac_{HA}, ($K_{F,i}$)]
$E \rightarrow A$: [RREP, H, A, (E,F), ($mac_{KE,i}$, $mac_{KF,i}$), mac_{KHA}, ($K_{F,i}$,$K_{F,i}$)]

missatge en difusió (*broadcast*). Quan la petició arriba a la destinació es genera un RREP. L'RREP conté els identificadors de la font i de la destinació, la ruta acumulada des del node origen, i la firma digital de la destinació a tots aquests elements. La resposta és enviada de tornada al node origen a través del camí de nodes trobat en l'RREQ. Cada node intermediari que rep la resposta verifica que el seu identificador està a la llista, i que l'identificador del node anterior i posterior pertanyen a nodes veïns. També verifica que les signatures digitals de la resposta són vàlides i que corresponen als identificadors dels nodes de la llista i al node destinació. Si aquestes verificacions fallen, aleshores la resposta és descartada. En cas contrari, se signa i es passa el paquet al pròxim node de la ruta cap a la font. Quan la font rep l'RREP verifica que el missatge procedeix d'un veí, verifica les signatures i, si tot és correcte, accepta la ruta que s'indica en el missatge.

Figura 6. Protocol endairA



$A \rightarrow *$: [RREQ, A, H, id, ()]
$E \rightarrow *$: [RREQ, A, H, id, (E)]
$F \rightarrow *$: [RREQ, A, H, id, (E,F)]

$H \rightarrow F$: [RREP, H, A, (E,F), (sig_H)]
$F \rightarrow E$: [RREP, H, A, (E,F), (sig_H, sig_F)]
$E \rightarrow A$: [RREP, H, A, (E,F), (sig_H, sig_F, sig_E)]

Una de les principals contribucions d'endairA és la seva gestió eficient dels càlculs criptogràfics pesats. Tot el procés criptogràfic es fa en els missatges de resposta, i per tant, això vol dir que només es veuen involucrats en aquest procés els nodes que formen part de la ruta real entre la font i la destinació.

Càlcul criptogràfic en Ariadne

En el cas d'Ariadne el càlcul criptogràfic l'havien de fer tots els nodes de la xarxa, ja que el descobriment de rutes es fa fent una inundació de paquets a tota la xarxa.

Un problema bàsic d'endairA és que és vulnerable a atacs d'inundació per petició de rutes. Com que els missatges RREQ no estan autenticats, qualsevol adversari pot iniciar un procés de descobriment de rutes. Això es pot solucionar indicant que les RREQ han d'anar firmades pel node origen, amb la càrrega subseqüent per al sistema.

2.2.2. AODV

El protocol AODV* és un protocol reactiu basat en les tradicionals taules d'encaminament, no en l'encaminament de font com fa el DSR. Va ser definit el 2003 per la IETF. El 2005 l'IETF va començar a treballar en una nova proposta de protocol reactiu i basat en taules d'encaminament, el DYMO**.

*** De l'anglès, *Ad-Hoc On-Demand Distance Vector.***
**** De l'anglès, *Dynamic MANET On-demand Routing Protocol.***

DYMO és una clara evolució del protocol AODV, per això les últimes propostes d'aquest protocol han pres el nom d'AODVv2. Actualment AODVv2 està en fase d'esborrany en previsió de convertir-se en l'estàndard d'encaminament sota demanda de IETF.

Lectura complementària

El protocol AODV està definit en una RFC:
C. Perkins; E. Royer; S. Das (2003). "Ad hoc on-demand distance vector (AODV) routing". *RFC 3561* (*experimental*).

AODV opera de manera similar a DSR, amb un mecanisme per a fer el descobriment de rutes i un altre per a fer el manteniment. La diferència rau en el fet que els nodes utilitzen taules d'encaminament. Una entrada d'una taula d'encaminament conté la informació següent:

- identificador de la destinació,
- nombre de salts que es necessiten per arribar a la destinació,
- identificador del node del salt següent de la ruta cap a la destinació,
- número de seqüència de la destinació.

Número de seqüència de la destinació

Els números de seqüència de la destinació serveixen per a identificar i descartar informació no actual i assegurar que el protocol no entra en bucles d'encaminament.

En AODV, quan un node vol enviar informació a una destinació i no té una entrada vàlida per a aquest node en la seva taula d'encaminament, el que fa és generar una petició de ruta RREQ, que conté els identificadors de la font i la destinació, un comptador del nombre de salts (inicialment a 0), un número de seqüència associat a la font i un número de seqüència associat a la destinació. Cada node té un únic número de seqüència que s'incrementa després que es detecti un canvi en el conjunt de veïns del node. L'RREQ també conté un número que identifica la petició amb el mateix objectiu que en el protocol DSR, ajudar els nodes intermedis a detectar duplicats de la mateixa petició i descartar-los. L'RREQ s'envia en difusió a tots el veïns.

Quan un node intermedi rep un RREQ primer identifica si és un duplicat o no. Si és un duplicat el descarta. En cas contrari, el node comprova si té una entrada vàlida en la seva taula de rutes per a la destinació indicada en la petició. Si no la té, o té una entrada vàlida però amb un número de seqüència de destinació menor al que hi ha en la petició, incrementa el comptador de salts i reenvia el missatge als seus veïns. D'altra banda, si el node intermediari té una entrada vàlida, envia una resposta RREP. Si la petició arriba a la destinació, evidentment aquesta també genera una resposta RREP.

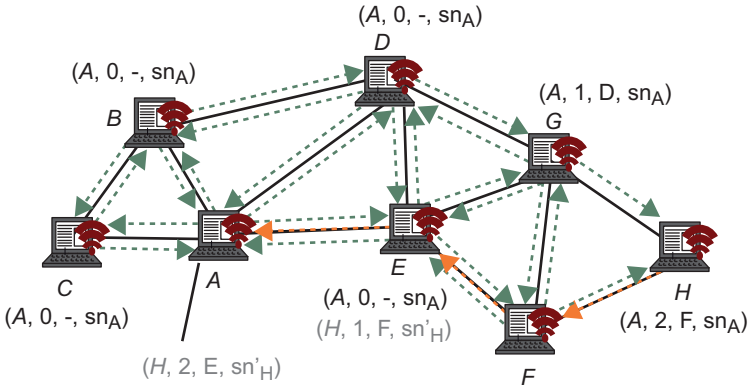
A més de les operacions descrites en el paràgraf anterior, quan un node rep un RREQ no duplicat també crea o actualitza l'entrada de la seva taula d'encaminament que correspon al node font de la petició. Si l'entrada de la taula ja està creada i el número de seqüència de la font és més gran que el que apareix a l'RREQ, o el número és igual però la longitud de la nova ruta a la font és més llarga que la que el node ja té (això és pot comprovar a partir del valor del nombre de salts que apareix en l'RREQ), la taula d'encaminament del node no s'actualitza. Si l'entrada de la taula sí que es crea o s'actualitza, la informació que s'hi ha de posar és la següent:

- l'identificador de l'entrada correspon a l'identificador de la font de l'RREQ;
- la longitud de la ruta que és establerta amb el valor del comptador de salts del RREQ;
- el salt següent per a arribar al node, que és l'identificador del node a través del qual s'ha rebut el missatge d'RREQ;
- i el número de seqüència de l'entrada, que és el número de seqüència de la font de la petició. Aquesta entrada de la taula d'encaminament només s'utilitzarà si aquest node rep finalment un missatge de resposta RREP per a ser reenviat a la font.

Com s'ha mencionat, l'RREP el pot construir el node destinació o un node intermedi que tingui informació actualitzada de la destinació en la seva taula de ruta. L'RREP conté informació sobre el número de seqüència de la destinació i la distància (en salts) a què es troba. El missatge d'RREP es reenvia cap al node origen a través del camí que s'ha establert a través de l'RREQ. El processament que fan els nodes intermedis del missatge RREP és molt similar al que fan per a l'RREQ; és a dir, incrementen el comptador de salts i creen/actualitzen l'entrada de la seva taula d'encaminament que correspon al node destinació.

Finalment, AODV també té mecanismes de manteniment de rutes que usen missatges d'error similars als de DSR si troben que algun enllaç de la xarxa està trencat.

Figura 7. Protocol AODV entre els nodes *A* i *F*



$A \rightarrow$ *:[RREQ, id, A, H, 0, sn_A, sn_H]
$B \rightarrow$ *:[RREQ, id, A, H, 1, sn_A, sn_H]
$C \rightarrow$ *:[RREQ, id, A, H, 1, sn_A, sn_H]
$D \rightarrow$ *:[RREQ, id, A, H, 1, sn_A, sn_H]
$E \rightarrow$ *:[RREQ, id, A, H, 1, sn_A, sn_H]
$F \rightarrow$ *:[RREQ, id, A, H, 2, sn_A, sn_H]
$G \rightarrow$ *:[RREQ, id, A, H, 2, sn_A, sn_H]

$H \rightarrow F$:[RREP, A, H, 0, sn'_H]
$F \rightarrow E$:[RREP, A, H, 1, sn'_H]
$E \rightarrow A$:[RREP, A, H, 2, sn'_H]

Vulnerabilitats de l'AODV

A continuació exposem algunes de les vulnerabilitats més grans de l'AODV. Com en el cas del DSR, utilitzarem la figura 3 per a il·lustrar alguns atacs contra el protocol d'encaminament.

1) Interrupció de rutes. Suposem que un node *S* vol trobar una ruta cap a la destinació *D*. Un adversari pot evitar el descobriment de la ruta entre aquests dos nodes manipulant el valor del nombre de salts en el missatge d'RREQ. Per exemple, si el node *A* defineix que el nombre de salts de la petició de ruta rebuda del node *H* és 0, el node *E* pensarà que la ruta més curta cap a *S* és a través de *A*, i per tant, *E* enviarà els missatges de resposta RREP destinats a *S* a través de *A*. Si *A* elimina els missatges d'RREP, la ruta entre *S* i *D* mai no es podrà descobrir.

2) Desviament de rutes. El desviament de rutes en AODV també es pot portar a terme modificant el valor del nombre de salts dels missatges RREQ i RREP. Per exemple, si el node *A* de la figura no elimina el missatge RREP en l'anterior atac d'interrupció de rutes, aleshores la ruta *S*, *H*, *A*, *E*, *G*, *D* és la que serà establerta com a oficial (quan en realitat n'hi ha una de més curta).

3) Creació d'estats d'encaminament incorrectes. Un node pot crear taules d'encaminament incorrectes si manipula el número de seqüència de la destinació o el comptador de salts en els missatges de control de l'encaminament. Per exemple, si *S* inicia un RREQ cap a *D* i *A* rep aquesta petició a través de *H*,

A pot incrementar el número de seqüència de la destinació abans de reenviar el missatge per difusió. Com a resultat, *E* fixa *F* com el seu salt següent cap a *S*, ja que *F* és el veí de *E*, i la ruta falsificada sembla més fresca que la correcta que venia del node *B*. Per tant, *E* obté un estat incorrecte ja que no hi ha cap ruta a *S* via *F*.

Hi ha diversos protocols que afegeixen seguretat a l'AODV, com ara els següents:

- SAODV (*secure AODV*)
- SEAODV (*security enhanced AODV*)
- ARAN (*authenticated routing for ad hoc networks*)
- SEDYMO (*secure DYMO*)

A continuació veurem el funcionament de SAODV, una de les propostes més eficients de les proposades. SAODV és una extensió del protocol bàsic AODV. Com hem comentat, a partir d'AODV han anat sorgint millores que s'han anomenat DYMO i AODV2. L'extensió que permetria donar seguretat a aquests protocols millorats seria la SEDYMO, que es basa en els mateixos principis de fucionament de SAODV, però que té en compte les funcionalitats esteses de DYMO.

SAODV

Podem considerar que els missatges d'encaminament d'AODV tenen dues parts, una mutable (la informació canvia a cada salt de l'RREQ/RREP), i una immutable. La part mutable dels missatges és la mètrica, és a dir, la informació sobre la distància (nombre de salts) a què es troba el node origen o la destinació. La part immutable inclou el número de seqüència dels nodes, les adreces de la font i la destinació, i l'identificador de la petició.

La seguretat en SAODV es basa en dos mecanismes: les cadenes resum per als camps mutables del missatge, i les signatures digitals per a protegir la part immutable i creada per la font de l'RREQ/RREP.

El mecanisme de cadenes resum en SAODV fa que s'hagin d'afegir tres camps específics per a aquest ús en els missatges d'encaminament, de manera que la informació final sobre la mètrica és la següent:

- `HopCount`: comptador del nombre de salts. Inicialment està a 0.

- `MaxHopCount`: valor que determina el nombre màxim de salts que un paquet pot fer a la xarxa. Generalment s'inicialitza amb el valor del TTL *(Time-to-Live)*.

- `Hash`: valor de la cadena resum. Aquest camp s'inicialitza amb un valor aleatori anomenat llavors *s*.

- `TopHash`: valor resultant d'aplicar `MaxHopCount` vegades la funció resum a la llavor *s*.

Els camps `MaxHopCount` i `TopHash` pertanyen a la part immutable del missatge (i estan protegides per la signatura digital), mentre que els camps `HopCount` i `Hash` canvien a cada salt. Cada node que rep un missatge d'encaminament calcula la diferència *d* entre els valors dels camps `MaxHopCount` i `HopCount`, i seguidament calcula *d* vegades una funció resum sobre el camp `Hash`. El missatge és validat satisfactòriament si el resultat de l'operació coincideix amb el valor del camp `TopHash`. Aleshores, el node incrementa en una unitat el valor del `HopCount`, actualitza el valor del `Hash` calculant el resum (*hash*) de l'anterior valor que hi havia en aquest camp, i reenvia el paquet en difusió als seus veïns.

El propòsit d'aquest mecanisme és que tothom pugui verificar el valor del `HopCount` i que es puguin evitar atacs derivats de la falsificació d'aquest valor. Tanmateix, aquest sistema només permet evitar que un adversari decrementi desmesuradament el valor del nombre de salts d'un paquet (fent que les rutes semblin més curtes del que en realitat són) però no que el mantingui igual o l'augmenti. Si un adversari no incrementa el valor del `HopCount` en el fons també està fent que aquesta ruta sigui una unitat menor del que hauria de ser, amb les conseqüències que això pot tenir per a l'encaminament. D'altra banda, si els usuaris augmenten el valor del `HopCount` poden crear atacs d'egoisme (*selfishness*), evitant que les rutes de la xarxa ad hoc passin per ells per no patir el cost que comporta participar en una xarxa col·laborativa.

Com hem comentat, l'altre mecanisme de seguretat del SAODV és la signatura digital. El node emissor de l'RREQ genera una signatura utilitzant la seva clau privada que protegeix la integritat i l'autenticitat dels paràmetres estàtics del missatge, i d'aquesta manera preveu atacs de modificació dels paquets.

2.2.3. OLSR

El protocol OLSR* és un protocol proactiu que es diferencia de DSR i AODV en tant que manté la informació sobre les rutes actives constantment actualitzada. Això evita que hi hagi retards a l'inici de les transmissions quan un node vol establir una comunicació amb una destinació remota. A canvi, requereix un flux constant de petits missatges de control que mantinguin actualitzada la informació sobre l'estat i la localització dels nodes de la xarxa.

L'OLSR periòdicament inunda la xarxa amb missatges de control. Tanmateix, aquesta inundació es fa a través de nodes seleccionats (els anomenats *multipoint relays*, MPR) per tal d'optimitzar al màxim la sobrecàrrega del sistema

*** De l'anglès, *Optimized Link State Routing.***

Lectura complementària

El protocol OLSR està definit en una RFC:
T. Clausen; P. Jacquet (2003). "Optimized link state routing protocol (OLSR)". *RFC 3626* (*experimental*) (octubre).

i reduir el nombre de retransmissions d'un missatge de control. El conjunt d'MPR d'un node donat és el subconjunt dels seus veïns que han estat seleccionats de manera que puguin cobrir (en termes de cobertura ràdio) tots els veïns que estan estrictament a dos salts de distància del node.

En OLSR hi ha bàsicament dos tipus de missatges:

1) Missatges *Hello*: com hem vist es tracta de missatges que s'envien en difusió local per tal de descobrir veïns.

2) Missatges *TC* (control de topologia): es tracta de missatges que inunden la xarxa a través dels MPR.

Els missatges *Hello* enviats per un cert node A contenen una llista dels seus suposats veïns. Per a cada veí de la llista es detalla l'estat de l'enllaç entre el node A i aquest veí i si es tracta d'un node MPR o no.

Quan un node *B* rep un missatge de *Hello*, obté diferent informació: per un cantó entén que el remitent del missatge és un node veí seu. Si *B* està llistat com a veí en el missatge de *Hello*, aleshores *B* aprèn que *A* el considera veí, la qual cosa vol dir que entre ells hi ha un enllaç bidireccional. Sinó, s'assumeix que l'enllaç és asimètric. D'altra banda, si *B* està marcat com un MPR, aleshores sap que *A* l'ha seleccionat com a tal i per tant, *A* es troba en el conjunt de selectors de *B*. Finalment, mirant la llista de veïns de *A*, *B* coneix els nodes que té a dos salts de distància.

Així doncs, els missatges de *Hello* en OLSR serveixen per a obtenir informació sobre l'estat dels enllaços, per a detectar veïns a un salt i a dos salts, i per a assignar els MPR. Indirectament, els missatges de *Hello* també serveixen per a fer la selecció d'MPR, ja que els nodes es basen en la informació de veïnatge per a determinar quins seran escollits amb aquest rol.

Els missatges de *TC* només els poden enviar i retransmetre els nodes MPR. Contenen una llista d'enllaços actius (com a mínim entre el node remitent de l'MPR i els seus selectors) i serveixen per a proporcionar suficient informació per tal que els nodes puguin construir la seva pròpia taula topològica, i aleshores deduir-ne la taula de rutes. Les rutes es calculen optimitzant el nombre de salts (per exemple, amb l'algorisme del camí més curt de Dijkstra).

La informació emmagatzemada en una taula d'encaminament es basa en el veïnatge i la topologia, per tant s'ha de recalcular si alguna d'aquesta informació canvia. En particular, s'ha d'actualitzar en els casos següents:

- Si es detecta un canvi en el veïnatge del node.
- Si expira una ruta i se n'ha d'actualitzar la informació.
- Si es detecta un camí millor (més curt) al mateix node de destinació.

Vulnerabilitats de l'OLSR

L'OLSR és vulnerable a diversos atacs. A continuació n'exposem alguns:

1) Atac de suplantació d'identitats (*spoofing*). Els atacants poden usar l'identificador d'un altre node i mostrar-se a si mateixos com si fossin un altre. Així, el node *A* podria enviar un missatge al seu veí *C* fent-se passar per *B*. Aleshores *C* consideraria que ell i *B* són veïns quan en realitat no ho són.

2) Disrupció de rutes. Un atacant pot eliminar els missatges d'encaminament que rep en comptes de retransmetre'ls a la resta de veïns tal com marca el protocol. D'aquesta manera és redueix la quantitat d'informació d'encaminament que els altres nodes tenen disponible.

3) Atac per replicació. En aquest atac es reenvia a la xarxa un missatge de control antic i tots els nodes actualitzen les taules amb informació errònia.

Hi ha diversos protocols que afegeixen seguretat a l'OLSR, com són els següents:

- *secure OLSR*
- *security aware OLSR* (SL-OLSR)
- *shared secret-based OLSR*

L'objectiu dels protocols segurs sobre OLSR és donar serveis com ara autenticació dels nodes en els missatges de *Hello*, proporcionar integritat als missatges de control i detectar nodes egoistes.

La majoria de solucions es basen en el següent:

1) Introduir un segell de temps en els missatges per tal d'evitar els atacs de replicació.

2) Usar mecanismes d'autenticació en els missatges de *Hello* i *TC*. En la mesura que sigui possible, s'han d'intentar evitar els mecanismes de clau pública per l'elevat cost computacional que tenen. El problema rau a com fer una gestió eficient de claus per a poder utilitzar algorismes de clau simètrica. Una solució és utilitzar el protocol TESLA.

Vegeu també

El protocol TESLA s'estudia al subapartat 2.1.1. d'aquest mòdul.

3) Correlacionar les dades dels diferents missatges de *Hello* per evitar incoherències. Per a evitar atacs de nodes interns a la xarxa (i autenticats) s'han d'intentar validar les dades que aporten el diferents nodes de la xarxa.

2.3. Privadesa

La propietat de la privadesa significa que un subjecte pot controlar quan, on i com és usada la informació sobre si mateix i per qui. La privadesa és un dels grans problemes de les xarxes ad hoc, ja que per tal que dos nodes remots puguin comunicar-se han de declarar que estan actius a la xarxa i han de participar en els protocols d'encaminament amb la conseqüent publicació de quina és la seva posició/localització a la xarxa.

Els atacs a la privadesa poden ser tant actius com passius. En els atacs actius els usuaris maliciosos participen en el protocol de xarxa el qual pretenen trencar, ja sigui a través d'atacs externs (com a usuaris aliens al sistema) o atacs interns (com a membres lícits de la xarxa). D'altra banda, els atacs passius no pertorben el funcionament normal dels protocols de xarxa, els atacants escolten de manera no autoritzada els paquets que es transmeten per la xarxa i a través d'una anàlisi de trànsit extrapolen informació com les rutes de transmissió, el contingut dels missatges o la identitat, posició o moviment dels nodes.

La privadesa és un concepte molt ampli, però en el cas de les comunicacions en xarxa, el que ens interessa és focalitzar-nos principalment en tres de les seves propietats:

1) Anonimat dels subjectes: propietat de no ser identificable entre un conjunt de subjectes.

2) Desvinculació de missatges: propietat d'ocultar la relació que hi ha entre una comunicació i les persones que la duen a terme.

3) Indetectabilitat: incapacitat de distingir si un element existeix o no. En el cas dels missatges, la indetectabilitat significa que aquests no són prou discernibles de, per exemple, soroll blanc.

A continuació veurem alguns mecanismes que permeten evitar les vulnerabilitats de privadesa en les xarxes ad hoc.

2.3.1. Anonimat dels subjectes

En aquest subapartat veurem dos mecanismes bàsics que permeten proporcionar anonimat als subjectes d'una comunicació: els pseudònims i les funcions unidireccionals amb trampa.

Pseudònims

Una manera d'amagar la identitat dels subjectes que actuen en una comunicació és a través de pseudònims, és a dir, l'ús d'una etiqueta privada que

permet, de manera discrecional, distingir els participants d'una transacció. A partir de la informació pública de la xarxa, els nodes són incapaços de generar i/o vincular pseudònims per a la resta de membres de la xarxa.

Lectura recomanada

Els esquemes basats en pseudònims van ser introduïts el 1985 per Chaum:
David Chaum (1985). "Security without identification: transaction systems to make big brother obsolete". *Communication ACM* (núm. 28, vol. 10, pàg. 1030-1044).

Les dificultats de posar en funcionament un sistema de pseudònims són les següents:

- **Temporalitat**. Els pseudònims s'han de renovar periòdicament perquè el seu ús revela certa informació que es podria utilitzar per a identificar o localitzar un subjecte.

- **Generació i gestió dels pseudònims**. El vincle entre un pseudònim i la identitat real del subjecte o enllaç al qual està associat és privada. No obstant això, s'han de proporcionar mecanismes per a fer arribar aquesta informació als usuaris autoritzats, de manera que la comunicació entre entitats sigui viable.

- **Autenticació**. L'autenticitat dels participants en una transacció hauria de poder ser garantida encara que es fessin servir pseudònims.

Els sistemes de pseudònims més comuns són els que utilitzen una tercera entitat de confiança (TTP) responsable de generar, renovar, revocar i autenticar els pseudònims.

Tot i que la TTP renovi els pseudònims periòdicament, aquest sol fet pot no protegir suficientment la xarxa davant d'un potencial usuari maliciós amb capacitat d'escoltar totes les comunicacions de la xarxa. La raó és que un atacant amb capacitats globals d'escolta podria enllaçar amb una alta probabilitat els diferents pseudònims d'un sol node basant-se en la posició i velocitat de la informació que s'origina en un cert espai de la xarxa. És a dir, un cop l'atacant coneix la localització d'un node, el seu perfil de moviment i amb quina freqüència sol enviar missatges, és molt fàcil seguir-li la pista encara que canviï de pseudònim.

Una manera de minimitzar l'efecte dels atacants amb gran poder d'escolta és la inclusió de zones de mescla (*mix zone*) a la xarxa. Això significa que els nodes canvien de pseudònim quan es troben en unes àrees predefinides, petites i fitades. Aquest model el que busca és la convergència en temps i en espai de diferents nodes canviant de pseudònim. En haver-hi diferents usuaris fent la mateixa operació alhora, un adversari ja no pot mapejar directament quin és el nou pseudònim de cada node i, per tant, s'incrementa la privacitat.

Funcions unidireccionals amb trampa

La funcions unidireccionals amb trampa* són funcions unidireccionals $f : X \to Y$ tals que és fàcil obtenir $f(x)$ per a qualsevol $x \in X$, i que permeten el càlcul eficient de la inversa (trobar $x \in X$ tal que $f(x) = y$) si i solament si es té

* En anglès, *trapdoor functions.*

certa informació addicional, la trampa. En cas contrari, el càlcul de l'invers és computacionalment intractable.

Les funcions unidireccionals amb trampa s'utilitzen per a la identificació anònima dels receptors d'una comunicació. L'emissor envia la informació d'identificació de la comunicació amagada en una funció unidireccional, de manera que només el receptor legítim de la transmissió, que posseeix la informació trampa, sigui capaç de recuperar-la.

La manera més simple d'implementar una funció unidireccional per a proporcionar anonimat de recepció és a través de criptografia de clau pública. La identitat del receptor s'envia xifrada amb la clau pública del mateix receptor, de manera que només ell pugui obrir amb èxit el paquet. No obstant això, aquesta solució és molt costosa ja que el descobriment de rutes en xarxes ad hoc es fa a través de mecanismes d'inundació *broadcast*, i si tots els nodes que reben un paquet han de fer una operació criptogràfica per descobrir si són els receptors d'un paquet, la càrrega total del sistema és insostenible.

Protocols d'encaminament com SDAR i AnonDSR utilitzen criptografia de clau pública per a protegir la identitat dels dos nodes d'una comunicació.

Una altra alternativa consisteix a usar criptografia de clau simètrica. En aquest cas s'assumeix que origen i destinació comparteixen una clau que s'han distribuït a través d'un canal segur. L'origen xifra la identitat de la destinació i un nombre aleatori amb la clau simètrica que comparteixen. El node que pugui obrir aquest sobre i comprovar que hi és el seu identificador, és el receptorlegítim. Finalment, en el missatge de resposta a l'origen, la destinació envia el número aleatori del sobre com a prova de recepció d'aquest sobre.

Els protocols ANODR i ASR utilitzen criptografia simètrica per a amagar la identitat de la destinació d'una comunicació.

Finalment, una altra solució és utilitzar funcions amb trampa més lleugeres, basades en funcions resum. Si l'origen i la destinació comparteixen un secret, la destinació es pot identificar a través d'un valor HMAC d'un cert valor aleatori (que pot ser públic).

2.3.2. Desvinculació de missatges

Les principals tècniques que s'utilitzen per a evitar que un adversari pugui inferir els subjectes que participen en una comunicació es basen en un model de mesclador (*mix router*). Un mesclador és un encaminador que amaga la correspondència entre missatges entrants i sortints a partir de la modificació de la seva aparença i del flux de la transmissió. A continuació veurem com s'utilitzen els mescladors.

Mesclador

Un mesclador és un encaminador que rep un conjunt de missatges d'entrada i els retorna transformats de manera que no es pugui relacionar l'entrada amb la sortida.

Lectura recomanada

El disseny original del mesclador va ser proposat per Chaum en l'article següent:
David L. Chaum (1981). "Untraceable electronic mail, return addresses, and digital pseudonyms". *Communication ACM* (núm. 24, vol. 2, pàg. 84-90).

Aquestes transformacions es poden produir tant en la forma (a força d'aplicar tècniques d'encriptació i farcit de missatges) com en la seqüència (a força de barrejar l'ordre i aplicar diferents retards en el lliurament dels missatges).

El disseny original del mesclador es va concebre per a una xarxa tradicional i consisteix a tenir un encaminador que processa els missatges per lots. L'encaminador emmagatzema missatges a la memòria fins que es compleix una certa condició de descàrrega, moment en què s'envia el lot de missatges desordenats. La condició de descàrrega pot ser una condició temporal, espacial o una combinació d'ambdues. La descàrrega temporal s'estableix cada cert període de temps (que pot ser fix o variable), mentre que l'espacial s'estableix quan s'arriba a sobrepassar un determinat llindar de capacitat.

Mesclador amb piscina

El disseny original del mesclador per lots va ser estès més endavant, de manera que en el moment de la descàrrega només s'enviessin un subconjunt dels missatges emmagatzemats en l'encaminador i la resta es preservessin per a rondes posteriors. Aquesta tècnica, anomenada mesclador amb piscina (*Pool Mix*), millora el grau d'anonimat en situacions de trànsit fluctuant compensant un moment de poca càrrega de trànsit amb un major retard en el lliurament dels missatges. Aquesta solució és ideal per a aplicacions que no tenen restriccions de lliurament molt ajustades, com ara el correu electrònic anònim, però no és adequat per a xarxes que necessiten comunicacions en temps real.

En contraposició al model de mesclador per lots tenim el mesclador continu, en el qual els usuaris defineixen un retard aleatori per a cada missatge i inclouen aquest retard en la capçalera del missatge. El mesclador emmagatzema el missatge durant el temps especificat i llavors el reenvia. L'avantatge d'aquest mètode és que els mateixos usuaris controlen el temps límit de transferència de la informació. Aquest model funciona bé en situacions de trànsit relativament estable i constant. No obstant això, en cas que es produeixin períodes de trànsit reduït, el grau d'anonimat d'aquest model és baix.

Tant el mesclador continu com el de piscina són vulnerables a atacs consistents en l'alteració del flux de $N - 1$ missatges (N és el nombre de missatges llindar necessaris perquè es produeixi la descàrrega a l'encaminador) amb l'objectiu de poder traçar un missatge concret. Per al cas del mesclador continu l'atacant ha de ser capaç de bloquejar l'entrada de missatges a l'encaminador, mentre que per al mesclador de piscina hauria d'injectar missatges marcats que provoquessin una descàrrega controlada del mesclador.

Per a mitigar l'efecte d'aquest atac, una solució és utilitzar les xarxes de mescladors.

Xarxa de mescladors

Per a incrementar el grau d'anonimat d'un sistema mesclador, els encaminadors mescladors solen combinar-se formant una xarxa de mescladors. D'aquesta manera, es pot arribar a preservar l'anonimat dels usuaris de la xarxa encara que alguns nodes mescladors resultin compromesos.

Hi ha dues topologies bàsiques de xarxes de mescladors:

- **Cascada**: la ruta o rutes que segueixen els missatges estan preestablertes.
- **Ruta lliure**: cada missatge pot seguir una ruta independent i diferent dels altres missatges.

Un avantatge de les cascades sobre els mescladors de ruta lliure és el fet que tendeixen a concentrar més trànsit per les seves rutes, cosa que augmenta el grau d'anonimat en les rutes. No obstant això, en una cascada un adversari podria arribar a conèixer exactament quins mescladors ha de controlar per a traçar a un usuari en particular. Són sistemes vulnerables a atacs $N-1$ efectuats per un adversari global. D'altra banda, l'establiment d'una única ruta debilita la seguretat del sistema resultant en fer-lo vulnerable a atacs del tipus *rushing* –en els quals l'adversari tracta d'enviar missatges de descobriment de la ruta abans que el node font per intentar "apropiar-se" de la ruta–, i intrusions d'un adversari sobre un dels nodes de la ruta.

Els models de mescladors combinats tracten d'obtenir els avantatges de les dues opcions. Un exemple és l'establiment de múltiples cascades lliures. Aquest cas es basa en el model en cascada, però les rutes s'estenen més enllà de la seva destinació o bé introdueixen rutes falses. D'altra banda, els protocols d'encaminament en MANET comencen a introduir l'establiment de canals de comunicació entre dos punts a través de múltiples rutes, la qual cosa els fa més robustos.

2.3.3. Indetectabilitat

Típicament les xarxes anònimes perden robustesa al llarg del temps pel fet que una anàlisi exhaustiva de les traces de la xarxa permet obtenir informació dels usuaris i les relacions que hi ha entre ells. Una forma d'atacar l'arrel d'aquest problema és emmascarar els missatges entre nodes, de manera que un atacant extern no pugui distingir quan la xarxa està enviant dades o soroll.

Entre les tècniques més utilitzades per a emmascarar els missatges en destaquen les següents:

- **Comunicacions a ràfegues curtes (*burst communications*)**. La transmissió de missatges molt curts és molt difícil de detectar pels usuaris externs a la xarxa. És per això que aquest tipus de transmissions s'utilitzen per a enviar la informació de control més sensible de la xarxa.

- **Enviament de missatges ficticis (*dummy data*)**. El seu objectiu és aconseguir un flux constant a la xarxa i que el tipus de trànsit (real o fals) sigui indestriable a ulls d'un atacant.

- **Modulació per espectre eixamplat (*spread spectrum*)**. Les transmissions per espectre eixamplat es caracteritzen perquè la informació és enviada a través d'una amplada de banda molt més àmplia que el mínim requerit. Les tècniques més usades són els sistemes de seqüència directa i els sistemes de salt de freqüència. L'avantatge d'aquests sistemes és que el senyal és molt difícil de detectar per a usuaris que desconeguin la tècnica i la codificació usada per a la transmissió del senyal.

- **Esteganografia**. Els mètodes esteganogràfics permeten amagar un missatge dins d'un flux de comunicació qualsevol de la xarxa, de manera que només el receptor legítim pugui extreure la informació del canal. Per a la resta d'usuaris el missatge és invisible.

De les tècniques per a emmascarar missatges, les més senzilla i usada és la d'enviament de missatges ficticis. Els missatges ficticis poden ser inserits en l'entrada o sortida dels mescladors. Normalment la inserció en la sortida proveeix major anonimat i menor retard pel fet que el mesclador pot regular de manera més precisa la introducció de missatges ficticis a la xarxa segons l'estat del trànsit. No obstant això, en el cas de l'atac $N-1$ la inserció a l'entrada del mesclador pot oferir un major nivell de protecció.

Quan tractem xarxes de mescladors, els missatges falsos poden travessar diversos encaminadors, tal com fan la resta de missatges. El camí per travessar es determina de manera aleatòria i normalment acaba en l'encaminador que el va generar. Això permet arribar a detectar atacs del tipus $N-1$ i actuar en conseqüència.

A l'entorn de xarxes ad hoc, en què l'ús de recursos és molt limitat, l'ús d'aquest tipus de missatges ha de ser realment minimitzat.

3. Xarxes de sensors

Una xarxa de sensors sense fils pot estar formada per una sèrie de milers, fins i tot milions de sensors (nodes), els quals posseeixen capacitat d'emmagatzematge, processament i energia limitada. Aquestes xarxes se solen desplegar en entorns hostils o de difícil accés per a poder obtenir regularment dades del context (militar, ambiental, biològic, mèdic, etc.) i controlar-lo. En aquests sistemes els nodes estan exposats a atacs físics i de programari, i per tant la prevenció i el control de la seguretat dels sistemes és un element essencial d'estudi en les xarxes de sensors.

Els nodes sensors es poden assumir com a petits computadors, extremadament rudimentaris en termes de característiques tècniques (nombre i tipus d'interfícies i components). Usualment consisteixen en una unitat de procés (microprocessador) amb capacitat computacional limitada i no gaire memòria (el que seria el sensor pròpiament dit), algun dispositiu de comunicacions (usualment ràdio) i alguna font d'energia (usualment una bateria). Algunes implementacions poden incloure elements addicionals, com ara sistemes de recàrrega de bateria, processadors secundaris i dispositius de comunicacions addicionals (RS-232, USB, etc.). La mida del node pot variar des de l'equivalent a una caixa de sabates fins a dispositius diminuts com grans d'arròs. Les restriccions de mida i cost dels nodes sensors afecten en gran mesura les restriccions quant a energia, memòria, capacitat computacional i amplada de banda que després podran utilitzar les aplicacions que en requereixin l'ús.

En aquest apartat estudiarem els principals problemes de seguretat relacionats amb les xarxes de sensors. En primer lloc veurem els esquemes de gestió de claus encarregats de la distribució i actualització del material criptogràfic a la xarxa per tal de dotar de seguretat les seves comunicacions. En segon lloc veurem com es pot optimitzar la transferència de dades des dels sensors cap a una estació base de manera que es pugui garantir la integritat i autenticitat de les dades.

3.1. Gestió de claus

L'establiment de claus de seguretat és el servei bàsic per a poder oferir mecanismes de prevenció i detecció d'atacs en xarxes. Les xarxes sense fils són particularment vulnerables a escoltes i al fet que els nodes poden ser capturats i compromesos.

Les solucions basades en clau pública són complicades de desplegar per dos motius:

1) els sensors posseeixen capacitats de processament, emmagatzematge i fonts d'energia limitades, l'ús de protocols basats en clau pública com Diffie-Hellman és massa costós,

2) la xarxa no té una infraestructura estable i, per tant, la verificació de revocació dels certificats de clau pública a través d'una autoritat de confiança no es pot portar a terme adequadament.

Així doncs, es considera que la criptografia de clau asimètrica s'ha d'evitar en les xarxes de sensors. Les solucions basades en claus simètriques són computacionalment més eficients, però més vulnerables a atacs. Els sistemes han d'assegurar que si un intrús aconsegueix capturar un node, no li sigui possible accedir a totes les claus de la xarxa i, per tant, a la informació confidencial del sistema.

L'objectiu és la creació i gestió de claus simètriques que puguin fer front a totes les necessitats de la xarxa. Els requisits per a l'establiment de claus depenen dels patrons de comunicació de la xarxa, que són:

1) Unidestinació, és a dir, enviar un missatge a un sol node;

2) Difusió local, és a dir, enviar un missatge als nodes veïns, amb què hi ha connexió directa;

3) Difusió global, és a dir, enviar un missatges a tots els nodes de la xarxa.

Els missatges unidestinació (*unicast*) s'utilitzen per a enviar informació de les dades capturades per un sensor a una estació base, o un altre sensor que farà funcions d'agregació de dades. L'agregació de dades permet reduir la quantitat de bits transmesos en una xarxa i incrementa l'eficiència i el temps de vida de la xarxa.

Els missatges de difusió local* s'utilitzen per al control i gestió de la xarxa, sobretot per a les operacions d'encaminament. Els missatges de difusió global** s'originen a l'estació base i s'utilitzen per a distribuir informació de control que involucra a tota la xarxa.

*** En anglès, *local broadcast*.**
**** En anglès, *global broadcast*.**

A partir d'aquests requisits de comunicació es defineixen 4 tipus de claus:

- **Clau de node**: clau que comparteix un node i l'estació base.
- **Clau d'enllaç**: clau que comparteixen una parella de nodes veïns.
- **Clau de grup**: clau compartida entre un node i tots els seus veïns.
- **Clau de xarxa**: clau compartida per tots els nodes de la xarxa i l'estació base.

Les claus dels nodes poden ser carregades als sensors en el procés de desplegament d'una xarxa. La clau de grup pot ser generada per un node i enviada de

manera individual a cada un dels seus veïns protegida amb la clau d'enllaç que tots dos comparteixen. Finalment, la clau de xarxa també pot ser carregada en els nodes abans del desplegament d'aquesta. Si es detecta que un node ha estat compromès, els seus veïns han de generar claus de grup noves i distribuir-les als veïns honrats. Aleshores, l'estació base ha de generar una clau de xarxa nova i distribuir-la salt a salt a tota la xarxa protegida amb les claus de grup.

Observeu que la clau més difícil de generar i gestionar és la clau d'enllaç entre dos nodes veïns.

Per a generar la clau d'enllaç es podria pensar a usar protocols de criptografia simètrica com Kerberos. En aquest cas l'estació base faria el paper de servidor. Per tal d'executar el protocol es necessita poder enviar missatges de l'estació base a la resta de nodes de la xarxa. El problema és que en una xarxa de sensors l'encaminament pot dependre d'altres nodes de la xarxa, i per tant pot necessitar que ja hi hagi establertes les claus d'enllaç. A més, la càrrega de comunicació associada a aquest protocol seria molt desigual entre els diferents sensors de la xarxa. Per últim, l'esquema seria poc robust per la presència de l'estació base com a punt únic de fallada.

Una altra aproximació seria precarregar les claus d'enllaç al sensor abans del desplegament de la xarxa, però això té diversos problemes. En primer lloc, en moltes aplicacions és difícil conèixer *a priori* quin serà el mapa exacte de la xarxa, ja que si, per exemple, els sensors es posicionen al seu lloc quan són llançats per una avioneta no hi ha forma de saber amb antelació quins seran els veïns de cada node. A més, en una xarxa es poden afegir sensors *a posteriori* per tal de reemplaçar sensors defectuosos o que han acabat el seu cicle de vida. És difícil anticipar en temps de desplegament de la xarxa on s'afegiran aquests nodes i, per tant, quins sensors necessiten estar precarregats amb claus criptogràfiques addicionals per a poder interactuar amb els nous sensors quan aquests arribin.

En els subapartats següents presentem diferents aproximacions a com es pot solucionar el problema de l'establiment d'una clau d'enllaç entre dos nodes veïns. El primer mètode es basa en una clau mestra de curt termini que és present en cada node durant un temps limitant després de cada desplegament. Els altres mètodes es basen a precarregar claus en els sensors d'una manera intel·ligent, sense haver de conèixer la topologia de la xarxa final i suportant l'addició posterior de nous nodes a la xarxa.

3.1.1. Establiment de claus a partir d'una clau mestra de curt termini

L'establiment de claus d'enllaç pot aprofitar el fet que les xarxes de sensor són xarxes relativament estàtiques i formades per nodes estacionaris. Això vol dir que el veïnatge d'un node no canvia sovint; alguns nodes poden morir,

altres poden ser afegits al cap d'un temps, però tot i així els canvis són molt esporàdics. Per tant, el que es pot fer és usar un protocol de descobriment de nodes veïns i establir les claus d'enllaç en el moment del desplegament inicial.

El protocol consisteix en quatre passos: càrrega de la clau mestra, descobriment de veïns, càlcul de la clau d'enllaç i supressió de la clau mestra. Vegem cada una d'aquestes fases.

La càrrega de la clau mestra es fa abans del desplegament de la xarxa, en un entorn segur. Durant aquesta fase, la clau mestra K_{init} es carrega en els nodes, i cada node u calcula una clau mestra del node anomenada K_u, tal que $K_u = f_{K_{init}}(u)$, on f és una funció pseudoaleatòria.

La fase de descobriment comença just després del desplegament d'un node. En primer lloc, el node inicialitza un temporitzador amb un temps T_{min}. Després intenta descobrir els seus veïns enviant un missatge *Hello* en difusió que conté el seu identificador. Un node veí v que senti el missatge de *Hello* respon amb un missatge *Ack* en què envia el seu identificador v. El missatge *Ack* també conté un codi d'autenticació MAC generat amb la clau mestra del node v, K_v. Com que el node u encara posseeix la clau mestra K_{init}, pot calcular K_u i verificar el MAC.

Un cop finalitzada la fase de descobriment de nodes, el node u entra en la fase de càlcul de la clau d'enllaç. La clau d'enllaç K_{uv} entre els nodes u i v es calcula com a $K_{uv} = f_{K_v}(u)$. Observem que els dos nodes poden calcular aquesta clau sense necessitat d'intercanviar cap més missatge. El node u no ha estat autenticat explícitament pel node v. Tanmateix, tots els missatges que u enviï a v seran autenticats amb la clau K_{uv} i, per tant, es pot assegurar que el node serà sempre el mateix.

Finalment, quan el temps del temporitzador T_{min} finalitza, el node u executa la fase de supressió de la clau mestra esborrant K_{init} i totes les claus mestres de nodes K_v de la seva memòria. No obstant això, no esborra la seva pròpia clau mestra K_u, i aquesta s'utilitzarà més endavant per a establir claus d'enllaç amb nodes que s'afegeixin posteriorment a la xarxa.

3.1.2. Piscines de claus

Lectura recomanada

El primer treball que fa servir el concepte de piscina de claus és:
Laurent Eschenauer; Virgil D. Gligor (2002). "A key-management scheme for distributed sensor networks". A: *Proceedings of the 9th ACM conference on Computer and communications security, CCS '02* (pàg. 41-47). ACM: Nova York.

Aquest tipus d'esquemes es basa en una predistribució aleatòria de claus el més òptima i eficient possible. Inicialment es genera un ampli conjunt de claus, el que anomenem piscina de claus. Abans de desplegar la xarxa s'entrega a cada node un subcojunt aleatori de la piscina de claus, un anell de claus. El fet que els nodes només hagin d'emmagatzemar un anell permet reduir els requisits de memòria i fa que el sistema sigui escalable i apropiat per a sensors. Tanmateix, és evident que inicialment no totes les parelles de veïns de la xarxa comparteixen una clau. Veurem que els protocols basats en piscina perme-

ten que qualsevol dels dos nodes que inicialment no comparteixen una clau puguin establir-ne una amb una alta probabilitat, a través de la comunicació amb nodes intermedis.

Per tal de calcular la probabilitat que dos nodes de la xarxa comparteixin una clau, expressem el problema de la manera següent: donat un conjunt S de k elements, de manera aleatòria escollim dos subconjunts S_1 i S_2 de m_1 i m_2 elements cada un. La probabilitat de $S_1 \cap S_2 \neq 0$ és:

$$P\{S_1 \cap S_2 \neq 0\} = 1 - \frac{(k-m_1)!(k-m_2)!}{k!(k-m_1-m_2)!} \quad (1)$$

Quan k és molt gran, podem utilitzar l'aproximació d'Stirling per a $n!$,

$$n! \approx \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n}$$

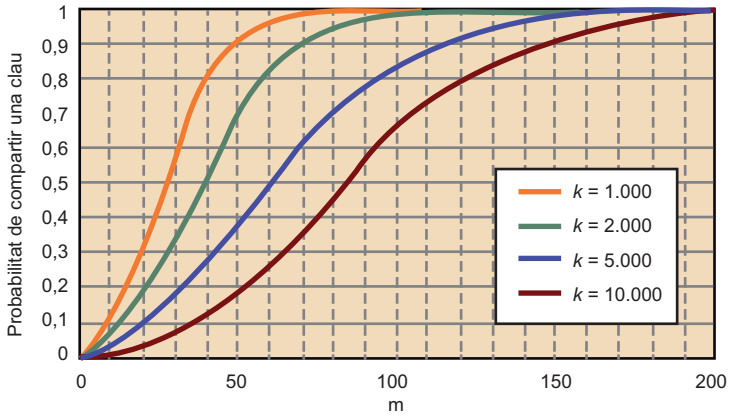
i per tant, assumint que $m = m_1 = m_2$ i simplificant l'expressió 1 obtenim:

$$P\{S_1 \cap S_2 \neq 0\} = 1 - \frac{(1-\frac{m}{k})^{2(k-m+\frac{1}{2})}}{(1-\frac{2m}{k})^{(k-2m+\frac{1}{2})}} \quad (2)$$

Exemple

La figura 8 il·lustra els valors de l'expressió 2 per k = 1.000, k = 2.000, k = 5.000 i k = 10.000. Com es pot veure, la probabilitat que dos usuaris comparteixin una clau creix ràpidament amb m. Per exemple, per a k = 1.000 claus només s'han de distribuir grups de 26 claus entre els nodes per tal que la probabilitat que dos veïns comparteixin una clau sigui més gran del 50%. Si incrementem en ordre de magnitud la dimensió de la piscina de claus (k = 10.000), el nombre de claus que s'han de distribuir entre els usuaris per a tenir una $P > 50\%$ és m = 83, que només és 3,2 vegades el nombre de claus distribuïdes en el cas d'una piscina k = 1.000. Així doncs, la solució de la piscina de claus és força escalable.

Figura 8. Predistribució de claus aleatòria



Durant la fase d'inicialització dels protocols basats en piscina de claus s'estableixen les mides de la piscina i de l'anell de claus, de manera que sigui fàcil trobar dos veïns que comparteixin una clau, però sigui difícil trobar-ne tres

o més que també la comparteixin. A partir d'aquí es genera la piscina i de manera aleatòria es va assignant un anell de claus a cada sensor.

Quan tots els sensors ja estan desplegats s'inicia la fase d'establiment directe d'una clau. En aquesta fase els sensors porten a terme un descobriment dels nodes veïns amb qui comparteixen una clau comuna. El descobriment de claus es pot implementar assignant un identificador curt a cada clau de S abans del desplegament i fent que cada node enviï en difusió el conjunt d'identificadors corresponent a les claus del seu anell. Quan dos veïns descobreixen que comparteixen una clau poden verificar que tots dos tenen la clau executant un protocol de repte-resposta. La clau que comparteixen s'estableix aleshores com la clau d'enllaç entre ells.

Algunes parelles de veïns poden no tenir cap clau en comú. Si passa això el que fan és executar la fase d'establiment multisalt d'una clau en la qual els nodes acorden una clau compartida a través d'una ruta formada per nodes de la xarxa que ja tenen una clau d'enllaç definida i que permeten unir la parella de sensors. La clau compartida la defineix un dels dos nodes a partir d'una clau no usada del seu anell de claus.

L'avantatge principal d'aquest esquema és que els sensors no han de fer càlculs intensius, és escalable i és fàcil afegir nodes al sistema. El problema és que si un nodes és capturat i compromès, les seves claus poden ser usades per altres nodes. D'altra banda, si el node compromès participa en l'establiment multisalt d'una clau per un node veí, la clau del veí també resulta compromesa. Finalment, aquest sistema no proporciona autenticació node a node. Això vol dir que un node pot establir unes claus compartides amb els seus veïns, però no sap exactament qui són aquests veïns. Així doncs, l'expulsió de nodes maliciosos o compromesos de la xarxa no és possible.

Predistribució de claus *q-composta*

Una manera per a millorar la capacitat de recuperació de l'esquema bàsic de la piscina de claus davant atacs de captura de nodes és utilitzar una predistribució aleatòria de claus q-composta. Aquest esquema es diferencia de l'esquema bàsic en tant que els nodes han de tenir com a mínim q claus en comú en els seus anells de claus per tal de poder establir una clau d'enllaç. La clau d'enllaç es calcula com el resum de totes les claus compartides.

Aquest esquema és més robust al compromís d'una clau d'enllaç, ja que per això l'atacant ha de ser capaç de capturar un node i descobrir quin subgrup de l'anell de claus s'està utilitzant en cada enllaç. D'altra banda, els requisits del protocol també són més grans, ja que la probabilitat d'establir una clau directament amb un altre usuari és més petita que en l'esquema de la piscina de claus (és menys probable tenir q claus comunes amb un veí que tenir-ne una) i per solventar-ho es pot incrementar la mida de l'anell de claus (amb

Quan $q = 1$ l'esquema q-compost té el mateix comportament que l'esquema bàsic de la piscina de claus.

la necessitat consegüent de més memòria) o la mida de la piscina de claus ha de ser més petita (amb l'efecte negatiu que això tindria sobre la captura de nodes).

Reforç de la clau a través de multicamí

Una altra alternativa de millora a l'esquema bàsic de la piscina de claus és establir les claus a través de camins disjunts. Aquest protocol s'aplica quan ja s'ha pogut establir una clau d'enllaç directe k_0 a través de la piscina. Aleshores un dels nodes identifica un conjunt de j camins disjunts cap a l'altre node i li envia j claus compartides $k_1, k_2, \cdots, k_j$, una per cada camí. Cada clau està protegida salt a salt a través de les claus d'enllaç dels nodes que col·laboren en la retransmissió. Quan el node receptor rep les claus, tots dos calculen la clau final d'enllaç com a $K = k_0 \oplus k_1 \oplus \cdots \oplus k_j$.

Aquest sistema comporta una sobrecàrrega més gran per a l'establiment de claus, però té l'avantatge que comprometre una clau d'enllaç és més costós. Per a fer-ho un atacant necessitaria comprometre com a mínim una clau d'enllaç de cada una de les rutes que participen en l'establiment de claus.

3.1.3. Predistribució aleatòria de claus basada en polinomis

L'objectiu d'aquest esquema és minimitzar la informació sensible que un atacant obté quan captura un sensor. La idea és que no sigui possible obtenir informació útil si no es captura un nombre mínim de nodes, i per això s'utilitza la criptografia llindar.

A continuació descrivim com funciona la predistribució de claus basada en polinomis i després mostrem com es pot combinar amb una piscina de claus:

1) Escollim un polinomi bivariat de grau t sobre un cos finit $GF(p)$, on p és un nombre primer gran tal que $f(x,y) = f(y,x)$.

$$f(x,y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$$

2) Cada sensor és precarregat amb un fragment del polinomi $f(id_i,y)$, on id_i és l'identificador del sensor i.

3) Qualsevol dels dos nodes i, j poden calcular una clau compartida. Per a fer-ho:

- i avalua el seu fragment de polinomi en el punt id_j, i obté $f(id_i,id_j)$
- j avalua el seu fragment de polinomi en el punt id_i, i obté $f(id_j,id_i)$

Aquest esquema és incondicionalment segur i resistent a atacs de t coalicions, és a dir, qualsevol atac que comprometi fins a t nodes no pot obtenir cap informació sobre les claus compartides calculades per qualsevol parella dels nodes compromesos. L'esquema no presenta sobrecàrrega de comunicacions per als sensors, però sí que té requisits de memòria rellevants, en concret $(t + 1)log(p)$ per node. Això fa que el llindar t estigui limitat per la capacitat de memòria dels sensors.

Exemple

Es dissenya la distribució de claus per a una xarxa de sensors que pugui suportar atacs de fins a tres nodes de coalició. S'escull el polinomi de treball següent a GF(139): $f(x,y) = 3 + 5xy + 18x^2y^2 + 2x^3y^3$. Els sensors id_{24} i id_{15} reben els fragments següents del polinomi:

- Sensor id_{24}: $f(y) = 3 + 120y + 82y^2 + 126y^3$
- Sensor id_{15}: $f(y) = 3 + 75y + 19y^2 + 78y^3$

Un cop desplegada la xarxa de sensors, els nodes id_{24} i id_{15} són veïns i volen establir una clau d'enllaç. Aleshores el que fan és:

- Sensor id_{24}: $f(15) = 3 + 120.24 + 82.24^2 + 126.24^3 = 8$
- Sensor id_{15}: $f(24) = 3 + 75.15 + 19.15^2 + 78.15^3 = 8$

La clau compartida és **8**.

Per tal de poder oferir un esquema més robust a atacants sense incrementar-ne els requisits de memòria, el que es fa és combinar la idea de la predistribució de claus basada en polinomis amb la piscina de claus. És el que anomenem esquemes de predistribució aleatòria de claus basada en polinomis. El seu funcionament és el següent:

1) Generem uns piscina S de polinomis bivariats de grau t.

2) Per a cada sensor i escollim un subconjunt de m polinomis de la piscina, i precarreguem el sensor amb un fragment de cada polinomi calculat al punt id_i.

3) Dos nodes que tinguin fragments d'un mateix polinomi f podran establir una clau compartida $f(i,j)$.

4) Si dos nodes no tenen polinomis en comú poden establir una clau compartida a través d'un camí d'intermediaris.

Com en l'esquema anterior, si un atacant compromet un node pot obtenir informació de les claus d'enllaç que aquest node utilitza amb els seus veïns, però en cap cas obtindrà informació reveladora sobre les claus compartides de qualsevol altre parella de claus. Per tal de comprometre un polinomi un adversari necessitaria obtenir $t + 1$ fragments d'aquest polinomi. Aconseguir aquests $t + 1$ fragments no és fàcil, ja que tots els nodes tenen fragments de polinomis diferents de la piscina i, per tant, s'han de capturar molts nodes per poder aconseguir $t + 1$ fragments d'un sol polinomi. Això si, quan s'aconsegueix comprometre un polinomi, totes les parelles de nodes que utilitzin fragments d'aquest polinomi es veuran afectades.

Els requisits de memòria d'aquest esquema són $m(t + 1)log(p)$. Veiem que es diferencia de l'esquema de predistribució de claus basada en polinomis en només un factor m, però en canvi pot suportar molts més atacs i, per tant, el valor de t pot ser molt menor.

3.1.4. Predistribució de claus basada en matrius

Els esquemes de predistribució de claus basada en matrius es basen en l'esquema de Blom proposat el 1985. En aquest esquema, una matriu simètrica $K_{n\times n}$ guarda totes les claus d'un grup de n nodes, on cada element k_{ij} és la clau d'enllaç entre els nodes i i j. L'esquema de Blom pot resistir atacs de fins a t nodes. Totes les claus d'enllaç dels nodes no compromesos d'una xarxa es mantenen segures mentre no hi hagi més de t nodes corromputs.

La predistribució de claus de Blom funciona de la manera següent:

1) Generem una matriu $G_{(t+1)\times n}$ sobre un cos finit $GF(p)$, on n és la dimensió de la xarxa i p és un número primer més gran que la longitud de la clau desitjada i més gran que n. G és una matriu pública i pot estar compartida entre diferents sistemes.

2) L'estació base de la xarxa de sensors genera una matriu simètrica aleatòria $D_{(t+1)\times(t+1)}$ sobre el cos finit $GF(p)$. D és una matriu secreta.

3) A partir de la matriu secreta i la pública, l'estació base calcula una matriu $A_{n\times(t+1)}$ i $K_{n\times n}$:

a) $A = (DG)^T$

b) $K = AG$. Podem comprovar que K és una matriu simètrica, ja que $K = AG = (DG)^T G = G^T D^T G = G^T DG = G^T A^T = (AG)^T = K^T$.

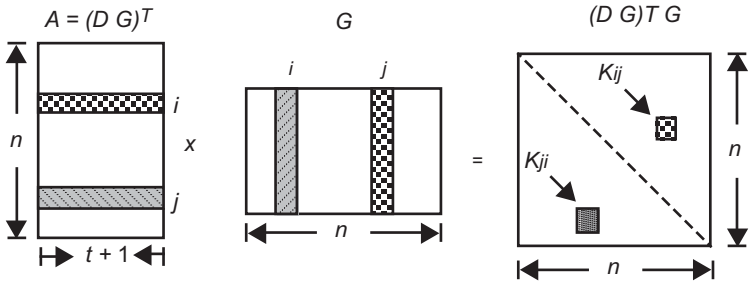
4) Cada node i guarda la fila i de la matriu A.

En cop desplegada la xarxa qualsevol dels dos nodes i i j poden calcular una clau compartida K_{ij} intercanviant les seves columnes de la matriu G en clar i fent:

- i calcula $A(i,.)G(.,j) = K_{ij}$
- j calcula $A(j,.)G(.,i) = K_{ji} = K_{ij}$

Observa que la clau és el producte d'una fila de la matriu privada i una columna de la matriu pública, i com que les files de A sempre es mantenen en secret, un observador que capturi el tràfic entre els nodes i i j no podrà deduir la clau d'enllaç que estableixen entre ells (vegeu la figura 9).

Figura 9. Generació de claus en l'esquema de Blom



Es pot demostrar que l'esquema de Blom és t-segur si les $t + 1$ columnes de la matriu G són linealment independents. Una possible manera de construir la matriu G és utilitzant una matriu de Vandermonde de la manera següent:

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^n \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^n)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s^t & (s^2)^t & (s^3)^t & \cdots & (s^n)^t \end{bmatrix}$$

Exemple

Tenim una xarxa de sensors amb $n = 5$ nodes que utilitza una predistribució de claus de Blom amb les dades següents:

$$p = 17, \quad t = 3, \quad G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 10 & 13 \\ 1 & 4 & 16 & 13 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 6 & 2 & 1 \\ 6 & 3 & 8 & 10 \\ 2 & 8 & 2 & 13 \\ 1 & 10 & 13 & 4 \end{bmatrix}$$

Aleshores, la matriu privada de claus és:

$$A = (DG)^T mod 17 = \begin{bmatrix} 10 & 10 & 8 & 11 \\ 6 & 8 & 8 & 8 \\ 8 & 12 & 5 & 1 \\ 14 & 2 & 0 & 8 \\ 5 & 15 & 16 & 11 \end{bmatrix}$$

Si els usuaris 2 i 4 volen establir una clau d'enllaç entre ells:

1) L'usuari 2 calcula k_{24} amb la seva fila privada de A, i la matriu pública G, de manera que:

$$\begin{bmatrix} 6 & 8 & 8 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 8 \\ 10 \\ 13 \end{bmatrix} mod 17 = 16$$

2) L'usuari 4 calcula k_{42} amb la seva fila privada de A, i la matriu pública G, de manera que:

$$\begin{bmatrix} 14 & 2 & 0 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} mod 17 = 16$$

Per tal de poder oferir un esquema més robust a atacants podem combinar l'esquema de Blom amb la piscina de claus, amb la qual cosa obtenim un esquema de predistribució aleatòria de claus basada en matrius. El seu funcionament és el següent:

1) Generem una matriu pública $G_{(t+1)\times n}$ sobre un cos finit $GF(p)$ com en el cas anterior.

2) Generem k matrius simètriques aleatòries $D_{(t+1)\times(t+1)}$ sobre el cos finit $GF(p)$.

3) A partir de G i les k matrius D, obtenim k matrius $A_{n\times(t+1)}$ que conformen el contingut de la piscina: $A_V = (D_V G)^T$.

4) Per a cada sensor i escollim un subconjunt de m matrius A de la piscina, i precarreguem al sensor la fila i de les matrius seleccionades (per exemple, $A_V(i,.)$ per a cada V seleccionada).

5) Dos nodes que tinguin seleccionada una matriu comuna A_V poden calcular una clau compartida utilitzant l'esquema de Blom.

6) Si dos nodes no tenen una matriu en comú poden establir una clau compartida a través d'un camí d'intermediaris.

3.2. Agregació de dades

Les dades recollides pels sensors s'han de processar d'alguna manera per a poder-ne treure informació interessant. Moltes vegades no fa falta enviar totes les dades recollides pels sensors (una a una) a l'aplicació final, sinó que les dades es poden presentar en un sol punt de vista que sintetitzi tot el recollit pels sensors (suma de totes les dades, mitjana, etc.).

L'agregació de dades és una manera eficient d'estalviar energia als sensors. Tanmateix, també crea un conjunt de nous riscos de seguretat:

1) Si les dades han de viatjar a través d'una xarxa multisalts, els nodes que reenvien la informació dels sensors poden manipular les dades.

2) Es perd la identificació de les fonts d'informació, fent que la detecció de nodes maliciosos sigui més difícil.

3) El node agregador pot falsificar el resultat de l'agregació de dades.

Per a evitar aquests problemes necessitaríem proporcionar serveis d'integritat i autenticitat de dades des que les dades surten del sensor fins que arriben a l'aplicació final. Generalment això es podria fer amb codis d'autenticació de missatges o amb signatures digitals, però ambdues opcions són massa costoses per a fer-les amb totes les dades que surten dels sensors. En les xarxes de sensors totes les operacions han de ser molt eficients ateses les limitacions dels nodes.

En aquest subapartat veurem com els mètodes basats en funcions resum proporcionen signatures eficients en entorns molt limitats.

3.2.1. Arbre de Merkle

Un arbre de Merkle és un arbre de resums que permeten la preautenticació d'un conjunt de valors amb una sola signatura digital (com en el cas de les cadenes resum).

Ralph Merkle

L'arbre de Merkle és una construcció introduïda per Ralph Merkle el 1979 amb l'objectiu principal de fer més eficient el poder manejar múltiples firmes d'un sol ús.

L'arbre de Merkle és un arbre en les fulles del qual hi ha els valors de resum de blocs de dades de, per exemple, els valors obtinguts per un sensor.

L'esquema de l'arbre de Merkle funciona de la manera següent. Siguin $x_1,x_2,\ldots,x_{2^l}$ els valors que volem autenticar. En primer lloc calculem un resum d'aquests valors amb una funció $h : \{0,1\}^* \rightarrow \{0,1\}^n$, de tal manera que obtenim $y_1,y_2,\ldots,y_{2^l}$, on $y_i = h(x_i)$. Aleshores assignem els valors y_i en les fulles d'un arbre binari. A més, a cada vèrtex intern d'aquest arbre hi assignem un valor que és calculat com el resum dels valors assignats als seus dos fills. Finalment, se signa el valor assignat a l'arrel de l'arbre.

Quan el propietari de l'arbre vol autenticar algun dels valors x_i revela el valor x_i i tots els valors assignats en els vèrtex germans en el camí entre y_i i l'arrel de l'arbre. El verificador pot fer un resum d'aquests valors en l'ordre apropiat i comparar el resultat amb el valor assignat a l'arrel (i del qual té una firma digital). Si els dos valors concorden, aleshores el valor x_i és autèntic (és a dir, es verifica que el valor prové de la mateixa entitat que va calcular l'arbre i va firmar digitalment l'arrel).

Atesa la propietat d'unidireccionalitat de les funcions resum, el fet de revelar un valor x_i i els valors assignats als nodes germans no permet calcular cap altre valor de $x_j, i \neq j$ de l'arbre. Observeu que això fa que els valors d'un arbre de Merkel es puguin revelar en qualsevol ordre, a diferència del que passa en una cadena resum.

Figura 10. Arbre de Merkle de nivell $K = 4$

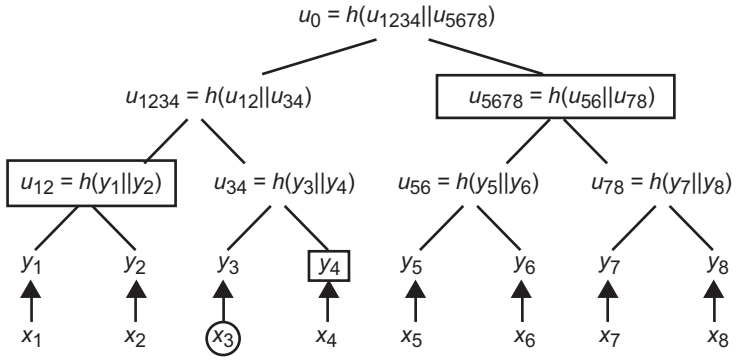


Figura 10

La figura mostra que el valor x_3 és autenticat revelant y_4, u_{12} i u_{5678}. El verificador pot calcular $h(h(u_{12}||h(h(x_3)||y_4))||u_{5678})$ i comparar el resultat amb el valor u_0 assignat a l'arrel de l'arbre. Si coincideixen, x_3 és autenticat favorablement.

En el cas d'una xarxa de sensors, suposem que els sensors s'associen en una forma jeràrquica. Cada sensor genera una lectura de dades i un resum de la lectura. Les dades i el resum es passen al node pare. El pare genera un resum de tots els resums que ha rebut dels seus fills i ho envia al node següent de l'arbre. El procés continua fins que el node arrel obté un resum que agrega totes les dades.

En qualsevol moment l'estació base pot demanar a algun sensor que li enviï la lectura de dades d'un cert instant per comprovar que les dades que va rebent dels nodes recol·lectors són correctes.

4. Protocols tolerants a retards i a interrupcions

L'arquitectura i protocols actuals d'Internet han demostrat ser perfectament adequats per a una gran varietat d'aplicacions, però sempre que hi hagi un bon nivell de connectivitat entre les parts en comunicació. En canvi, el rendiment d'aquests protocols es pot degradar fàcilment, o fins i tot poden deixar de funcionar completament, en escenaris on no hi hagi un canal continu entre les parts que es comuniquen, o on el retard de les comunicacions sigui significatiu. Dos exemples d'aquestes situacions degradades es donen quan usuaris de dispositius mòbils intenten comunicar-se mentre estan situats en xarxes sense fils ad hoc diferents, o quan les aplicacions que intenten comunicar-se no estan connectades de manera simultània, o tenen un accés temporal discontinu a xarxes de comunicació. En el primer cas parlem d'una connectivitat intermitent en l'espai i en el segon cas, d'una connectivitat intermitent en el temps.

La solució per als escenaris de connectivitat intermitent que preveu un enfocament més global, i que més èxit ha tingut fins al moment, és la basada en els protocols tolerants a retards i a interrupcions (*Delay-and Disruption-Tolerant Networking*, DTN) (3; 2).

En l'enfocament DTN les comunicacions es realitzen exclusivament en el nivell d'aplicació, entre nodes adjacents quan aquests tenen l'oportunitat de comunicar-se, i sense necessitat d'una connectivitat contínua d'extrem a extrem. Aquest mètode permet les comunicacions fins i tot en les situacions més adverses, però a costa de perdre la interactivitat, ja que cada pas d'una comunicació pot prendre un temps no limitat.

La xarxa DTN de la NASA

L'exemple més mediàtic de protocols DTN, encara que sens dubte el menys quotidià dels exemples, és el de la comunicació entre sondes enviades a l'espai exterior i la posterior retransmissió de dades a les bases terrestres (per exemple en les últimes missions a Mart). La NASA va fer públic a finals del 2008 que va provar amb èxit la primera xarxa de comunicacions en l'espai exterior modelada segons els principis d'Internet, però usant protocols DTN. En aquestes situacions de distàncies enormes, el temps de propagació dels missatges és molt superior als màxims establerts per TCP i per tant, fins ara, les comunicacions interplanetàries havien de ser modelades de manera específica per a cada missió, sense poder aprofitar l'enorme potencial que comporta l'experiència de la comunitat Internet.

Enllaç d'interès

Podeu accedir als detalls l'experiència de la NASA a l'ús d'una xarxa DTN a l'adreça http://www.jpl.nasa.gov/news/news.cfm?release=2008-216

Hi ha molts altres escenaris més "terrestres" on els protocols DTN poden ser d'utilitat. De fet, aquests escenaris de connectivitat intermitent són cada vegada més freqüents a mesura que les xarxes sense fil i els dispositius mòbils es tornen més omnipresents en els entorns urbans de la nostra societat. També són més freqüents a mesura que volem estendre la cobertura d'Internet cap a entorns no considerats anteriorment, com ara àrees rurals aïllades o de països subdesenvolupats, sense (bones) infraestructures de comunicació; o àrees de països àrtics, sense cobertura de satèl·lit; o dispositius diminuts com els sensors (l'anomenada Internet de les coses) amb limitada capacitat comunicativa. En alguns casos les aplicacions busquen expressament ser independents de les infraestructures convencionals de comunicació (GSM/GPRS/UTM/WiMAX), tant per a estalviar els costos de comunicació associats com per a superar el seu model inherent de comunicacions contínues entre els extrems.

Hi ha diferents tipus de DTN associats a les característiques de les aplicacions i dels entorns de comunicació. En el cas de satèl·lits i sondes espacials les trajectòries són predictibles i, per tant, es pot saber per endavant els moments en què la connexió és possible. En canvi, si considerem una xarxa de sensors instal·lats al voltant d'un llac per controlar-ne el nivell de contaminació, no es pot predir els instants en què aquests sensors quedaran coberts d'aigua o sense possibilitat de carregar la seva bateria per culpa del mal temps, i per tant sense possibilitat de comunicar-se. En conseqüència, per a cada situació serà necessari identificar una aproximació al concepte DTN que s'ajusti a les seves característiques: mòbils o estàtics, amb disponibilitat contínua o interrompuda, etc.

Lectura complementària

Farrell, S; Cahill, V. (2006) *Delay- and Disruption-Tolerant Networking*. Artech House. El capítol 3 d'aquest llibre conté una detallada descripció d'una dotzena d'aplicacions DTN.

Hi ha un grup de recerca dins l'IRTF (Internet Research Task Force) dedicat als protocols DTN. Les activitats del DTNRG (Delay and Disruption Tolerant Networking Research Group) han consistit en la publicació d'esborranys de dues propostes de protocols (*Bundle Protocol*, BP, i *Licklider Transmission Protocol*, LTP), en el desenvolupament d'implementacions d'aquests protocols i en la publicació de fins a onze estàndards d'Internet (*Request for Comments*, RFC) de caràcter informatiu i experimental des de l'any 2007 fins al 2011. El protocol BP està pensat per a necessitats DTN terrestres i l'LTP, per a necessitats DTN a l'espai exterior.

Enllaç d'interès

Podeu accedir a les activitats del grup de recerca en DTN de l'IRTF a l'adreça http://www.dtnrg.org.

Atès que les situacions de connectivitat intermitent poden ser molt diverses, hi ha altres propostes d'arquitectures de protocols DTN que no segueixen el model DTNRG. Els subapartats següents descriuran les arquitectures DTN més significatives: l'arquitectura del DTNRG, l'arquitectura col·lapsada de Haggle, i l'arquitectura basada en missatges actius.

4.1. Arquitectura DTN proposada pel DTNRG

L'arquitectura "oficial" pels protocols DTN es defineix en el RFC 4838 (2). Aquest estàndard situa les funcionalitats necessàries per a la tolerància a re-

tards i a interrupcions en una capa pròpia integrada en la capa d'aplicació, o com a mínim per sobre de la capa de transport. En aquest sentit, l'enfocament DTN representa una capa sobreposada (*overlay network*) que pot situar-se per sobre de qualsevol combinació de xarxes existents.

RFC4838

L'estàndard d'Internet (*Request for Comments*, RFC) amb numeració AAAA es pot descarregar de http://www.rfc-editor.org/rfc/rfcAAAA.txt.

Podem veure aquesta arquitectura a la figura 11*b*. En aquesta figura també es mostra com es fa la transferència de dades en el model DTN, comparada amb una transferència de dades a la Internet actual, que es mostra en la figura 11*a*.

Figura 11. Arquitectura de protocols DTN

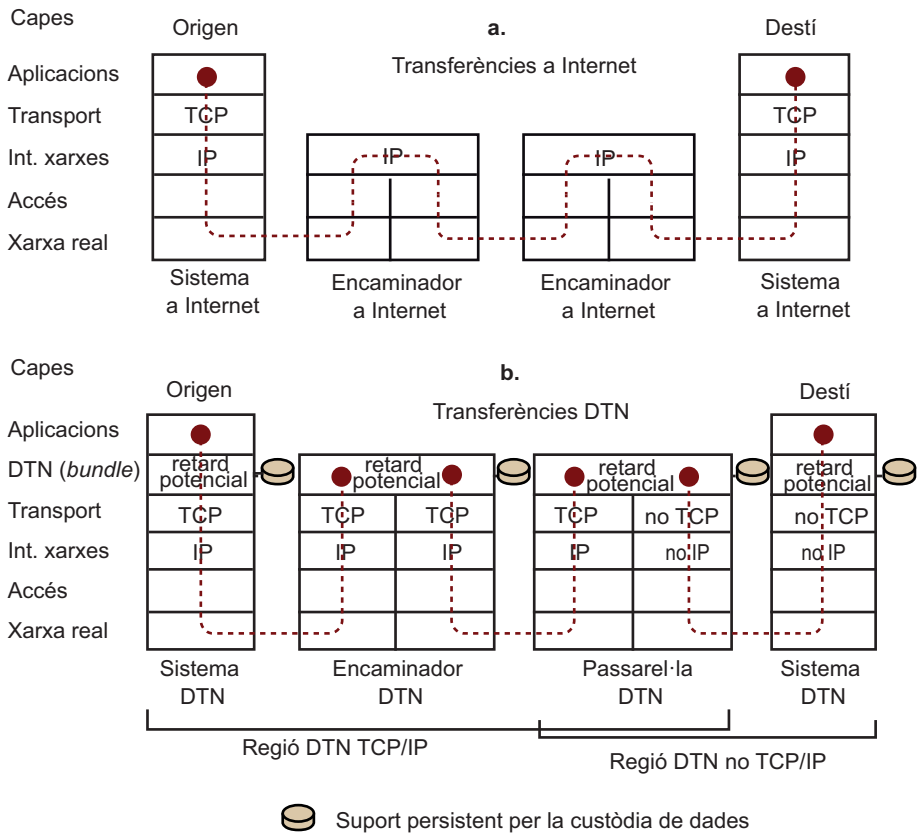


Figura 11

a. Esquema de comunicació tradicional d'Internet.
b. Esquema de comunicació DTN, amb la nova capa DTN entre les capes de transport i d'aplicacions.

A Internet (figura 11*a*) hi ha un seguit d'encaminadors o *routers* que interconnecten un conjunt de xarxes diferents fins a oferir una visió global i única a les aplicacions. Aquestes aplicacions disposen d'una comunicació d'extrem a extrem (a través de connexions TCP o de datagrames d'usuari UDP) que els permet la comunicació directa, permanent i continua. Els datagrames IP surten de la màquina d'un extrem i van passant pels encaminadors intermedis fins a arribar a la màquina de l'altre extrem.

En un entorn DTN (figura 11*b*), com que no hi connectivitat d'extrem a extrem, la capa DTN proporciona una connectivitat intermitent entre les diferents xarxes subjacents. Noteu que aquestes xarxes subjacents poden arribar a ser nodes individuals, on cada node es comporta com a encaminador o passarel·la DTN. La capa DTN recull les dades d'una aplicació i les agrupa fins a formar una unitat de dades anomenada farcell (*bundle*), que serà emmagatze-

mat dins de la capa en un suport persistent, i que no es mourà cap al node o encaminador següent fins al moment en què la comunicació sigui possible. En aquesta comunicació no hi ha restriccions temporals ni restriccions de connectivitat de cap tipus. Es diu que cada node actua com a custodi dels farcells de dades que estan en trànsit des del node origen fins al node destinació.

La capa DTN proporciona un mecanisme d'interconnexió de xarxes similar a l'ofert per la capa IP, però en el nivell d'aplicació. Les capes inferiors a la DTN no han de ser necessàriament les capes d'Internet (TCP/IP) tal com les coneixem, sinó que poden ser les capes de qualsevol família de protocols. A la figura 11*b* es distingeix entre un encaminador DTN, que interconnecta dues xarxes amb protocols TCP/IP (internets) que a més tenen la capa DTN, i una pasarel·la DTN, que interconnecta una internet amb una xarxa amb protocols no TCP/IP. Les funcions de custòdia dels farcells de dades són les mateixes tant en els encaminadors com en les pasarel·les.

4.2. Arquitectura col·lapsada Haggle

Haggle va ser un projecte europeu, que va finalitzar el 2010, que es proposava oferir aplicacions per a la comunicació intermitent entre dispositius intel·ligents sense utilitzar cap tipus d'infraestructura. La comunicació es diu que és oportunística perquè s'assumeix que només és possible quan els dispositius entren en cobertura de la xarxa sense fils que incorporin, és a dir, els dispositius es comuniquen només quan tenen l'oportunitat de fer-ho i no es preocupen de les desconnexions. Un exemple seria la comunicació que es pot establir entre dos *smartphones* de dues persones que coincideixen uns moments en una andana de tren.

Enllaç d'interès

Es pot accedir a tota la informació i resultats del projecte europeu Haggle a http://www.haggleproject.org/.

Figura 12. Arquitectura DTN de Haggle

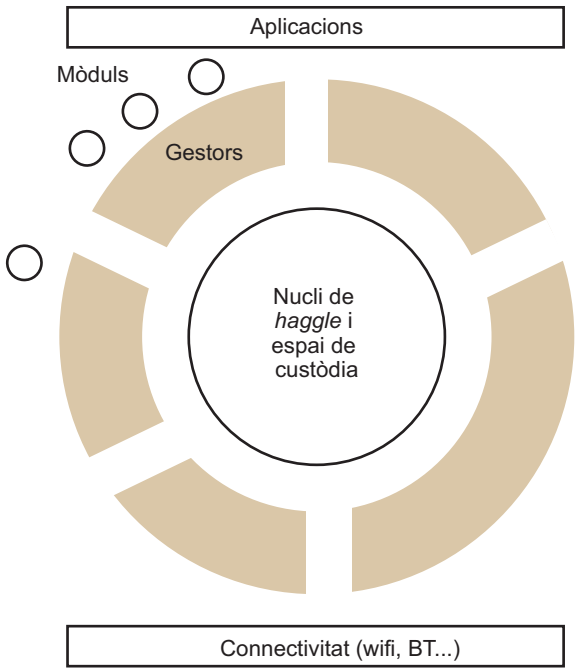


Figura 12

La capa Haggle intermèdia treballa directament sobre la xarxa sense fils inferior i per sota de les aplicacions. Conté una sèrie de gestors i mòduls per a rebre, emmagatzemar i reenviar les dades d'una manera oportunística.

Haggle preveu una arquitectura de tres capes (vegeu la figura 12):

- una capa superior de les aplicacions i una capa inferior de connectivitat a través de qualsevol tecnologia sense fils disponible;

- entre la capa de connectivitat i la d'aplicacions no hi ha cap capa tradicional TCP/IP sinó una sola capa (per això es diu arquitectura col·lapsada) que proporciona tota la funcionalitat per a rebre un farcell de dades o més d'un en una oportunitat de comunicació, custudiar-los, encaminar-los i reenviar-los quan es tingui una nova oportunitat de comunicació.

4.3. Arquitectura DTN amb missatges actius

Hi ha una dependència molt gran entre l'aplicació concreta que utilitzarà la xarxa DTN i l'algorisme d'encaminament. Això no es veu en xarxes molt connectades, com ara Internet, ja que el millor algorisme per a encaminar, és a dir, per a decidir quin és el millor veí per a transmetre-li un missatge per tal que arribi a la seva destinació, és unic i conegut. Tots els encaminadors d'Internet segueixen el mateix algorisme, descrit a l'estàndard del protocol IP, que va bé per a qualsevol aplicació.

En el cas de les DTN, però, la situació és ben diferent. Ja no es té una única xarxa connectada, sinó que hi haurà moments de discontinuïtat en què el missatge haurà de ser retingut. Quan arriba un conjunt de veïns a aquest node, cal decidir quina és la millor opció, i aquí és on rau una de les dificultats més grans de les DTN. Com que hi ha moltes possibilitats de tenir xarxes no connectades continuament, no podem establir quina serà l'estratègia que anirà millor sempre. D'altra banda, cada aplicació pot tenir necessitats d'encaminament diferents, i pot disposar d'informació addicional que permetria fer una millor tria del node següent perquè el missatge arribi a la seva destinació de la millor manera.

Alguns exemples de DTN

SENDT: monitorització de la contaminació de llacs. En aquest projecte s'utilitzen sensors sense manteniment per a mesurar la contaminació de llacs irlandesos. Les barques dels pescadors serveixen de mules de dades, recol·lecten tota la informació dels sensors i la porten a terra (13).

Zebranet: seguiment de zebres a Kenya. Les zebres porten un collaret amb un receptor GPS que va enmagatzemant la seva posició cada pocs minuts. Cada dues hores s'intenta enviar tota la informació a alguna zebra propera perquè la retransmeti a una altra, i finalment arribi a un col·lector de dades (17).

Gripau gegant a Austràalia. Per tal de monitoritzar la invasió del gripau gegant a Austràlia s'utilitzen uns sensors que detecten el cant d'aquesta espècie. Unes mules de dades recullen la informació recopilada per aquests sensors i la transporten a una estació base (21).

Les arquitectures presentades fins ara tenen un problema comú: els nodes que reencaminen els missatges ho fan amb un algorisme d'encaminament comú que ha d'estar disponible i configurat en cada un dels nodes. Si volem utilitzar una mateixa xarxa DTN per a dues aplicacions diferents, o per a una única aplicació amb més d'un tipus de missatge amb encaminament diferenciat, ens trobarem que la xarxa no s'estarà utilitzant de la millor manera possible per a tothom, i que fins i tot alguns missatges podríen tenir problemes per a arribar a la seva destinació. Això també s'aplica per a diferents escenaris. Una xarxa DTN on els nodes es van trobant amb certa periodicitat requerirà un protocol d'encaminament diferent d'aquella en què els nodes no poden predir quines seran les trobades futures.

Històricament, això ja ha passat en alguna altra ocasió. Per exemple, per a les xarxes ad hoc sense fils tenim un gran nombre de protocols d'encaminament disponibles (AODV, AORP, ZRP, DSR, etc.) que funcionen bé, cada un per a un cert tipus d'aplicacions o d'escenaris. També passa en l'encaminament per l'esquema de direccionament multidestinació (*multicast*) d'IP. Aquí tornem a tenir diversos protocols (DVMRP, MOSPF, PIM-DM, PIM-SM, etc.) incompatibles entre ells que aniran bé per a escenaris o aplicacions concretes, però cap no pot aplicar-se sempre. En ambdós casos, ad hoc i multidestinació, l'ús simultani d'aquests protocols no és un cas previst, i sempre cal triar-ne un que afavorirà uns certs escenaris o aplicacions, i en desafavorirà d'altres.

Si volem una xarxa DTN de propòsit general hauríem de disposar d'una varietat d'algorismes d'encaminament coexistint, i que els missatges els utilitzessin segons les seves necessitats. Algunes arquitectures DTN s'aproximen a això amb uns protocols prefixats en els nodes que els missatges poden triar. El problema és que afegir nous protocols en una xarxa DTN amb nodes no connectats permanentment pot no ser possible, o requerir molt de temps. D'altra banda, d'aplicacions noves amb necessitats d'encaminament particulars, en poden aparèixer en qualsevol moment. Una alternativa són les xarxes DTN basades en missatges actius, que utilitzaran el codi mòvil per a resoltre aquests problemes.

4.3.1. Missatges actius

En les xarxes DTN basades en missatges actius, el codi d'encaminament que haurà de decidir quin és el millor node veí perquè un missatge continui el seu camí cap a una certa destinació, viatjarà juntament amb el mateix missatge. És a dir, cada missatge tindrà una part d'informació, que és la que haurà d'arribar a la destinació, i una part de codi, que s'executarà en cada node i que decidirà la millor opció entre una sèrie de nodes veïns (figura 13).

Figura 13. Nodes en arquitectures DTN basades en missatges actius

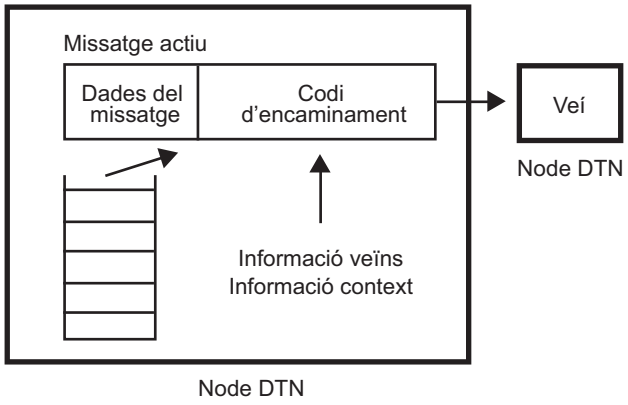


Figura 13

En les xarxes basades en missatges actius, la decisió sobre quin és el millor node veí per a reencaminar-se la pren el mateix missatge mitjançant el seu codi d'encaminament.

L'aplicació, per tant, podrà incidir en la manera d'utilitzar la xarxa DTN posant un cert algorisme en els missatges que utilitza. D'aquesta manera és possible tenir xarxes DTN de propòsit general en què cada aplicació no està obligada a seguir un protocol comú.

El canvi conceptual d'aquestes arquitectures respecte a d'altres de més clàssiques és profund. Fins ara, l'encaminament era una funció de la capa de xarxa del model OSI que utilitzava informació disponible només en aquesta mateixa capa. Amb les arquitectures de DTN basades en missatges actius, la informació utilitzada pot provenir de la mateixa capa, i de la capa d'aplicació, on és més probable tenir un millor coneixement de com està funcionant la xarxa DTN i de quina estratègia d'encaminament serà més encertada. A més, també hi ha disponible la informació sobre el context proporcionada pel mateix node, que pot ser decisiva a l'hora de determinar el node següent en algunes aplicacions.

En la figura 13 es pot veure l'esquema general de funcionament d'un node DTN utilitzant aquesta arquitectura. Els missatges pendents de ser enviats estan en una cua de sortida. Quan hi ha veïns a l'abast, el node DTN tria el missatge que serà enviat i n'executa el codi d'encaminament, i li passa com a argument la llista de veïns i altra informació contextual. La funció de decisió, en el codi del missatge, tornarà la llista de nodes als quals interessa saltar. Normalment aquesta llista serà només d'un element, però l'algorisme d'encaminament particular podria triar no anar a cap dels veïns disponibles, o saltar a més d'un (fent una clonació). Aleshores, el node DTN intentarà enviar el missatge al node seleccionat i passarà a processar el següent.

Per tal de contruir aquest tipus de xarxes DTN, cal que els nodes disposin d'un entorn d'execució de codi. Per als casos en què els missatges no portin codi, el node pot tenir una estratègia d'encaminament per defecte. Certs *frameworks* de desenvolupament faciliten molt la creació de nodes DTN amb suport de missatges actius. Ens referim a les plataformes d'execució d'agents mòbils, com ara JADE* i Mobile C**.

*** http://jade.cselt.it**
**** http://www.mobilec.org**

4.3.2. Un exemple pràctic: PROSES

PROSES (*Protocols for the Single European Sky*, 2010-2012) va ser un projecte finançat pel Ministeri d'Indústria, Comerç i Turisme del govern d'Espanya, i en què van participar socis acadèmics i de la indústria. L'objectiu del projecte era la creació d'una xarxa DTN basada en missatges actius de què poguessin formar part aviació comercial, aviació general i vehicles aeris no tripulats.

Tot i que pugui semblar que els avions comercials estan connectats constantment a través d'algun enllaç satel·lital, la realitat ens mostra que això no sempre és possible, i que quan ho és, el seu ús és molt limitat. L'enllaç satel·lital té un cost molt elevat, i s'utilitza principalment per a funcions de gestió aèrea. D'altra banda, aquest enllaç no sempre és possible, com quan les aeronaus volen lluny de l'equador. A partir de certes latituds, i en els vols que travessen les zones polars, no hi ha cobertura, i es deixa de tenir possibilitat de transmissió de dades digitals (s'utilitza aleshores la ràdio de veu convencional, amb moltes limitacions). L'equipament necessari per a realitzar aquestes comunicacions també resulta car, i generalment és inaccessible per a vehicles petits en aviació general.

El projecte PROSES planteja crear una xarxa DTN de propòsit general formada pels vehicles que ocupen l'espai aeri i basada en missatges actius. El programari creat està basat en l'entorn de desenvolupament d'agents JADE, i utilitza comunicacions sense fils IEEE 802.11 per a les transmissions entre vehicles. En la part experimental s'han utilitzat vehicles aeris no tripulats (figura 14), on un equip dedicat a bord actua com a node de la xarxa, i és independent de l'equip de control i pilot automàtic. A banda del petit ordinador, cada node té un dispositiu de connexió a la xarxa sense fils, una antena i un receptor GPS.

Figura 14

Vehicles aeris no tripulats (UAV) utilitzats en els vols experimentals del projecte PROSES, fotografiats en les instal·lacions d'un dels socis dels projecte. Aquests vehicles formaven una xarxa DTN aèrea, en què tenien connectivitat entre ells només si volaven a prop.

Figura 14. Vehicles aeris no tripulats (UAV) en PROSES

Cada node va anunciant la seva presència periòdicament, a través de missatges balisa (*beacon*). En aquests missatges s'envia informació sobre el node mateix, que podrà ser utilitzada per a l'encaminament. El centre de coordinació del

projecte per als experiments estava situat dins d'una furgoneta preparada per a contenir aquests equipaments (figura 15), i actuava com un node més de la xarxa.

Figura 15. Estació de control i detall de l'equipament

Figura 15

A l'esquerra, la base de control dels experiments de PROSES, que estava dins una furgoneta preparada especialment. Aquesta base era un node més de la xarxa DTN. A la dreta, detall de l'ordinador muntat sobre un dels UAV.

En el marc del projecte s'han desenvolupat alguns protocols i aplicacions per a analitzar el funcionament de la xarxa i comparar-lo amb els resultats previs obtinguts en les simulacions.

Una aplicació consistia a enviar missatges originats asíncronament en els UAV i que havien d'arribar a terra al més aviat possible. En aquest cas, el protocol d'encaminament que portaven els missatges triava els nodes amb un temps previst d'arribada a destinació (ETA) menor.

Una altra aplicació enviava un missatge a tots els UAV que complissin unes condicions. El protocol d'encaminament, a diferència del cas anterior, era epidèmic, i els missatges anaven a tots els veïns disponibles.

4.4. Investigacions en marxa i problemes oberts

Hi ha molta investigació encara en desenvolupament en el camp dels protocols DTN. A part del desenvolupament de les aplicacions DTN, hi ha dos problemes oberts en aquests protocols que tenen molta importància: l'encaminament i la seguretat.

Pel que fa a l'encaminament, cap dels protocols especificats, ni el BP ni l'LTP, detalla com es poden establir les rutes entre els nodes que es comuniquen. El tractament de la connectivitat intermitent de les DTN requereix un canvi de paradigma, passant del model clàssic d'encaminament *store-and-forward* a un nou model *store-carry-forward*. En aquest nou model, quan un farcell de dades arriba a un node, el salt següent per a aquest farcell pot no estar disponible immediatament, i per tant el node haurà d'emmagatzemar aquest farcell i carregar amb ell possiblement durant un període de temps considerable. La

dificultat en el disseny de protocols que lliurin els farcells de manera eficient i satisfactòria a les seves destinacions rau a determinar, per a cada farcell, el millor node i el millor moment per a reenviar, aspectes no evidents quan els nodes no tenen connexió contínua amb altres nodes. Tota aquesta nova funcionalitat en l'encaminament se situa en la capa d'aplicació o a nivell de la capa sobreposada en l'arquitectura del DTNRG. Actualment hi ha una gran activitat en aquest problema, amb enfocaments molt diversos, segons es considerin xarxes de dispositius fixos o de dispositius mòbils, i tant si són amb evolució determinista (encaminament prefixat) o amb evolució estocàstica (encaminament probabilístic). Per a dispositius fixos una possibilitat és usar el model de mules de dades de les xarxes clàssiques de sensors, és a dir, elements mòbils que recullen les dades d'aquests sensors i els condueixen a la seva destinació. La manera de combinar aquesta tria de protocol d'encaminament de manera dinàmica, adaptant-se al tipus de missatge i a les característiques específiques de la mateixa xarxa semblen conduir cap a les arquitectures DTN basades en missatges actius.

Aquestes arquitectures que utilitzen codi en els missatges per a fer l'autoencaminament* tenen una sèrie d'avantatges clars, com ara la creació de xarxes DTN de propòsit general o la coexistència de protocols d'encaminament, però encara no solucionen totalment els problemes. Veiem algunes línies de recerca en aquest sentit.

*** Vegeu el subapartat 4.3.**

Per tal de funcionar tan bé com sigui possible, molts protocols d'encaminament han d'intercanviar informació entre nodes. Aquesta informació és del nivell de xarxa, i és entesa per tots els nodes. Ara bé, si s'utilitzen missatges actius, la informació utilitzada per l'encaminament pertany directament a la cap d'aplicació. Això té dues implicacions directes: la primera, que la quantitat d'informació ja no depèn del nombre de nodes de la xarxa, sinó del nombre d'aplicacions, que pot ser sensiblement superior; la segona implicació és que els nodes no entendran aquesta informació, i per tant no podran operar-la (fusió d'informació, descartar redundàncies, adaptar paràmetres pròpis, etc.). Com que molta d'aquesta informació d'encaminament caldrà tenir-la simultàniament en el node, s'utilitzen ontologies per a la seva organització i gestió. De quina manera s'intercanvien informacions ontològiques els nodes és un problema similar a l'encaminament de missatges. En aquest cas, podria pensar-se que la ontologia també pot incloure codi per a especificar de quina manera s'intercanviaran aquests tipus de missatges entre nodes, tal com passa en l'encaminament.

Vegeu també

Dediquem l'apartat 5 a presentar les qüestions relacionades amb la seguretat en els protocols tolerants a retards i a interrupcions.

5. Problemes de seguretat en protocols DTN

Els efectes de l'enfocament DTN sobre la seguretat són immediats. Així, perd el sentit qualsevol infraestructura centralitzada per a la gestió de les claus criptogràfiques, per la impossibilitat d'interactuar continuadament amb el centre de la infraestructura. De manera relacionada, qualsevol protocol clàssic basat en una negociació de diversos passos deixa de poder utilitzar-se en entorns DTN.

Lectura recomanada

Com el protocol TLS, definit en una RFC:
Dierks, T.; Rescorla, E. (2008). "The transport layer security (TLS) protocol version 1.2". *RFC 5246*.

Indubtablement, la seguretat es converteix en un dels problemes oberts més importants per resoldre en aquest enfocament (3; 16). Addicionalment, atès que cada entorn de connectivitat intermitent té les seves pròpies especificitats, les solucions de seguretat hauran de ser també específiques per a cada entorn considerat. Així, no és el mateix considerar la seguretat de les comunicacions interestel·lars que la d'una aplicació de correu en una zona rural sense infraestructura de comunicació.

Les úniques solucions de seguretat generals publicades fins ara són els mecanismes interns als diferents protocols DTN estandarditzats, BP i LTP, per a assegurar la privacitat i autenticitat de les dades enviades. Aquests mecanismes són similars als usats per IPSec, si bé amb claus compartides prèvies de llarga durada, i no tracten el problema inicial de la gestió de claus.

Els estàndards de seguretat per als protocols LTP i BP són els RFC 5327 i 6257, respectivament.

Descriurem tot seguit les solucions parcials plantejades per als principals problemes de seguretat dels entorns DTN; d'una banda, la gestió de les claus criptogràfiques, i d'altra banda, la seguretat relativa al nou model d'encaminament DTN.

5.1. Gestió de claus

Els mecanismes tradicionals basats en infraestructures de clau pública (*Public Key Infraestructures*, PKI) clarament no són vàlids perquè obliguen cada participant a contactar amb un centre de confiança diverses vegades, inicialment per a obtenir la signatura d'una autoritat de certificació sobre la seva identitat i la seva clau pública (certificat digital), i posteriorment per a obtenir les claus públiques d'altres participants i/o verificar la validesa dels certificats que són presentats per altres participants. En un entorn DTN no es pot garantir cap fita temporal per a aquests contactes amb el centre de confiança.

Donada aquesta situació, els únics esquemes de gestió de claus immediatament aplicables són equivalents a esquemes de compartició de secrets o bé esquemes de clau pública irrevocable, simples però poc segurs. Algunes propostes de solució utilitzen adaptacions d'esquemes basats en criptografia basada en la identitat* que solucionen parcialment el problema. Altres propostes inclouen l'ús d'esquemes descentralitzats, com SPKI/SDSI**.

* En anglès, *Identity Based Cryptography* (IBC).
** De l'anglès, *Simple Public Key Infrastructure / Simple Distributed Security Infrastructure.*

5.1.1. Solucions basades en IBC

En una infraestructura basada en IBC la clau pública d'un usuari s'obté de manera directa a partir de la identitat del mateix usuari, que pot ser una simple cadena de text. En aquestes arquitectures les claus privades les genera una tercera entitat de confiança, anomenada generador de claus privades*. Inicialment el PKG difon una clau pública mestra i guarda la clau privada mestra corresponent. La clau pública de qualsevol usuari pot ser fàcilment calculada mitjançant la combinació de la clau pública mestra i de la identitat d'aquest usuari. La clau privada corresponent ha de ser sol·licitada per l'usuari al PKG, que la calcula combinant la clau privada mestra amb la seva identitat (vegeu la figura 16).

* En anglès, *Private Key Generator* (PKG)

Figura 16. Criptografia basada en la identitat

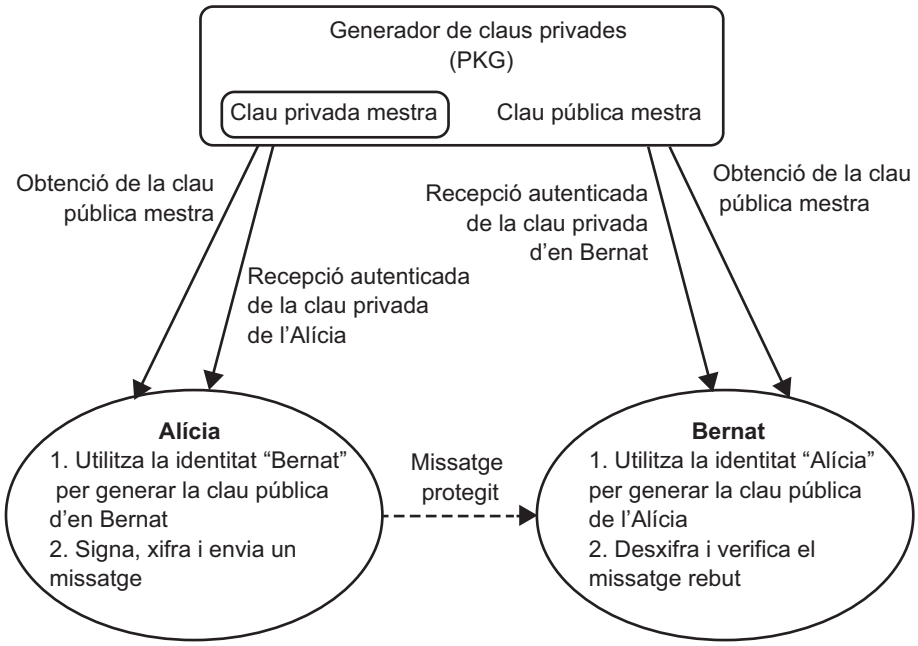


Figura 16

Relació entre els diferents participants en una comunicació segura per criptografia basada en la identitat. L'Alícia i el Bernat poden enviar-se directament missatges xifrats i signats amb una interacció inicial, que pot ser fora de línia, amb el PKG.

Addicionalment, cal notar que el PKG té la capacitat de generar la clau privada de qualsevol usuari i que, per tant, aquest mòdul ha de ser de total confiança.

Veiem que no fa falta una distribució de claus públiques entre els usuaris per tal que aquests puguin xifrar missatges i verificar signatures, i per tant que IBC és adequada *a priori* per a entorns DTN, sempre que es trobi una solució al

problema de la distribució prèvia de la clau pública mestra i de l'accés de cada usuari al PKG per a obtenir la seva clau privada.

Per a resoldre aquesta distribució prèvia algunes propostes inclouen l'utilització d'HIBC*. HIBC ofereix una arquitectura distribuïda, amb una jerarquia de PKG. D'aquesta manera, fins i tot en el cas en que no tots els nodes de la xarxa estiguin interconnectats, encara és possible adquirir una clau d'alguns dels PKG distribuïts.

De l'anglès, *Hierarchical Identity Based Cryptography.*

En HIBC s'usa un PKG arrel i es creen diverses subregions, cadascuna amb el seu propi PKG local. El PKG arrel estableix inicialment una sèrie de paràmetres per a tota la xarxa. A partir d'aquest moment el PKG arrel només necessita generar claus privades per als PKG de nivell inferior i aquests PKG de subregió generen claus privades per als usuaris. Els nodes que actuen com a usuari són afegits a una regió i això es converteix en part de la seva identitat. A més, per tenir en compte la possibilitat que hi hagi nodes en una regió completament desconnectada, sense accés directe al seu PKG local, altres propostes introdueixen una nova entitat física, l'operador de quiosc, responsable de la validació de la identitat dels usuaris i la distribució de dispositius USB que continguin un conjunt de credencials. Aquestes credencials es fan servir després per a obtenir les claus IBC utilitzant com a mitjà la mateixa xarxa DTN.

Altres aproximacions decideixen prescindir completament del servei centralitzat PKG. En aquests plantejaments es defensa que l'establiment de relacions de confiança es pot fer utilitzant material obtingut d'una capa de xarxa inferior, en els moments en què aquesta xarxa estigui disponible. Un exemple seria un identificador de targeta SIM (en xarxes de telèfons mòbils).

Aquests darrers plantejaments assumeixen un model de comunicació organitzat en diferents fases, com a mínim una fase inicial curta de configuració (*setup* o *bootstraping*) amb connectivitat a una infraestructura de base, i una fase llarga de funcionament desconnectat, on l'enfocament DTN prevaleix.

5.1.2. Solucions basades en SPKI/SDSI

L'últim tipus d'enfocaments aplicats consisteix en l'ús d'esquemes descentralitzats com SPKI/SDSI. En aquestes arquitectures la gestió de claus es realitza per a tots els membres de la xarxa d'una manera distribuïda. Aquest tipus d'aproximacions ha estat utilitzat per exemple en el projecte Haggle descrit anteriorment. La solució donada a la distribució de claus es basa en certificats d'atributs que s'utilitzen tant en l'autenticació, per a validar nodes i demostrar que el titular disposa dels atributs especificats, com per a confidencialitat, ja que s'usen per a emmagatzemar la clau pública que s'utilitzarà per a establir connexions segures.

Per a la distribució de certificats s'utilitzen una sèrie de nodes repartits per la xarxa que actuen com a emissors. Tots els nodes de la xarxa coneixen *a priori*

la seva clau pública i poden obtenir certificats de manera segura quan entren en contacte amb algun d'aquests nodes de distribució.

5.1.3. Revocació de claus

La revocació de claus planteja també problemes perquè la distribució de les tradicionals llistes de revocació* no és fàcil en entorns DTN. La tendència majoritària preveu la utilització de claus criptogràfiques de curta durada (fins i tot de poques hores, en funció de l'aplicació) per a fer innecessàries les CRL. En el cas d'usar criptografia basada en la identitat, les claus passen a dependre, a més de l'identificador de cada usuari, d'una marca de temps. Les claus compromeses no es revoquen, simplement caduquen ràpidament i no són renovades.

*** En anglès, *Certificate Revocation List* (CRL)**

5.2. Encaminament

L'algorisme d'encaminament en una xarxa DTN serà determinant en les propietats de seguretat del sistema. Algunes estratègies d'encaminament poden ser dissenyades sense mecanismes de protecció de la confidencialitat ni d'autenticació (5), però d'altres poden necessitar-los per al seu funcionament correcte. La utilització o no d'autenticació i protecció de la confidencialitat en el procés d'encaminament no treu que siguin necessàries també d'extrem a extrem, és a dir, entre aplicacions.

Tal com suggereix Farrell (3), és necessari que els protocols i les implementacions suportin mecanismes d'encaminament basats en polítiques. És a dir, cada protocol DTN hauria d'especificar quines variables de seguretat han de ser considerades des del punt de vista de l'encaminament, de manera que les implementacions puguin prendre decisions sobre l'enviament i reenviament dels missatges. Això és d'especial rellevància si considerem que el reenviament o enmagatzematge d'un missatge suposa una despesa de recursos. En el cas DTN, el reenviament de missatges tindrà sempre un ús de recursos associat, com ara l'espai de memòria o disc necessari, l'energia consumida, etc., per la qual cosa s'hauria d'incoporar una política d'encaminamet que permetés prendre decisions considerant aquestes variables. Per exemple, un node intermedi podria exigir que tot els missatges d'entrada haguessin d'estar autenticats, i els missatges que no complissin serien rebutjats.

Fins ara hem vist que, tot i disposar de diversos mecanismes per al transport de la informació i el seu encaminamet bàsic en xarxes DTN, encara no és possible aprofitar al màxim les seves possibilitats. El motiu és doble: d'una banda, necessitem que els missatges es dirigeixin cap als nodes que ofereixin una probabilitat més alta que el missatge arribi a la destinació amb les restriccions imposades per la mateixa xarxa i per l'aplicació; d'altra banda, són necessaris esquemes de seguretat que siguin tolerants a la no-contemporaneïtat de les

connexions entre nodes, característica en DTN. Per les dues coses encara no hi ha un estàndard consensuat, però hi ha força propostes per a arribar-hi. En apartats anteriors hem vist algunes alternatives per a fer l'encaminament, entre les quals es destaquen les arquitectures basades en missatges actius que ofereixen una flexibilitat molt gran per a aquest tipus de xarxes.

Pel que fa a la seguretat, es poden aplicar les tècniques habituals de xifrat i autenticació, tenint en compte les consideracions particulars en aquest tipus de xarxes ja vistes a l'inici d'aquest apartat. Hi ha una altra qüestió molt important en les xarxes DTN: la responsabilitat que adquireix un node quan accepta transportar un missatge, és a dir, ser el seu custodi. Això és nou en el model *store-carry-and-forward*, ja que està associat al concepte de *carry*. Des del punt de vista de la seguretat podem considerar esquemes de no-repudi, que obtinguin rebuts criptogràfics que permetin reconstruir traces de missatges *a posteriori*, per exemple, i detectar i actuar contra els nodes que no hagin complert els seus compromisos.

5.2.1. No-repudi

La primera dificultat que trobem quan intentem aplicar un esquema de no-repudi en DTN és que no ens podem basar en cap proposta que utilitzi una tercera part de confiança per la mateixa naturalesa d'aquest tipus de xarxa. D'altra banda, en un protocol de transmissió no repudiable de missatges s'ha de garantir que tant el missatge com el justificant de la seva recepció arriben a les parts corresponents de manera simultània. Hem d'aconseguir, doncs, un protocol que asseguri aquesta concurrència i que no utilitzi una tercera part de confiança.

En criptografia, el problema de l'intercanvi just de signatures, en què les signatures d'emissor i receptor queden vinculades al mateix temps, ja ha estat estudiat prèviament. En aquest tipus d'esquemes, ni emissor ni receptor poden retractar-se de la comunicació, i d'aquesta manera proporcionen no-repudi.

Susilo i altres (22) introdueixen el concepte de les signatures concurrents perfectes, en què l'emissor i el receptor poden produir signatures ambigües que no són vinculants fins que un dels dos allibera una peça extra d'informació. Directament, aquests esquemes no es poden utilitzar per a l'intercanvi de missatges en DTN per la necessitat d'accés a una tercera part de confiança, però admeten variacions utilitzant criptografia basada en identitat, per exemple*, en què les claus públiques són un tret identificatiu dels nodes i, per tant, conegut. Chow i Susilo (9), per exemple, proposen un esquema eficient per a l'intercanvi just de claus amb signatures concurrents perfectes utilitzant IBC.

* Vegeu el subapartat 5.1.1.

Per tal d'adaptar aquests esquemes al problema del no-repudi en l'encaminament en xarxes DTN s'han de fer algunes modificacions. Martínez-Bea i altres (15) proposen una variant que utilitza el mateix missatge per ser enviat com

a peça final d'informació que permet validar les signatures de l'esquema, i per tant els justificants de recepció/enviament. Vegem amb una mica més de detall com funcionaria l'esquema. Si el node Alice vol enviar un missatge al node Bob perquè el retransmeti necessitem, abans de tot:

1) Triar dos nombres primers grans p i q tals que $q \mid p-1$.

2) Triar un g d'ordre q tal que $g \in \mathbb{Z}_p^*$.

3) Tant per a Alice (A) com per a Bob (B), generar el parell de claus (sk_i,pk_i) tal que $pk_i = g^{-sk_i} \ mod \ p$, on sk_i és la clau privada i pk_i la clau pública. Així Alice, per exemple, tindrà la parella *skA* (pública) i *pkA* (privada).

4) Tant Alice com Bob han de tenir preparades les seves claus IBC, *ipk* clau pública, i *isk* clau privada.

Un cop tenim això, tant Alice com Bob poden preparar les seves respectives signatures i enviar-les:

1) $k = H(missatge \ || \ Sal)$, on $H()$ és una funció resum i Sal serà un nombre aleatori.

2) Escollir un w tal que $w \in \mathbb{Z}_p^*$

3) Calcular $m = \langle E_{ipk}(pk),S_{isk}(H(pk))\rangle$, on *ipk* és la clau pública IBC del receptor, *isk* la privada IBC de l'emissor, *pk* la clau pública de l'emissor, i $E()$ i $S()$ funcions de xifrar i signar respectivament.

4) Calcular $r = g^w \ mod \ p$.

5) Calcular $e = H(m,k,r)$ on H és una funció resum.

6) Calcular $c = w + esk \ mod \ q$, on *sk* és la clau privada de firma.

7) Construir signatura $\sigma = \langle m,k,s\rangle$, on $s = \langle r,e,c\rangle$

Finalment, un cop intercanviats σ_A i σ_B, Alice envia el missatge i la sal, de manera que ja quedarà tot preparat per a verificar les signatures. Alice haurà aconseguit enviar el missatge a Bob, i tots dos tindran un justificant que podrà demostrar davant un tercer l'acció que van fer (enviar/rebre). La figura 17 mostra el funcionament del protocol en les seves tres passes.

Figura 17. Protocol criptogràfic per a l'encaminament no repudiable

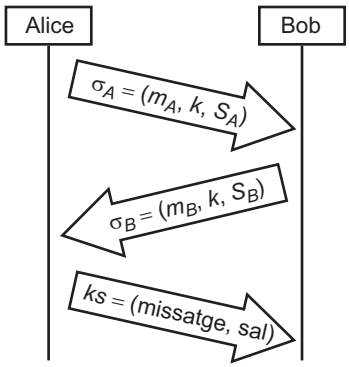


Figura 17

Mitjançant un protocol d'intercanvi simultani de signatures és possible tenir uns justificants que impediran que cap de les parts es retracti de la comunicació.

5.3. Control d'accés en els nodes DTN

En les arquitectures DTN basades en missatges actius, on el codi d'encaminament acompanya el mateix missatge, cal accedir a informació emmagatzemada en el node per a poder prendre la decisió d'encaminament. Normalment, aquesta informació estarà estructurada en ontologies, que facilitaran al node les tasques per la seva gestió, actualització i utilització.

Vegeu també

Les arquitectures DTN basades en missatges actius s'estudien al subapartat 4.3.

En aquest punt es presenta un problema de seguretat important. La informació que ha de ser utilitzada des d'aplicacions concretes pot ser l'objectiu d'un atacant que vulgui alterar el funcionament d'una aplicació particular. Es fa precís, doncs, controlar l'accés a aquesta informació per a evitar que pugui ser consultada o modificada des d'un missatge no autoritzat. Aquest control d'accés pot estar basat en IBC per a garantitzar la confidencialitat i la integritat de la informació utilitzada durant l'encaminament, o es pot utilitzar un esquema més senzill basat en el mateix codi d'encaminament. Seria com un control d'accés biomètric per ADN, si considerem el codi d'encaminament l'ADN del missatge.

Vegeu també

L'IBC s'estudia al subapartat 5.1.1. d'aquest mòdul.

Dos missatges diferents que comparteixin el codi d'encaminament tindran accés a la mateixa informació. Això ha de ser així perquè els dos codis han de poder accedir a la mateixa informació d'encaminament.

Un exemple de control d'accés amb aquestes característiques el trobem a l'obra de Sánchez-Carmona i altres (20). En aquesta proposta la informació està xifrada en el node, i es té una llista de regles per al control d'accés. Quan un missatge sol·licita una certa informació durant l'encaminament, s'aplicaran dues funcions resum criptogràfic diferents sobre el codi que ha sol·licitat la informació. Aquests valors s'utilitzaran per a recuperar la clau simètrica que permet desxifrar la informació.

Vegem com funciona amb més detall. Les informacions d'encaminament d'un cert protocol estaran xifrades i indexades d'aquesta manera:

$$(j, E_{k_j}(I_j)) \qquad (3)$$

On el primer element de cada dupla serveix per a identificar les diferents I_j i poder realitzar una cerca ràpida d'una informació concreta, i el segon element, I_j, és la informació d'encaminament en qüestió, degudament xifrada perquè cap codi c_i (o un altre procés) sense autorització pugui accedir-hi.

Les regles de control d'accés són així:

$$(j,h'(c_i),E_{h(c_i)}(k_j)) \quad (4)$$

On:

- j és l'identificador de la informació I_j.
- E és un algorisme de xifratge amb clau simètrica.
- h y h' són dos algorismes de resum diferents.
- c_i fa referència al codi d'encaminamient del missatge i.
- k_j és la clau simètrica necessària per a xifrar i desxifrar la informació I_j sobre la qual es controla l'accés.

Ara només cal definir de quina manera pot accedir el codi c_i a la informació I_j en quest esquema:

1) Es calcula $h'(c_i)$ i es busca la regla de control d'accés corresponent. En aquest cas podria ser $(j,h'(c_i),E_{h(c_i)}(k_j))$.

2) Es calcula $h(c_i)$ per a desxifrar la clau d'accés k_j.

3) Amb k_j ja es pot desxifrar $E_{k_j}(I_j)$ per a obtenir I_j.

Aquest esquema permet que la informació d'encaminament estigui accessible només per a un cert codi, i que fins i tot en cas que un node resultés compromés, no seria possible accedir a aquesta informació (necessitarem un missatge amb el codi autoritzat). Fer un codi maliciós que intentés enverinar la informació d'encaminamet fent-se passar per codi lícit, caldria fer un doble atac de preimatge simultani a les dues funcions resum (una per a enganyar el sistema d'indexació de claus, i l'altra per al desxifratge de la informació). La probabilitat de tenir èxit en un doble atac d'aquestes característiques, tenint en compte que són funcions resum diferents, és molt baixa.

Resum

En aquest mòdul didàctic hem fet referència a mecanismes i protocols de seguretat per a xarxes formades per dispositius sense fils amb limitacions quant a autonomia i capacitat de càlcul, potencialment mòbils, o sense connectivitat permanent entre les aplicacions en comunicació.

Quant a les xarxes ad hoc, hem vist el problema d'establir una xarxa amb nodes desconeguts. En primer lloc, hem vist com funcionen els **mecanismes de descobriment segur de nodes veïns**. A continuació hem analitzat els **protocols d'encaminament** i hem descrit com funcionen les extensions de seguretat d'aquests protocols. Finalment, hem explicat quins són riscos que aquestes xarxes comporten per a la **privadesa** dels usuaris, i quins són els mecanismes que es poden implementar per tal de protegir-ne la identitat i localització.

Pel que fa a les xarxes de sensors, hem vist diferents **protocols de gestió de claus criptogràfiques** per tal de poder posar les bases d'una infraestructura de seguretat en les xarxes de sensors. També hem presentat solucions eficients basades en **agregació de dades** per tal d'autenticar la gran quantitat de dades que els sensors van acumulant de les seves lectures i que han d'enviar a una estació base.

Pel que fa al protocols tolerants a retards i a interrupcions, hem vist les diferents arquitectures proposades per a tractar les molt variades situacions de connectivitat intermitent que hi poden haver, i les solucions proposades per a afrontar els problemes de seguretat encara oberts que tenen aquests protocols: bàsicament la **gestió de les claus criptogràfiques** i l'**encaminament amb custòdia de la informació** en els nodes intermedis.

Activitats

1. Busca informació sobre els mecanismes de *Watchdog* i *Pathrater*. Com poden millorar la seguretat de les xarxes ad hoc?

2. El protocol d'encaminament DSR és vulnerable a atacs de *blackhole*. Explica per què el protocol és vulnerable a aquests atacs, si les extensions de seguretat Ariadne-MAC i Ariadne-TESLA poden prevenir-los i com poden fer-ho.

3. Dissenya una xarxa de tipus DTN per a recollir dades d'una sèrie d'estacions meteorològiques disperses en un entorn rural. Quins serien els nodes? Quina estratègia d'encaminament faries servir?

Exercicis d'autoavaluació

1. En una xarxa ad hoc que utilitza encaminament SAODV amb funcions resum sha-1, dos usuaris reben els missatges següents *RREQ*. Són correctes?

a) HopCount=4,
MaxHopCount=255,
Hash (en base64)="EQM3MrVYrwvM+hZoJanqKsrHDD4=",
TopHash (en base64) = "sXLlKsKjqrm5RDVDS14HeF+VXyE="

b) HopCount=5,
MaxHopCount=255,
Hash (en base64)="tzvBaHrNpAph0tKworJdoQ8nbHQ=",
TopHash (en base64)="sXLlKsKjqrm5RDVDS14HeF+VXyE="

2. Indica quina clau utilitzaries i quina operació faries per a enviar els missatges següents autenticats en una xarxa de sensors:

a) Un node envia un missatge a l'estació base.
b) L'estació base envia un missatge en difusió (*broadcast*) a tots els nodes de la xarxa.
c) Un node envia un missatge en difusió (*broadcast*) a tots els seus veïns.

3. Quina afirmació és **certa** sobre les xarxes DTN?

a) Per a l'encaminament de missatges podem utilitzar un protocol pensat per a xarxes ad hoc, perquè també són xarxes emergents.
b) Es poden utilitzar esquemes de seguretat clàssics basats en PKI per a dissenyar mecanismes d'encaminament segur.
c) L'estratègia d'encaminament ha de ser la mateixa per a tots els missatges en la mateixa xarxa.
d) En un futur, les DTN poden arribar a desplaçar els protocols actuals d'Internet.
e) Cap de les respostes anteriors.

Solucionari

1.

a) correcte
b) incorrecte

2.

a) El node utilitza la clau de node que comparteix amb l'estació base per a enviar-li un missatge i un MAC generat amb aquesta clau.
b) Una xarxa de sensors té una clau de xarxa que comparteixen l'estació base i tots els nodes. Tanmateix, aquesta clau no es pot utilitzar per a autenticar missatges, ja que un node que rebés un missatge amb un MAC calculat a partir d'aquesta clau no podria estar segur de qui ha generat el missatge; qualsevol node que tingui la clau podria haver-lo generat.
L'autenticació en difusió basada en claus simètriques es pot fer amb el protocol TESLA. Aquest protocol ha de distribuir l'element arrel d'una cadena resum a través d'un missatge autenticat. En el cas de fer una difusió global, l'element arrel de l'estació base es pot precarregar en cada sensor abans del desplegament de la xarxa.

c) L'autenticació de missatges difusió locals és similar al punt anterior, es pot utilitzar el protocol TESLA. En aquest cas, per a establir l'element arrel de la cadena resum s'utilitzaria un missatge autenticat unidestinació (*unicast*) entre el node i tots els seus veïns (vegeu la resposta del primer punt).

3. e)

Glossari

Atac DoS De l'anglès *Denial of Service*, atac de denegació del servei.

Atac MITM De l'anglès *Man-In-The-Middle*, atac de l'home a mig camí

Atac Sybil Atac en el qual un usuari pren diferents identitats per tenir més influència en la xarxa.

DTN De l'anglès *Delay- and Disruption-Tolerant Networking*, protocols tolerants a retards i a interrupcions.

IBC De l'anglès *Identity-Based Cryptography*, criptografia basada en la identitat.

MANET De l'anglès *Mobile Ad Hoc Network*, xarxes ad hoc mòbils.

Paquet RREP De l'anglès *Route Reply*, missatge de resposta del protocol de descobriment de rutes per a crear taules d'encaminament.

Paquet RREQ De l'anglès *Route Request*, missatge de sol·licitud de descobriment de rutes per a crear taules d'encaminament.

WMN De l'anglès *Wireless Mesh Networks*, xarxes amb topologia de malla.

WSN De l'anglès *Wireless Sensor Networks*, xarxes de sensors.

Bibliografia

Bibliografia bàsica

Çayirci, E.; Rong, C. (2009). *Security in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons. http://books.google.co.uk/books?id=3EvhTrocBZUC.

Cerf, V.; Burleigh, S.; Hooke, A.; Torgerson, L.; Durst, R.; Scott, K.; Fall, K.; Weiss, H. (2007). «RFC 4838, Delay-Tolerant Networking Architecture». *IRTF DTN Research Group*.

Farrell, S.; Cahill, V. (2006). *Delay- and Disruption-Tolerant Networking*. Norwood, MA, USA: Artech House, Inc. ISBN 1596930632.

Perkins, C. E. (2008). *Ad Hoc Networking*. (1a. ed.). Addison-Wesley Professional. ISBN 0321579070, 9780321579072.

Bibliografia complementària

Burgess, J.; Bissias, G. D.; Corner, M. D.; Levine, B. N. (2007). «Surviving Attacks on Disruption-Tolerant Networks Without Authentication». A: «MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing», (pàgs. 61–70). New York, NY, USA: ACM. ISBN 978-1-59593-684-4. doi: http://doi.acm.org/10.1145/1288107.1288116.

Buttyan, L.; Hubaux, J.-P. (2008). *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. Cambridge University Press.

Chaum, D. (1985). «Security without identification: transaction systems to make big brother obsolete». *Commun. ACM*, volum 28 (núm. 10, pàgs. 1030–1044). ISSN 0001-0782. doi: http://doi.acm.org/10.1145/4372.4373.

Chaum, D. L. (1981). «Untraceable electronic mail, return addresses, and digital pseudonyms». *Commun. ACM*, volum 24 (núm. 2, pàgs. 84–90). ISSN 0001-0782. doi: http://doi.acm.org/10.1145/358549.358563.

Chow, S.; Susilo, W. (2005). «Generic Construction of (Identity-Based) Perfect Concurrent Signatures». *Lecture Notes in Computer Science 3783: Information and Communications Security*, (pàgs. 194–206).

Clausen, T.; Jacquet, P. (2003). «Optimized Link State Routing Protocol (OLSR)». RFC 3626 (Experimental). http://www.ietf.org/rfc/rfc3626.txt.

Dierks, T.; Rescorla, E. (2008). «RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2». *IETF*.

Eschenauer, L.; Gligor, V. D. (2002). «A key-management scheme for distributed sensor networks». A: «Proceedings of the 9th ACM conference on Computer and communications security», CCS '02, (pàgs. 41–47). New York, NY, USA: ACM. ISBN 1-58113-612-9. doi: 10.1145/586110.586117. http://doi.acm.org/10.1145/586110.586117.

Farrell, S. (2007). «Lakes, Noise and DTN Protocols: SeNDT & DTN Transport». SeNDT Presentation at Nokia. http://down.dsg.cs.tcd.ie/sendt/SeNDT-LTP-T-20070619.ppt.

Johnson, D.; Hu, Y.; Maltz, D. (2007). «The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4». RFC 4728 (Experimental). http://www.ietf.org/rfc/rfc4728.txt.

Martínez-Bea, S.; Castillo-Pérez, S.; Robles, S.; Gozalbo-Baró, M. (2012). «Protocolo de No-repudio para Redes DTN Basado en Intercambio Justo de Firmas». A: «XII Reunión Española sobre Criptología y Seguridad de la Información», Mondragon Unibertsitatea.

Martínez-Vidal, R.; C. de Toro, M.; Martí, R.; Borrell, J. (2012). «Esquema de gestión de claves criptográficas tolerante a retrasos e interrupciones en entornos aeronáuticos». A: «XII Reunión Española sobre Criptología y Seguridad de la Información», Mondragon Unibertsitatea.

Martonosi, M. (2005). «The ZebraNet Wildlife Tracker». Princeton. http://www.princeton.edu/ mrm/zebranet.html.

Perkins, C.; Royer, E.; Das, S. (2003). «Ad hoc On-Demand Distance Vector (AODV) Routing». RFC 3561 (Experimental). http://tools.ietf.org/html/rfc3561.

Perrig, A.; Song, D.; Canetti, R.; Tygar, J. D.; Briscoe, B. (2005). «Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction». RFC 4082 (Informational). http://www.ietf.org/rfc/rfc4082.txt.

Sánchez-Carmona, A.; Borrego, C.; Robles, S.; Andújar, J. (2012). «Control de Acceso para Mensajes Pro-activos en Redes DTN». A: «XII Reunión Española sobre Criptología y Seguridad de la Información», Mondragon Unibertsitatea.

Shukla, S.; Bulusu, N.; Jha, S. (2004). «Cane-toad Monitoring in Kakadu National Park Using Wireless Sensor Networks». Proc. Network Research Workshop, as part of 18th APAN Meetings, Cairns, Australia. http://www.cse.unsw.edu.au/ sensar/publications/kakadu.pdf.

Susilo, W.; Mu, Y.; Zhang, F. (2004). «Perfect Concurrent Signature Schemes». *Lecture Notes in Computer Science 3269: Information and Communications Security*, (pàgs. 14–26).

Zhang, J.; Varadharajan, V. (2010). «Wireless sensor network key management survey and taxonomy». *Journal of Network and Computer Applications*, volum 33 (núm. 2, pàgs. 63–75). ISSN 1084-8045. doi: 10.1016/j.jnca.2009.10.001. http://www.sciencedirect.com/science/article/pii/S1084804509001313.