Universitatea Bucureşti

Facultatea de matematică şi informatică

Probleme de securitate şi administrarea cheilor în reţelele ad hoc mobile

(Security Issues and Key Management in MANETs)

LUCRARE DE DOCTORAT

REZUMAT

Coordonator:

Prof. Univ. Dr. Adrian ATANASIU

Doctorand:

Ahmad Khalifah Mahmoud ALOMARI

Bucureşti

2014

Cuprins:

List of Figures:

1. Introducere

Reţelele ad hoc mobile (MANET) sunt noi tipuri de reţele fără fir, caracterizate printr-o topologie dinamică, adică o topologie care se creează şi se intreţine singură. Arhitectura reţelelor ad hoc a apărut pentru prima dată acum câţva ani şi carcteristica sa principală este aceea că nu se bazează pe nici un fel de infrastructură fixă. Aceste caracteristici prezintă vulnerabilitţi in ceea ce priveşte securitatea şi creează dificultăţi în asigurarea serviciilor de securitate in reţelele ad hoc [11]. Au fost efectuate cercetări masive şi actualizate în încercarea de a dezvolta protocoale de securitate pentru reţelele ad hoc.

Pe parcursul studiilor noastre de doctorat ne-am concentrat atenţia pe cercetarea securităţii in reţelele ad hoc mobile, în principal pe două ramuri majore: protocolul de rutare şi autoritatea de certificare.

Rutarea este o funcţie de bază a reţelelor ad hoc, dar care poate fi abuzată, acest lucru provocând diverse tipuri de atacuri. În general, protocoalele de securitate sunt predispuse la atacuri din partea dispozitivelor maliţioase. Aceste protocoale sunt proiectate de obicei fără a se ţine cont de securitate şi sunt foarte vulnerabile în faţa dispozitivelor cu o conduită neadecvată. În reţelele ad hoc, acest lucru este şi mai frecvent deoarece ele sunt proiectate pentru a minimiza nivelul de surplus al resurselor şi pentru permiterea fiecărui dispozitiv să participe la procesul de rutare. Eficientizarea protocoalele de rutare creşte riscul nesecurizării protocolului şi permite unui singur dispozitiv să aibă un impact semnificativ în activitatea protocolului din cauza lipsei redundanţei.

Datorită lipsei unei administraţii auto-centralizate şi a unei infrastructuri pre-existente, în reţelele ad hoc au fost adoptate de obicei diverse autorităţi de certificare răspândite în reţea. Fiecare dintre ele deţine o parte actualizată din cheia secretă. Criptografia ce foloseşte curbe eliptice este o tehnică criptografică ce capătă avânt şi este foarte potrivită pentru dispozitivele mici, ca cele folosite în comunicaţiile fără fir. Marele avantaj al acestei tehnici în comparaţie cu cele anterioare este acela că necesită o cheie mult mai scurtă pentru acelaşi nivel de securitate.

O autoritate de certificare de încredere este trimisă de obicei în infrastructura de securitate pentru a valida autenticitate cheilor publice [9]. Autorităţii de certificare i se

solicită să autentifice fiecare cheie publică înainte ca dispozitivul să o distribuie părţilor destinate şi să emită acelui dispozitiv un certificat digital cu cheia publică ataşată, certificat semnat cu cheia privată a autorităţii de certificare. O infrastructură bazată pe cheie publică asistată de o autoritate de certificare pare cea mai viabilă soluţie pentru securizarea reţelelor ad hoc.

2. Vulnerabilităţile reţelelor ad hoc şi analiza securităţii

2.1 Vulnerabilităţile reţelelor ad hoc

Datorită faptului că reţelele ad hoc mobile sunt supuse mult mai multor vulnerabilităţi decât reţelele tradiţionale cu fir, securitatea este mult mai greu de menţinut.

2.1.1 Lipsa graniţelor sigure

Semnificaţia acestei vulnerabilităţi este evidentă: nu există nici o graniţă sigură în reţelele ad hoc mobile [25] care să poată fi comparată cu linia de apărare clară din reţelele tradiţionale cu fir. Această vulnerabilite işi are originea în însăşi natura reţelelor ad hoc mobile: libertatea de a se alătura, de a părăsi şi de a se mişca înăuntrul reţelei.

2.1.2 Ameninţări din partea dispozitivelor compromise din interiorul reţelei

În secţiunea anterioară am discutat despre faptul că nu există graniţe sigure clare in reţelele ad hoc, fapt ce poate duce la apariţia diverselor atacuri asupra conexiunii. Aceste atacuri pun accentul asupra conexiunii dintre dispozitive, conducând comportamente maliţioase [21] în încercarea de a distruge aceste legături. Totuşi, reţelele ad hoc pot suferi diverse feluri de ameninţări, unul dintre ele fiind dispozitivul compromis din interiorul reţelei. Acesta este preluat de atacatori ce folosesc metode neoneste pentru a-l discredita, după care este folosit pentru a efectua acţiuni maliţioase.

2.1.3 Lipsa facilităţii de administrare centralizată

Un avantaj al reţelelor ad hoc este lipsa unui server fix. Totuşi, acest lucru se poate transforma şi într-un dezavantaj, provocând nişte probleme de vulnerabilitate. Din acest motiv, în urmatoare secţiune discutăm despre lipsa administrării centralizate.

Traficul dintr-o reţea ad hoc de mari dimensiuni este dificil de controlat, de asmenea şi verificarea lipsei conduitelor maliţioase. Detectarea şi prevenirea atacurilor se poate dovedi a fi o sarcină istovitoare datorită lipsei unui dispozitiv de administrare centralizată. Defecţiuni minore, ca aruncarea pachetelor, întreruperea căilor sau subminarea transmisiei, apar frecvent in reţelele ad hoc.

2.1.4 Alimentarea restricţionată

Reţelele ad hoc mobile nu pot preveni mobilitatea utilizatorilor, de aceea resursele lor energetice sunt mult mai limitate decât ale reţelelor cu fir. Sursa principală de alimentare a dispozitivelor este bateria [21].

2.1.5 Scalabilitate

Ultima problemă majoră în ceea ce priveşte vulnerabilitatea reţelelor ad hoc este scalabilitatea [21]. În reţelele convenţionale scara este predefinită din design şi nu este constrânsa la schimbări importante atunci când este întrebuinţată.

2.2 Analiza securităţii

Structura reţelelor ad hoc le face foarte vulnerabile în faţa multor tipuri de atacuri cum ar fi: interceptarea pasivă, interferenţa activă, personificarea, blackhole (gaura neagră), manipularea datelor şi, unul dintre ccele mai importante tipuri de atac: refuzul serviciului. Detectarea terminalelor compromise dintr-o reţea ad hoc vastă este împiedicată de:

- terminalele mobile sunt mereu interpretate ca noduri;
- protocoalele implementate au caracter concurent;
- există o lipsă de infrastructură fixă şi un punct central de concentrare în locul în care sistemul de detecţie al intruziunii poate colecta date verificabile;
- nu se face nici o distincţie între un nod normal şi o anomalie dintr-o reţea fără fir;
- din cauză că reţelele ad hoc sunt un mediu deschis, toate nodurile pot accesa datele aflate in raza lor de comunicaţie.

2.2.1 Paradigma Ad Hoc

Pentru a explica acest model ne vom concentra asupra protocolului AODV (Ad hoc On demand Distance Vector). Însăşi ideea de a securiza acest protocol reprezintă o provocare pentru că, mai intâi, trebuie înţelese atributele şi mecanismele securităţii. Aceasta este văzută ca un amestec de procese, proceduri şi sisteme. Toate aceste componente asigură controlul accesului, confidenţialitatea, integritatea, autentificarea, disponibilitatea şi ne-repudierea[19].

- Confidenţialitatea se obţine prin securizarea datelor împotriva accesului nodurilor neautorizate;
- Autentificarea se foloseşte pentru mai multe lucruri: pentru a se asigura de identitatea nodului sursă şi a nodurilor vecine; pentru a împiedica accesul neautorizat al unui nod la date confidenţiale şi la resurse; pentru a preveni un nod neautorizat să intervină în funcţionarea reţelei;
- Integritatea este foarte importantă pentru că împiedică nodurile maliţioase să retrimită datele alterate de ele (proces denumit câteodată atac de reluare sau atac wormhole)
- În ceea ce priveşte repudierea [12][19]: nodul care trimite mesajul nu poate nega că a fost trimis de el.

2.2.2 Diverse atacuri în reţelele ad hoc

Primul pas pentru apărarea reţelelor ad hoc trebuie făcut împotriva atacurilor pasive. Unelte de folosit sunt: criptarea, semnătura digitală, autentificarea şi controlul accesului. O altă provocare o reprezintă protecţia împotriva atacurilor active, a intruziunii şi descreşterea numărului de noduri egoiste. Criptarea şi autentificarea sunt bazate pe criptografia simetrică şi asimetrică [19].

În cele ce urmează vom parcurge diverse tipuri de atacuri, ilustrând cum funcţionează.

Personificarea – (atac denumit şi escrocherie): atacatorul este capabil de a juca rolul unui nod nevinovat şi de a se alătura reţelei. În această situaţie, mai multe astfel de noduri se alătură reţelei; obţin controlul asupra reţelei şi efectuează comportamente maliţioase. Cheile de criptare şi datele de autentificare pot fi şi ele accesate de nodurile compromise. În unele cazuri, activitatea obişnuită de rutare poate fi alterată de un nod maliţios care inundă reţeaua cu pachete de rutare false sau schimbă datele de rutare.

Interceptarea – este un atac pasiv în care atacatorul trage cu ochiul asupra reţelei pentru a colecta date. În acest fel, în timp ce informaţia este rutată, atacatorul va obţine informaţii private despre topologie, rutele optime şi locaţii geografice din reţea. În acest caz, atacul este foarte greu de detectat, iar informaţii vitale, cum ar fi cheia publică şi cheia privată a nodurile, parola etc., pot fi compromise.

Atacul wormhole – este unul din cele mai sofisticate şi mai severe atacuri din reţelele ad hoc. Atacatorul conectează două părţi (aflate la o distanţa precizată) ale reţelei şi apoi trimite mesajul primit într-o parte a reţelei în cealaltă parte [17].

Refuzul serviciului (Denial of Service - DoS) – scopul acestui atac este să facă un anumit nod indisponibli pentru serviciu. Întraga activitate ar putea fi compromisă dacă cineva foloseşte acest tip de atac. DoS este rezultatul manipulării integrităţii, redundanţei şi disponibilităţii reţelei. În acest tip de atac adversarul se foloseste de resursele nodului sau de lăţimea de bandă prin inundarea reţelei cu informaţii lipsite de importanţă.

Atacul blackhole (gaura neagră) – atacatorul ademeneşte traficul în aşă fel încât compromite nodul şi formează o gaură neagră, punând adversarul în centru [16]. În acest tip de atac nodurile maliţioase îşi păcălesc vecinii să atragă toate pachetele de rutare către ele.

Atacul Sybil – un nod încearcă să aibă mai multe identităţi. Pentru a realiza aceasta, nodurile maliţioase trebuie să comploteze şi să îşi partajeze cheile secrete între ele. În această situaţie nodul maliţios obţine mai multe date despre reţea.

În reţelele ad hoc mobile nu există o topologie fixă [17]. În acest tip de reţea nodurile trebuie să se descopere între ele. În protocoalele de rutare ale MANET nodurile trebuie să-şi anunţe prezenţa nodurilor colege şi de asemenea trebuie să ştie de prezenţa următoarelor noduri vecine din reţea.

3. Administrarea cheilor în reţelele mobile ad hoc

Această secţiune a tezei se concentrează asupra descrierii detaliate a sistemelor peer-to-peer (P2P) de administrare a cheilor. Putem clasifica schemele existente după cum urmează:

1. Autoritatea de certificare parţial distribuită.
2. Autoritatea de certificare distribuită în totalitate.
3. Administrarea cheilor bazată pe identitate.
4. Administrarea cheilor bazată pe lanţuri de certificate.
5. Administrarea cheilor bazată pe clustere (grupuri).

Cea mai bună metodă de a asigura autentificarea, ne-repudierea şi integritatea este folosirea criptografiei cu cheie publică. Această metodă este dovedit superioară altor metode criptografice şi majoritatea subdiviziunilor enumerate mai sus o folosesc. În schimb, dacă se vrea folosirea schemelor cu cheie simetrică, este necesară folosirea unui canal pentru asigurarea confidenţialităţii şi integrităţii datelor. De asemenea, de cele mai multe ori, asigurarea confidenţialităţii se dovedeşte a fi o provocare datorită necesităţii unei autorităţi de încredere sau a unui canal lateral (de exemplu o interfaţă în infraroşu) pentru a o face disponibilă.

3.1 Autoritatea de certificare parţial distribuită

Această schemă este una din primele încercări de rezolvare a problemei administrării cheilor în reţelele ad hoc mobile. A fost publicată în „Securing Ad Hoc Networks” [26], iar autorii săi, Zhou şi Z. J. Haas, au propus un serviciu distribuit de administrare a cheilor publice pentru reţelele ad hoc asincroane. Acest sistem funcţionează prin permiterea unui set de noduri, care au fost încredinţate cu încrederea, să partajeze un secret. N noduri servere formeaza autoritatea de certificare distribuită (DCA). Totalitatea lor beneficiază de o pereche de chei public/privată K/k.

3.2 Modelele distribuite total

Abordarea autorităţii de certificare distribuită total a fost adusă în atenţia noastră, pentru prima dată, de Luo şi Lu în „ Robust Authentication Services for Ad hoc Networks” [18]. În momentul în care se alătură reţelei fiecare nod primeşte o componentă din secret folosind o schemă de distribuţie în sistem confidenţial (n, k)-threshold. De

asemenea foloseşte mecanisme de partajare secretă verificabile şi proactive pentru a evita compromiterea cheii de semnare a certificatelor şi a asigura reţeaua împotriva atacurilor DoS.

3.3 Trecere în revistă a unor protocoale de rutare ad hoc

Tipurile principale de protocoale de rutare ad hoc sunt:

- protocoale de rutare proactive: tabelele de rutare sunt actualizate prin trimiterea periodică de mesaje. Cele mai importante astfel de protocoale sunt: OLSR (Optimized Link State Routing protocol) şi DSDV (Destinatio-Sequenced Distance Vector);
- protocoalele de rutare reactive (la cerere): rutele sunt genereate doar când este necesar. Două din cele mai importante protocoale de acest tip sunt: DSR (Dynamic Source Routing protocol) şi AODV (Ad hoc On demand Distance Vectort protocol).

3.3.1 Protocoalele de rutare proactive

3.3.1.1 Destination-Sequenced Distance Vector (DSDV) – Vector distanţă cu destinaţie dinamică ordonată

Acest algoritm [23] este o modificare a algoritmului de rutare Bellman Ford cu câteva îmbunătăţiri. Principal obiectiv de proiectare de DSDV a fost de a dezvolta un protocol care păstrează simplitatea protocolului de rutare RIP [15].

Toate nodurile mobile menţin o tabelă de rutare care conţine diferite date necesare procesului de rutare: toate destinaţiile disponibile, câte noduri trebuie depăşite pentru a ajunge la destinaţie şi secvenţa de numere alocate de transmiţător. Ultima este necesară în diferenţierea rutelor vechi de cele noi, fapt ce împiedică formarea buclelor. Din când în când, nodurile îşi trimit tabelele de rutare către vecinii imediaţi. Tot acest mecanism asigură folosirea doar a unei singure căi către destinaţie.

3.3.2 Protocoalele de rutare reactive (On demand)

În acest tip de protocoale rutele sunt generate pe loc, la cererea nodurilor. Când o transmisie de la sursă la destinaţie ia naştere, un proces este lansat în cadrul reţelei, numit descoperirea rutei. Când o rută este desoperită procesul se încheie.

3.3.2.1 Ad hoc On demand Distance Vector (AODV) – Rutare cu vector distanţă la cerere

AODV este un protocol de rutare reactiv unicast pentru reţelele ad hoc mobile. În AODV doar datele de rutare ale căii active necesită întreţinere [22].

Acest protocol poate fi denumit ca sistem de achiziţie a rutelor doar la cerere. Nodurile sunt mai independente deoarece nu se bazează pe calea activă, nu reţin datele de rutare şi nici nu iau parte în schimburi periodice a tabelelor de rutare. O rută trebuie creată între două noduri doar la momentul iminentei comunicări. Ruta poate fi descoperită şi întreţinută doar dacă nodul doreşte să fie folosit ca nod intermediar pentru alte două noduri pentru ca pachetele să fie tansmise prin el.

Există tehnici multiple pentru a stabili conectivitatea când este necesar. Una dintre aceste tehnici este mesajul de tip „Hello”, care, în acest caz, este difuzat doar local, între nodurile interesate, nu în toată reţeaua. AODV are două faze: descoperirea rutei şi întreţinerea rutei.

4. Dezvoltarea protocoalelor de rutare în reţelele mobile ad hoc (MANETs)

4.1 Preliminarii

Au fost create multiple protocoale de rutare pentru reţelele ad hoc mobile în încercarea de a asigura autentificarea dintre noduri. În acest capitol propunem scheme de securitate care pot fi aplicate majorităţii protocoalelor de rutare. Pentru aplicarea schemelor noastre am ales protocolul AODV, datorită popularităţii sale şi a folosirii frecvente [22].

Teza noastră se concentrează pe autentificarea dintre noduri. Reţelele ad hoc mobile ce folosesc noduri autorizate devin din ce in ce mai disponibile uzului general şi de asemenea reprezintă un subiect de cercetare a cărui importanţă creşte de la an la an.

Ideile propuse de noi folosesc funcţiile hash, dar şi semnăturile digitale. Pentru o securitate puternică, schema SAODV (o extensie a protocolului AODV) include algoritmi ce folosesc semnăturile digitale şi lanţurile hash pentru a realiza securitatea diferitelor nivele. Semnătura digitală este ataşată fiecărui nod pentru a furniza integritatea şi autentificarea mesajelor ce privesc rutarea: mesajul de cerere a rutei (RREQ – route request), mesajul de răspuns (RREP – route reply) şi mesajul de eroare (RRER – route error). Toate nodurile vecine ce primesc pachetul verifică semnătura digitală. Lanţurile hash sunt folosite pentru a securiza mecanismul cuantificării hopurilor.

4.2 Îmbunătăţirea autentificării în protocolul AODV (EA-AODV) pentru a rezista atacurilor

Dispozitivele folosite pentru controlul accesului utilizează deseori primitive criptografice cu cheie publică sau primitive cu cheie simetrică ce necesită distribuţia cheii. În cercetarea noastră funcţia hash cu lacăt îndeplineşte sarcina unui simplu mecanism de acces al controlului bazat pe funcţia one-way hash. Următoarele scheme propuse se aplică pe protocolul de rutare la cerere şi le demonstrăm analizând operarea protocolului SAODV.

4.2.1 Aplicarea funcţiei hash cu în schema noastră de autentificare a protocolului AODV

Schema hash cu lacăt înzestrează fiecare nod cu o funcţie hash lock. În această schemă [1] se foloseşte atât criptografia asimetrică cât şi cea simetrică. Cu alte cuvinte, acestă schemă este potrivită pentru cheile secrete şi pentru perechea de chei public/privată.

Presupunem că fiecare nod are o cheie verificabilă (vk), distribuită de dealer în momentul iniţializării MANET şi poate calcula valoarea hash a vk; rezultatul hash este dorit ca metaID al nodului, de exemplu: nodul G calculează funcţia hash pentru cheia sa verificabilă, iar metaID-ul nodului G este H (vk_G). Nodul îşi depozitează metaID-ul care va fi distribuit de nodul însuşi la iniţializarea reţelei. De asemenea, depozitează toate cheile verificabile ale altor noduri, care au fost distribuite de arbitru la iniţializarea MANET sau au fost distribuite chiar de noduri.

Dacă nodul A doreşte să trimită un pachet nodului G mai întâi folosim următoarea schemă pentru a ne asigura că nodul G este nodul autentificat (nodul destinaţie). Putem rezuma această schemă în paşii următori (Figura 4.1):

- nodul A adresează nodului destinaţie (nodul G) întrebarea: cine eşti?;
- nodul G îşi trimite metaID-ul către nodul A în momentul când primeşte întrebarea;
- când nodul A primeşte metaID-ul nodului G calculează toate metaID-urile şi potriveşte rezultatul cu valoarea primită;
- nodul A trimite nodului G cheia sa verificabilă (vk_G) care este criptată cu cheia publică a nodului G ($E_{PK_G}(vk_a)$). În final, nodul destinaţie primeşte ultimul mesaj de la nodul sursă şi obţine valoarea cheii sale verificabile prin decriptarea mesajului cu cheie privată a nodului destinaţie.

Prin accest pas nodul A se asigură că nodul G aparţine reţelei şi de asemenea nodul G autentifică nodul A.

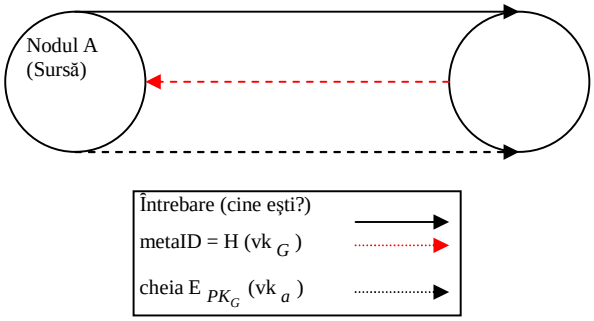

Figura 4.1: Schema hash cu lacăt

Deficienţa acestei scheme este faptul că orice nod al unei reţele poate fi urmărit de un atacator care poate inunda nodul cu întrebări. Nodul răspunde întrebărilor cu aceeaşi valoare (ca în Figura 4.2). În acest fel, atât cheia cât şi funcţia hash pot fi descoperite de atacator, făcând reţeaua vulnerabilă diverselor tipuri de atacuri ca personificarea, interceptarea, blackhole sau wormhole.

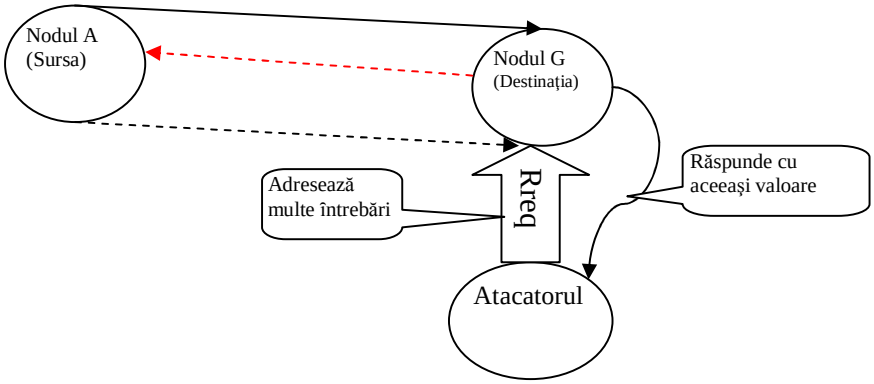

Figura 4.2: Atac asupra schemei de autentificare hash cu lacăt

O îmbunătăţire adusă acestei scheme ar fi folosirea generării numerelor aleatoare, metodă pe care o vom aplica în următoarea schemă, în încercarea de a împiedica un atacator să urmărească protocoalele de rutare.

4.2.2 Aplicarea numărului aleatoriu cu funcţia hash în schema noastră de autentificare a protocolului AODV

Abordarea noastră este bazată pe funcţia practică one-way hash [1]. De asemenea furnizăm o fundaţie mai puternică teoretic pentru funcţia pseudo-aleatoare (PRF).

După cum am declarat mai sus, în această schemă folosim şi generatorul de numere aleatoare pe lângă funcţia one-way hash utilizată în schema precedentă. Folosim această metodă pentru a îmbunătăţi comunicarea dintre noduri şi o explicăm cu exemplul:

Nodul sursă A doreşte să comunice cu nodul destinaţie G după cum este ilustrat în figura 4.3 [1] [3].

Pentru început nodul A adresează o întrebare simplă nodului destinaţie G. Când nodul G primeşte întrebarea generează un număr aleatoriu nonce R si calculează hash (IDg\\ R) prin concatenarea IDg cu R, apoi calculează funcţia hash pentru această valoare.

În final, nodul G îi răspunde nodului sursă A cu un mesaj conţinând atât nonce-ul precum şi rezultatul hash, adică perechea (enc $_{PK_A}$ (R), h (IDg\\ R)), unde PK_A este cheia publică a nodului A.

Când un nod A legitim primeşte perechea (R ,h(IDg\\R) execută o căutare puternică a tuturor identităţilor cunoscute prin calcularea hash a tuturor concatenată cu R până când găseşte o potrivire; nodul sursă A cunoaşte valoarea IDg.

Rezumăm aceşti paşi în figura 4.3:

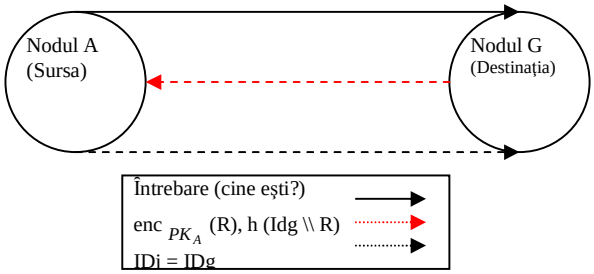

Figura 4.3: Schemă cu număr aleatoriu şi funcţie hash

- nodul A adresează întrebarea nodului G;
- nodul G generează un nonce aleatoriu R şi calculează (IDg\\R);
- nodul G (destinaţia) trimite (enc $_{PK_A}$ (R), hash(IDg\\R)) către nodul A (sursa);

- nodul A calculează hash(IDj\\R) pentru toate valorile identităţilor cunoscute de el;
- dacă nodul A găseşte o potrivire în aşa fel încât hash(IDj\\R) = hash(IDg\\R), atunci nodul A trimite IDj nodului G după ce îl criptează cu cheia publică a nodului G (E_{PK_G} (IDj)), unde PK_G este cheia publică a nodului G;
- nodul G se deblochează dacă primeşte IDj = Idg.

La final, autentificarea este efectuată şi transmiterea datelor între noduri este securizată. Acum nodurile pot să asigure trăsături precum integritatea, autentificarea şi ne-repudierea. Atacatorii găsesc că e foarte dificil de urmărit nodurile din cauza schimbării permanente a numărului aleatoriu R în fiecare mesaj de cerere sau de răspuns. Prin urmare, atunci când nodurile vor să comunice între ele vor folosi funcţia hash, numărul aleatoriu, dar şi semnătura digitală în orice RREQ şi RREP.

4.2.3 Altă metodă de aplicare a numărului aleatoriu şi a funcţiei hash în schema noastră de autentificare a protocoalelor de rutare la cerere.

Această schemă [3] menţine caracteristicile celor precedente, dar de asemena propune o tehnică îmbunătăţită de a rezolva punctele slabe din ultima schemă [1]. Îmbunătăţirea este reprezentată de securizarea în ceea ce priveşte protecţia intimităţii, rezistenţa împotriva atacurilor contrafăcute şi obţinerea autentificării reciproce.

În această schemă folosim de asemenea generatorul de numere aleatorii [3] pe lângă funcţia one-way hash. Folosim această metodă pentru a îmbunătăţi comunicarea şi autentificarea nodurilor. Paşii acestei scheme sunt explicaţi în figura 4.4

Parametri folosiţi:

ID_s : identitatea nodului sursă;

ID_d : identitatea destinaţiei;

R_s : numărul aleatoriu generat de nodul sursă;

PK_s : cheia publică a nodului sursă;

PK_d : cheia publică a nodului destinaţie.

Pasul 1: Un număr aleatoriu R_s este generat de nodul sursă şi criptat cu cheia publică a destinaţiei ($E_{PK_D}(R_s)$). Apoi este trimis cu întrebarea şi cu identificatorul ID_s ataşat către nodul destinaţie în timpul fazei de descoperire a rutei.

Pasul 2: În momentul în care nodul destinaţie primeşte pachetul de la nodul sursă îl decriptează $E_{PK_D}(R_s)$ cu cheia sa privată, calculează $h(ID_s \oplus R_s) \oplus h(ID_d)$ şi îl trimite direct către nodul sursă dacă este în aria de acoperire sau prin noduri intermediare dacă nu este în aria de acoperire.

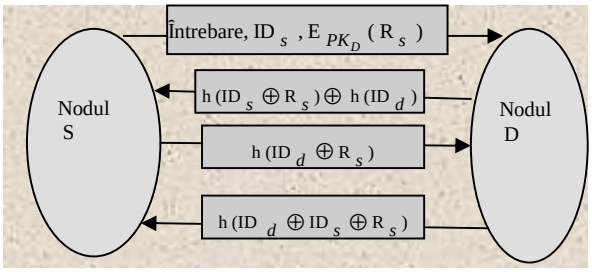

Figura 4.4: Schemă cu număr aleatoriu şi funcţie hash

Pasul 3: Când nodul sursă primeşte valoarea $h(ID_s \oplus R_s) \oplus h(ID_d)$ calculează funcţia hash a $ID_d \rightarrow h(ID_d)$ şi compară această valoare cu rezultatul dintre $h(ID_d)$ $=\{ h(ID_s \oplus R_s) \oplus h(ID_d)\} \oplus h(ID_s \oplus R_s)$. Dacă cele două valori sunt egale verificarea a reuşit. După aceea nodul sursă calculează $h(ID_d \oplus R_s)$ şi îl trimite către destinaţie.

Pasul 4: Nodul destinaţie primeşte $h(ID_d \oplus R_s)$ şi autentifică nodul sursă dacă verificarea acestei valori ţine. În final, nodul destinaţie calculează $h(ID_d \oplus ID_s \oplus R_s)$ şi îl trimite înapoi la sursă, care îl verifică la primire. Dacă procesul de verificare ţine, nodul sursă autentifică nodul destinaţie.

În această schemă folosim operaţia de disjuncţie exclusivă („sau exclusiv” - XOR $\oplus$) şi funcţia hash h ceea ce reduce semnificativ preţul calculului. Această schemă este securizată şi are capacitatea de a proteja intimitatea. Tot aici destinaţia şi sursa schimbă valorile $h(ID_s \oplus R_s) \oplus h(ID_d)$, $h(ID_d \oplus R_s)$ şi $h(ID_d \oplus ID_s \oplus R_s)$ la

fiecare proces de autentificare, ceea ce îngreunează sarcina adversarului de a descoperi aceste valori şi apoi de a descoperi nodurile sursă şi destinaţie prin mesaje de interceptare.

Apoi, comunicaţia dintre noduri se poate face mai uşor cu ajutorul funcţiei hash, al numărului aleatoriu şi al semnăturii digitale întrebuinţate în fiecare cerere de rută sau răspuns, de către nodurile care vor să trimită sau să primească pachete.

4.2.4 Folosirea numărului aleatoriu cu valoarea secretă şi amprenta de timp pentru protejarea intimităţii

Această schemă propune ca, pentru a securiza intimitatea nodurilor, să se utilizeze un număr aleatoriu cu o amprentă de timp (timestamp). Valori secrete sunt folosite pentru securizarea numărului aleatoriu [2].

În această schemă folosim în continuare aceeaşi parametri ca în precedentele două scheme, dar cu nişte completări:

- r: număr aleatoriu;
- Ts: amprentă de timp (timestamp);
- SVn: valoarea secreta n-th, unde n=1,2,3,....n,;
- λ: scăderea dintre numărul aleatoriu şi amprenta de timp;
- IDn: identităţile nodurilor din MANET;
- H(): funcţia securizată one-way hash;
- metaID: resultatul funcţiei hash pentru ID, h (ID).

Schema propusă constă în patru etape şi asigură securitatea prin numărul aleatoriu şi amprentă de timp. De asemenea, asigurarea anonimatului poate proteja intimitatea nodurilor. În această schemă mai folosim şi valoarea secretă (SV) care reprezintă un schimb securizat între oricare două noduri din reţea ca în schema de schimbare a cheilor Diffie-Hellman.

Explicăm aceste etape ca în figura 4.5:

Pasul 1: Un nod al unei reţele ad hoc doreşte să comunice cu alt nod al aceleiaşi reţele sau al unei reţele diferite. Figura 4.5 explică procesul de autentificare dintre

nodurile sursă (S) şi destinaţie (D) înainte ca acestea să înceapă să trimită şi să primească date importante. Nodul sursă generează un număr aleatoriu r, amprenta de timp Ts şi calculează XOR folosind valoarea secretă SV, care este partajată pentru a produce r ⊕ SV şi Ts ⊕ SV. Aceste valori sunt criptate cu cheia publică a nodului destinaţie (E_{PK_D} (r ⊕ SV), E_{PK_D} (Ts ⊕ SV)). Cele două valori sunt egalate pentru a forma valoarea hash h (r \\ Ts). Nodul sursă trimite E_{PK_D} (r ⊕ SV), E_{PK_D} (Ts ⊕ SV) şi h (r \\ Ts) către destinaţie.

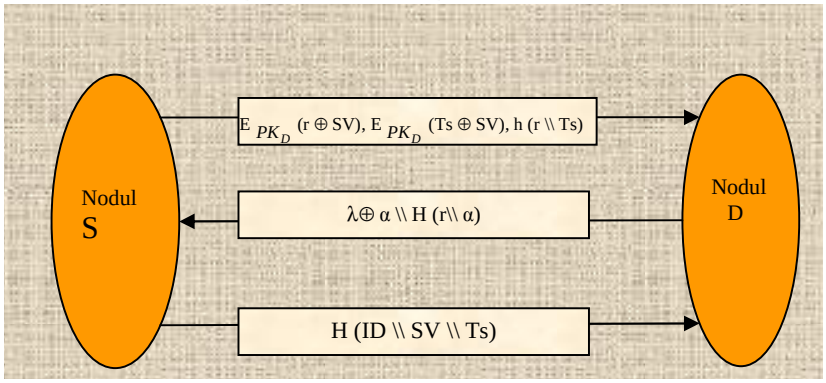

Figura 4.5: Autentificarea cu număr aleatoriu, valoare secretă şi amprentă de timp

Pasul 2: În această etapă se autentifică valoarea trimisă de nosul sursă. Destinaţia decriptează cu cheia sa privată valorile E_{PK_D} (r ⊕ SV), E_{PK_D} (Ts ⊕ SV), apoi utilizează poarta XOR pentru a introduce SV în r ⊕ SV şi Ts ⊕ SV pentru a obţine r şi Ts. Pentru a autentifica r şi Ts destinaţia foloseşte funcţia hash asupra celor două valori pentru a produce H (r\\Ts)'. Această valoare este egalată cu valoare provenită de la nodul sursă H (r\\Ts). Când cele două valori sunt identice r şi Ts sunt autentificate. Când destinaţia autentifică sursa începe să producă λ, unde λ este diferenţa dintre r şi Ts (λ = r – Ts), apoi calculează XOR metaID, unde metaID = α = h (ID), pentru a produce λ ⊕ α. Valoarea hash H (r\\ α) este generată şi potrivită cu λ ⊕ α pentru a forma λ ⊕ α \\ H (r\\ α), valoare ce va fi înaintată sursei pentru autentificare.

Pasul 3 : Când nodul sursă primeşte valoarea λ ⊕ α \\ H (r\\ α) de la destinaţie calculează λ din nou, scăzând r din Ts (λ = r – Ts). Apoi, se obţine valoarea metaID prin folosirea porţii XOR pe λ ⊕ α. Pentru a autentifica metaID-ul obţinut metaID (α) valoarea lui r transferat de la destinaţie este egalată pentru a forma valoare hash H (r \\ α)′. Dacă valoarea transferată H (r \\ α) este egală cu valoarea produsă H (r \\ α)′, valoarea

metaID (α) este autentificată împreună cu destinaţia căreia îi aparţine metaID-ul. De asemenea, sursa calculează metaID = H (ID)' pentru a se asigura ca ID-ul aparţine destinaţiei. Pentru a recunoaşte autentificarea destinaţiei nodul sursă combină SV cu variabila Ts pentru a forma valoarea hash al H (ID\\SV\\Ts), pe care o trimite destinaţiei.

Pasul 4: Când destinaţia primeşte ultima valoare de la sursă începe procesul de verificare prin autentificarea valorii hash H (ID\\SV\\Ts) ce a fost transferată de la sursă. Destinaţia combină SV cu Ts pentru a produce valoarea hash H (ID\\SV\\Ts). Sursa şi destinaţia se autentifică reciproc. Bineînţeles, valoarea hash este autentificată împreună cu ID-ul destinaţiei, care este o componentă a valorii hash.

4.2.5 Altă schemă ce foloseşte numărul aleatoriu şi valoarea secretă

În continuare, propunem o altă tehnică prin care folosim mai întâi amprenta de timp generată şi apoi numărul aleatoriu generat. Schema propusă foloseşte metoda autentificării provocare-răspuns, utilizând un identificator static şi funcţia one-way hash aleatorie. În plus, schema noastră [3] foloseşte amprentă de timp crescătoare pentru a face răspunsul mai uşor de identificat şi anonim. Explicăm această tehnică în figura 4.6 şi în paşii următori:

Pasul 1: Un nod al unei reţele ad hoc doreşte să comunice cu alt nod al aceleiaşi reţele sau al unei reţele diferite. Figura 4.6 explică procesul de autentificare dintre nodurile sursă (S) şi destinaţie (D) înainte ca acestea să înceapă să trimită şi să primească date importante. Nodul sursă generează o amprentă de timp Ts pe care o trimite cu întrebarea către nodul destinaţie în timpul fazei de descoperire a rutei. Valoarea amprentei de timp a sursei trebuie să fie mai mare decât valoarea amprentei de timp a destinaţiei (Td < Ts).

Pasul 2: Când destinaţia primeşte mesajul de la sursă şi se asigură că Td < Ts, generează un număr aleatoriu R_d pe care îl criptează cu cheia publică a sursei ($E_{PK_s}(R_d)$). Apoi, destinaţia calculează M_d =h (ID_d) $\oplus$ SV $\oplus$ h (SV \\ R_d \\ Ts) şi îl trimite sursei împreună cu $E_{PK_s}(R_d)$.

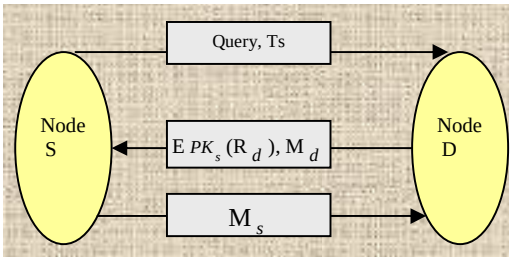

Figura 4.6: Autentificare cu amprentă de timp şi valoare secretă

Pasul 3: Sursa primeşte prin mesajul de cerere de rută valorile de la destinaţie pe aceeaşi cale, dacă aceasta este încă valabilă, sau pe altă cale de rutare dacă cea veche nu mai este disponibilă. Nodul sursă începe procesul de verificare prin calcularea valorii h (SV \\ R_d \\ Ts) şi a h (ID_d) = M_d ⊕ SV ⊕ h (SV \\ R_d \\ Ts). Apoi, sursa verifică dacă h (ID_d) aparţine într-adevăr destinaţiei.

Pasul 4: Când destinaţia primeşte mesajul de la sursă începe procesul de verificare prin calcularea valorii h (ID_d \\ SV \\ R_d \\ Ts)′. Dacă valoare calculată este egală cu valoarea primită, procesul de autentificare a reuşit şi amprenta de timp Td ← Ts este actualizată.

Schemele detaliate mai sus demonstrează îmbunătăţirile aduse pentru a securiza protocolul de rutare la cerere. Toate aceste îmbunătăţiri sunt bazate pe parametri aflaţi deja în uz, care vor fi explicaţi mai jos:

Fiecare nod ce primeşte un mesaj RREQ sau RREP poate verifica, datorită folosirii lanţurilor hash, ce controlează mereu integritatea câmpului de hopuri, dacă numărul de hopuri a fost modificat de un nod maliţios. Lanţurile hash constă în aplicarea continuă a funcţiei one-way hash pe un număr de seed-uri.

Când un nod reprezintă destinaţia sau are cunoştinţă de o cale către destinaţie trimite un mesaj RREP unicast înapoi către nodul sursă. Mesajul de răspuns conţine următorii parametri: adresa sursei, adresa destinaţiei, numărul secvenţial al destinaţiei, numărul de hopuri, durata de viaţă. Fiecare RREQ primit şi trimis mai departe de către un nod este depozitat într-un cache, permiţând destinaţiei sa trimită mesajul RREP unicast înapoi prin ruta memorată.

4.3 Autentificarea reciprocă şi actualizarea autentificării cheilor şi identităţilor

Propunem o schemă pentru a îmbunătăţi şi sprijini autentificarea reciprocă şi criptarea comunicaţiilor între noduri în reţelele ad hoc mobile [5]. Această schemă include căutarea cheilor, autentificarea reciprocă şi actualizarea k/ID, ce pot rezista majorităţii atacurilor, cum ar fi: urmărirea, personificarea, falsificarea şi interceptarea. Când două noduri doresc să comunice între ele şi sa facă schimb de date se pun de acord asupra unei chei secrete sau a unei sesiuni de chei. De fiecare dată, nodul care începe comunicaţia se numeşte nod sursă sau iniţiator, iar nodul care primeşte mesajul este destinaţia.

Următoarele variabile şi următorii operatori sunt folosiţi:

- E_k (M): rezultatul criptării tradiţionale a unui text sursă m cu cheia k;
- D_k (M): rezultatul decriptării tradiţionale a unui ciphertext (text criptat) m cu cheia k;
- H (M): funcţia one-way hash;
- $\oplus$: disjuncţia exclusivă;
- R_s : număr aleatoriu generat de sursă pentru verificarea destinaţiei;
- R_d : număr aleatoriu generat de destinaţie pentru verificarea sursei. De asemenea poate fi folosi şi pentru codificarea informaţiilor cheii după autentificare;
- $R_{s,d}$: număr aleatoriu iniţiat de sursă reprezentând noua cheie de autentificare pentru destinaţie;
- $K_{i,auth}$: actuala cheie de autentificare şi identitate a nodului destinaţie;
- $K_{i+1,auth}$, metaID_{i+1} : noua cheie de autentificare şi identitate a destinaţiei, iniţiată de sursă.

Sugerăm cheia de autentificare dintre sursă şi destinaţie şi o folosim pentru a genera metaID.

Explicăm schema de autentificare reciprocă în paşii următori [5]:

1. Sursa începe comunicarea, difuzează cererea de rută şi crează un număr aleatoriu R_s, criptat cu cheia publică a destinaţiei ($enc_{pk_D}(R_s)$, unde PK_D este cheia publică a destinaţiei, şi trimite mesajul către destinaţie împreună cu cererea de rută.
2. Destinaţia primeşte mesajul de la sursă, direct, dacă se afla în raza de comunicaţie sau prin noduri intermediare, dacă se află în afara ariei. Destinaţia generează $metaID_i$ ($metaID_i = h(K_{i,auth}, ID)$, apoi criptează R_s cu $K_{i,auth}$. În momentul în care primeşte R_s de la sursă, nodul destinaţie generează numărul aleatoriu R_d şi îl criptează cu $K_{i,auth}$ pentru a produce w_1 respectiv w_2 :

$$w_1 = enc_{k_{i,auth}}(R_s), \qquad w_2 = enc_{k_{i,auth}}(R_d)$$

după care destinaţia trimite către sursă $w_1 // w_2$ împreună cu $metaID_i$.
3. Când sursa primeşte $w_1 // w_2$ şi $metaID_i$ caută $K_{i,auth}$ corespunzător, cu $metaID_i$ ca index. Apoi, sursa verifică destinaţia: $w_1 \stackrel{?}{=} enc_{k_{i,auth}}(R_s)$. Dacă valorile sunt egale, procesul de comunicaţie şi autentificare continuă; dacă nu, procesul se întrerupe şi eşuează. Dacă destinaţia este validă, sursa decriptează w_2 cu $K_{i,auth}$ pentru a primi R_d de la destinaţie. În continuare, sursa criptează R_s cu R_d pentru a produce u_2, unde $u_2 = enc_{R_d}(R_s)$ şi trimite mesajul nodului destinaţie.
4. Când destinaţia primeşte de la sursă u_2, începe să verifice procesul de autentificare controlând dacă $u_2 \stackrel{?}{=} enc_{R_d}(R_s)$. Dacă valorile sunt egale, procesul de comunicaţie şi autentificare continuă; dacă nu, procesul se întrerupe şi eşuează. Dacă nodul sursă este valid, nodul destinaţie trimite o confirmare către sursă.
5. Când sursa recunoaşte că procesul de autentificare a reuşit generează un nou număr aleatoriu $R_{s,d}$ ca noua cheie de autentificare $K_{i+1,auth}$ pentru destinaţie şi calculează noul metaID corespunzător prin formula

$metaID_{i+1}$ = H ($K_{i+1,auth}$); numărul aleatoriu $R_{s,d}$ trebuie ales cu grijă pentru a fi sigur că este unic. Apoi sursa operează XOR pe noul $metaID_{i+1}$, $K_{i+1,auth}$, în pereche cu R_d şi le trimite către destinaţie α, β unde:

$$\alpha = metaID_{i+1} \oplus R_d \,,\ \beta = K_{i+1,auth} \oplus R_d$$

6. Destinaţia primeşte mesajul de la sursă cu noul $metaID_{i+1}$, $K_{i+1,auth}$, prin XOR α respectiv β cu R_d. Apoi destinaţia actualizează $metaID_i$ şi cheia de autentificare. La final, trimite o confirmare nodului sursă pentru a informa despre succesul actualizării.
7. Sursa primeşte confirmarea şi înlocuieşte vechea pereche ($metaID_i$, $K_{i,auth}$) cu noua pereche ($metaID_{i+1}$, $K_{i+1,auth}$).

Sursa şi destinaţia pot folosi cheia securizată R_d pentru criptarea datelor schimbate între ele după finalizarea reuşită a procesului de autentificare.

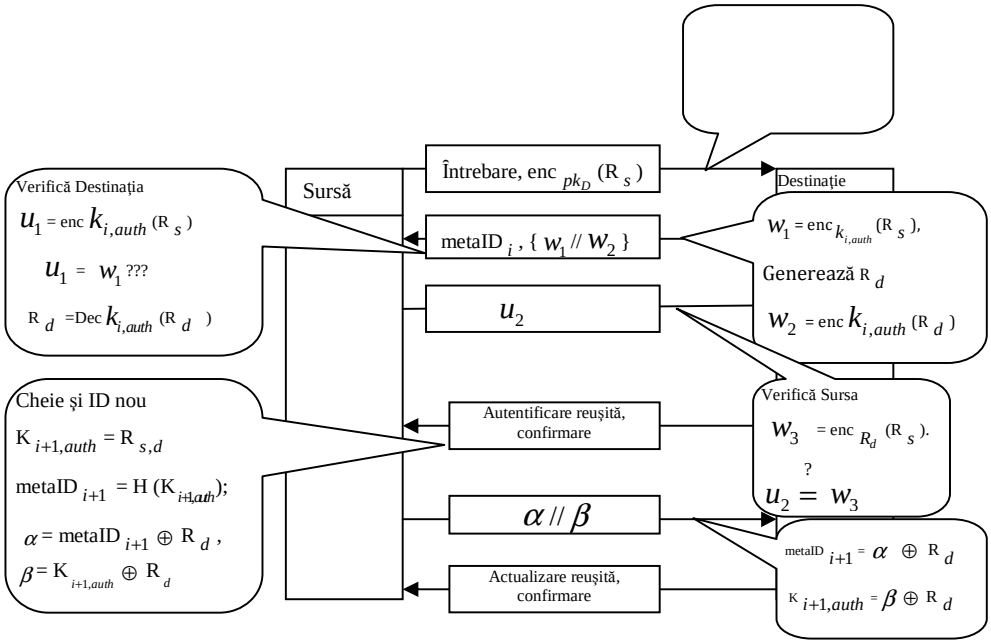

Figura 4.7: Procesul de autentificare provocare-răspuns şi actualizarea cheii de autentificare

Soluţia noastră depăşeşte domeniul securităţii şi asigură un serviciu de autentificare a nodurilor din reţelele ad hoc mobile mai bun. Ideile propuse folosesc

funcţia hash, amprenta de timp, valoarea secretă criptarea puzzle[4], cheia de autentificare şi autentificarea reciprocă dintre noduri, într-o manieră securizată.

5. Autoritatea de certificare distribuită total în reţelele ad hoc mobile

Acest capitol se concentrază pe administrarea peer-to-peer a cheilor în reţelele ad hoc mobile auto-organizate total. Un MANET auto-organizat total înseamnă că orice utilizator, cu echipament (şi software) adecvat, se poate alătura reţelei sau o poate părăsi când doreşte. Numim acest tip de reţea „reţea deschisă”; nu există nici o formă de control a accesului. Acest tip de reţea nu îşi va găsi aplicare în mediile militare ostile, ci, mai degrabă, în mediile comerciale, bazate pe comunităţi. Metodele actuale folosite în serviciile de autentificare depind de administrarea centralizată prin autorităţi de certificare (CA) sau prin centre de distribuţie a cheilor. O abordare centralizată poate fi acceptată în cazul în care un anumit nod poate fi protejat şi este accesibil altor noduri din reţea. Totuşi, pentru reţelele ad hoc fără fir pe care le dorim pentru aplicaţiile noastre ţintă, o abordare centralizată poate suferi din cauza refuzului într-un singur punct al serviciului şi poate fi inaccesibilă nodurilor ce au nevoie de servicii de cerificare. De aceea, o abordare mai puternică a autorităţii de certificare trebuie folosită. Reţelele ad hoc fără fir sunt, la acest moment, o arie de cercetare foarte activă. Succesul schemelor din accest capitol depinde de acel unic nod cu funcţie de autoritate de certificare. Datorită faptului că defecţiunea unui singur nod poate duce la oprirea întregului sistem, această abordare nu tolerează nici un fel de greşeală. De asemenea, această abordare este foarte vulnerabilă, deoarece unui adversar îi este de ajuns să compromită un singur nod pentru a obţine cheia secretă. Datorită imprevizibilităţii şi a mobilităţii preconizate a reţelelor ad hoc mobile, este posibil ca nodurile să nu poată fi capabile să acceseze autoritatea de certificare în timp util, făcând foarte greu de prezis când vor fi disponibile. Aşadar, o autoritate de certificare unică nu poate deservi eficient o întreagă reţea ad hoc.

5.1 Autoritatea de certificare distribuită total bazată pe polinom peste curbă eliptică în MANET

5.1.1 Preliminarii

Aici, schema noastră [6] este bazată pe un polinom peste curbă eliptică pentru reţelele ad hoc mobile. În această secţiune vom trece în revistă câteva preliminarii în legătură cu această tehnică: criptografia bazată pe curba eliptica (ECC) [20]. O curbă eliptică E peste un câmp limitat $\mathbb{F}_q$ constă în toate aceste puncte: (x; y)$\in \mathbb{F}_q \times \mathbb{F}_q$

$$y^2 = x^3 + ax + b \pmod q \qquad q > 3$$

$$4a^3 - 27b^2 \neq 0$$

împreună cu punctul la infinit: (E ($\mathbb{F}_q$)). Există o operaţie de adunare a punctelor al cărei element neutru este punctul la infinit (punctul ideal). Setul de puncte de sub această operaţie este denumit grup Abelian. Aşadar, un punct Q$\in$ E ($\mathbb{F}_q$) poate fi înmulţit cu o mărime scalară:

$$eQ = \underbrace{Q + \ldots + Q}_{e \text{ times}} = P$$

Problema inversă (de exemplu: se dă P şi Q, găsiţi e în aşa fel încât P=eQ), numită Problema logaritmilor discreţi peste o curbă eliptică (ECDLP), pare dificil de rezolvat din punct de vedere al calculului. Există mai multe sisteme cripografice a căror securitate este bazată pe dificultatea de rezolvare a problemei ECDLP. Prin comparaţie cu problema logaritmilor discreţi (DLP) pentru grupuri multiplicative, cea mai mare îngrijorare în ceea ce priveşte ECDLP este că, algoritmii sub-exponenţiali folosiţi ca indice de calcul în DLP pe grupuri multiplicative nu pot fi folosiţi pentru a rezolva ECDLP. Prin urmare, acesta se dovedeşte a fi o problemă mai complicată. Din punct de vedere practic, se pare că în ECDLP se pot folosi chei mai scurte, pentru a oferi aceeaşi securitate ca în DLP.

5.1.2 Autoritatea de certificare bazată pe polinom peste curbă eliptică

Considerăm o reţea ad hoc fără fir cu m noduri mobile. Comunicarea dintre noduri se face pe canale nesecurizate şi cu o lăţime de bandă limitată. Numărul de „m noduri” este dinamic şi este predispus schimbării datorită naturii mobile a reţelei, unde nodurile vin şi pleacă când doresc şi, de asemenea, se defectează în timp. De altfel, „m”

nu este limitat. Reţeaua nu furnizează nici infrastructură logică, nici ajutor fizic [13].

Presupunem că avem o autoritate de certificare (CA) şi m noduri participante într-o reţea ad hoc mobilă. CA-ul va distribui cheia secretă fiecărui nod participant din reţea. Cheia secretă SK_{CA} poate fi recuperată dacă şi numai dacă numărul de participanţi este mai mare sau egal cu t. CA-ul deţine o pereche de chei (PK_{CA} SK_{CA}), unde PK_{CA} este cheia publică cunoscută de toată lumea şi SK_{CA} este cheia privată cu confidenţialitate externă. În construcţia noastră folosim intens partajarea secretă cu polinom şi autoritatea de certificare distribuită total, bazate pe metodele descrise de Shamir [24] şi respectiv Luo şi Lu [18] pe care construim schema noastră de distribuţie totală peste curbă eliptică.

5.1.2.1 Iniţializarea schemei propuse

În această schemă am ales o curbă eliptică securizată E($\mathbb{F}_q$) peste câmpuri finite $\mathbb{F}_q$ (q este un număr prim):

$$y^2 = x^3 + ax + b \qquad q > 3$$

unde a,b$\in \mathbb{F}_q$ şi satisfac relaţia $4a^3 - 27b^2 \neq 0$. G este un punct peste curbă eliptică cu un număr prim mare de ordin n (n G = Ω) a cărui lungime binară este de cel puţin 160 de biţi. Arbitrul va alege un polinom de puterea r g(x) unde 1 < r < t, care ar putea fi factorizat peste $\mathbb{F}_q$ după cum urmează:

$$g(x) = g_1(x)\, g_2(x) \ldots\ldots g_k(x)$$

$g_i(x)$ este polinomul la puterea r_i care nu a putut fi descompus. Numărul de polinoame care este prim cu g(x) peste $\mathbb{F}_q$ este:

$$\phi_n(g(x)) = n^r \prod_{i=1}^{k} (1 - \frac{1}{n^r})$$

Arbitrul schimbă coordonatele lui G în polinom cu formula: G = $\langle h(x), h'(x) \rangle_{g(x)}$; G = $\langle h(x), h'(x) \rangle_{g(x)}$, $\langle h(x), h'(x) \rangle_{g(x)}$ înseamnă că ambele polinoame h(x), h′(x) vor opera modulo polinomul g(x). H (x) este o funţie one-way hash criptografică. Coaliţia de noduri care este răspunzătoare pentru CA publică parametrii publici (E, G, g(x), $\phi_n(g(x))$, h (x)).

5.1.2.2 Autoritatea de certificare distribuită total folosind polinom peste curbă eliptică

Schema propusă de noi [6] pentru autoritatea de certificare distribuită total este bazată pe o abordare a lui Luo şi Lu în „Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks" [18]. Aplicăm autoritatea de certificare a reţelei ad hoc mobile peste criptografia bazată pe curbă eliptică(ECC), unde fiecare nod are o pereche de chei: cheie publică PK_{CA}, cheie privată SK_{CA}, modulo g(x). Mai întâi arbitrul iniţializează un număr de t noduri şi aceste t noduri iniţializează restul reţelei. În schema distribuită total, cheia privată SK_{CA} este distribuită prin metoda partajării secrete a lui Shamir, prin fixarea lui SK_{CA} ca rădăcină a polinomului peste $E(\mathbb{F}_q)$. Arbitrul alege aleatoriu un punct R şi polinomul F(x) la puterea t – 1 peste curba eliptică $E(\mathbb{F}_q)$, le distribuie tuturor nodurilor din MANET şi, de asemenea, determină parametrii domeniului T = (p, a, b, G, N, h), care nu sunt ţinuţi secret:

$$F(x) = a_{\circ} + a_1 x + a_2 x^2 + \ldots\ldots\ldots + a_{t-1} x^{t-1}$$

unde $a_k \in [1, \phi_n(g(x))]$, (k = 0, 1, 2, …..,t-1) , $f(0) = a_{\circ} = SK_{CA}$.

Fiecare nod ce deţine o componentă din secret, cu o identitate unică diferită de zero, primeşte o componentă $s_i = f(i) \pmod n$, unde i = 1, 2… m, unde m este numărul de noduri din MANET.

Cunoscând cel puţin un număr t de componente, polinomul pate fi validat prin calculul:

$$F(x) = \sum_{i=1}^{t} s_i . l_i \ (x) \pmod n$$

Unde $l_i(x)$ este coeficientul Lagrange, definit ca:

$$l_i(x) = \prod_{j=1,\, j\neq i}^{t} \frac{x-i}{i-j}$$

f(x) este păstrat secret, apoi arbitrul efectuează următoarele:

- comandă r $p_i = (x_i, y_i) \pmod n$, unde $p_i = s_i G$ este calculat de nodurile participante în rețea;
- calculează $y_i = f(x_i) \pmod n$ (i= 1, 2… m)

$$R = r\,G \pmod n$$

Pentru CA, arbitrul publică punctul R şi parametrii y_i, în vederea verificării validităţii partajării secrete a cheii secrete (SK). Apoi, nodurile vor calcula şi publica parametrii $H_i = H(r\,p_i)$, (i= 1, 2… m).

Nodurile participante trebuie să efectueze două proceduri. Prima, fiecare nod al reţelei selectează cheia privată P_{riv} peste $\mathbb{Z}_q^*$ şi apoi calculeaza cheia publică $P_{Ki} = P_{riv}$. G mod p, P_{Ki} este cheia publică a nodului i pentru criptare şi verificare, P_{riv} este cheia privată pentru decriptare şi semnare. Nodurile folosesc aceste chei pentru criptarea şi decriptarea pachetelor de date. Un protocol provocare-răspuns poate fi folosit pentru a dovedi cunoaşterea cheii private P_{riv}, iar un certificat dovedeşte asocierea. A doua procedură, nodurile participante calculează $p_i = s_i G = \langle h(x), h'(x) \rangle_{g(x)}$, i = 1,2, ….,m, unde acest parametru este folosit pentru verificarea certificatului parţial când nodul i îşi semnează certificatul cu s_i.

A. Auto-iniţializarea

Reţelele fără fir folosite în aplicaţiile noastre ţintă sunt formate din noduri mobile. Acest lucru rezultă în continua alăturare şi plecare a nodurilor din reţea. După cum am spus anterior, arbitrul iniţializează un număr t de noduri, care, la rândul lor, iniţializează reţeaua. Când un nod nou intră în reţea şi nu are acces la un arbitru este nevoie de o metodă alternativă pentru ca acel nod să se alăture coaliţiei de noduri capabilă să asigure componente secrete. Această metodă este de asemeni necesară pentru a furniza nodului abilitatea de a genera dinamic componente secrete noi compatibile cu alte noduri existente în reţea.

B. Înnoirea certificatului

Certificatele au o dată de expirare; din această cauză reînnoirea lor este necesară. Când un nod p trebuie să-şi reînnoiască certificatul, o coaliţie de t noduri emite o reînnoire a certificatului la cererea acestuia. Certificatul propus spre reînnoire este verificat de fiecare nod al coaliţiei pentru a nu fi expirat sau revocat. Dacă a fost revocat, nodurile ignoră cererea. Fiecare nod al coaliţiei emite câte un certificat parţial ce conţine noua dată de expirare şi îl trimit înapoi nodului p.

C. Revocarea certificatului

Dacă un nod suspectează compromiterea cheii publice a altor noduri, poate revoca certificatele acelor noduri. De asemenea, îşi poate revoca propriul certificat dacă crede că a fost compromis. Este presupus că fiecare nod îşi ţine sub supraveghere vecinii şi păstrea o listă de revocare proprie.

5.1.3 Analiza Securităţii

Autoritatea de certificare a a cestei soluţii are nevoie e o infrastructură organizaţională/administrativă pentru a asigura operaţiile de înregistrare şi iniţializare. Avantajul major al acestei scheme este disponibilitatea sa şi faptul că foloseşte polinom pe curbă eliptică.

Securitatea schemei noastre [6] depinde de dificultatea de rezolvare a ECDLP. Comparăm algoritmii RSA şi ECC prin aceleaşi mărimi de chei în ceea ce priveşte efortul de calcul pentru criptanaliză. Comparat cu RSA, ECC necesită o mărime a cheii mult mai scurtă, acest lucru fiind mai profitabil din punct de vedere al calculului.

Serviciul de CA necesită prezenţa tuturor nodurilor din reţea. Pentru a fi disponibil, serviciu de CA răspunde doar cererilor venite de la nodurile cu vecini la un hop distanţă de t. Cantitatea traficului din interiorul întregii reţele este de asemenea restricţionată.

Protocoale de întreţinere elaborate (ca actualizarea componentelor sau iniţializarea componentelor) sunt folosite pentru a asigura disponibilitatea autorităţii de certificare. Multe componente sunt expuse pentru că, în această abordare, fiecare nod posedă componenta sa, în contrast cu abordările distribuite parţial, unde componentele sunt întreţinute doar de noduri specializate. Un adversar poate compromite o cantitate mare de componente la fiecare actualizare dacă parametrul t nu este selectat la cel mai înalt nivel. Dar cu cât t este mai mare, cu atât este mai mică disponibilitatea. Un dispozitiv de sincronizare trebuie asigurat, de asemenea, de către soluţie, pentru cazul în care reţeaua se fragmentează.

5.2 Autoritatea de certificare auto-organizată total în MANET

De obicei, sunt adoptate diverse autorităţi de certificare distribuite în reţea, fiecare cu o cheie secretă actualizată periodic. Într-o reţea ad hoc auto-organizată total toate

nodurile joacă rolul arbitrului. În această secţiune descriem o schemă eficientă de administrarea cheii publice, adecvată acestui tip de reţea. Metoda noastră afirmă că nodurile însăşi efectuează toate operaţiile reţelei, cum ar fi iniţializarea, distribuirea sau revocare cheii publice a nodurilor. Această abordare este o încercare de îmbunătăţire a procesului de construcţie a autorităţii de certificare auto-organizată total prin folosirea schemei Harn-Lin: schema verificabilă de partajare a secretului (n, t, n) puternică. Propunerea noastră va furniza reţelei ad hoc mobile o autoritate de certificare flexibilă şi eficientă.

5.2.1 Autoritatea de certificare auto-organizată total folosind schema de partajare secretă (n, t, n)

5.2.1.1 Iniţializarea schemei

În schema noastră [8] construim o autoritate de certificare auto-organizată distribuită total, bazată pe schema lui Lin şi Harn [14], care depinde de o partajare secretă (n, t, n). În majoritatea reţelelor ad hoc mobile arbitrii construiesc autoritatea de certificare când reţeaua este iniţializată. Dar, în schema noastră, prima coaliţie poate iniţia şi organiza reţeaua fără să se bazeze pe nici un arbitru [8]. Fiecare nod din MANET se va purta ca un arbitr pentru a genera componenta principală şi submulţimi ale componentelor pentru toate nodurile. Aşadar, fiecare nod va face aceleaşi operaţiuni ca toate celelalte.

Când un număr de n noduri dintr-o reţea ad hoc mobilă încep să iniţializeze autoritatea de certificare, fiecare nod va defini un secret $s \in \mathbb{Z}_p$ pe care îl distribuie printre ceilalţi pentru a construi cheia principală. Se iau *p* şi *q*, două numere prime mari în aşa fel încât *q/(p-1)* şi *g*, $h \in \mathbb{Z}_p$ sunt două elemente de ordin *q*. Putem rezuma aceşti paşi după cum urmează:

- fiecare nod va fi arbitru şi fiecare nod $D_i \in \{ D_1, D_2, \ldots\ldots, D_n \}$ va genera un polinom pentru submulţimile secretului $f_i(x) = a_{i,0} + a_{i,1} x + \ldots\ldots + a_{i,t-1} x^{t-1}$, de grad t-1, în care submulţimile secretului $a_{i,0} = f_i(0) = s_i$ şi toţi coeficienţii sunt în $\mathbb{Z}_p$.

- fiecare nod alege aleatoriu $b_{i,1}$, $b_{i,2}$, ……., $b_{i,t-1} \in \mathbb{Z}_p$ şi generează $k_j(x)$ în aşa fel încât:

$$k_j(x) = b_{j,1} + b_{j,2}\, x + \ldots\ldots + b_{j,t-1}\, x^{t-1}$$

- fiecare nod D_i calculează toate submulţimile componentei ($s_{i,j}$, $t_{i,j}$) şi angajamentul coeficienţilor faţă de $f_i(x)$ şi $k_j(x)$ după cum urmează:

$$s_{i,j} = f_i(j)\ ;\ t_{i,j} = k_i(j) \text{ pentru j = 0, 1, 2, ……, } t\text{-1}$$

şi calculează $c_{i,j} = g^{a_{i,j}}\ h^{i,j}$ (mod p)

- apoi, fiecare nod D_i distribuie submulţimile componentei ($s_{i,j}$, $t_{i,j}$) $\forall$ j = 0, 1, 2, ….., t-1 şi $i \neq j$ şi difuzează $c_{i,v}$ tuturor nodurilor din MANET.
- După ce D_i primeşte toate submulţimile componentei şi informaţia difuzată de la alte noduri, calculează componenta principală, unde:

$$S_i = s_{1,i} + s_{2,i} + \ldots\ldots\ldots + s_{n,i}$$

Şi

$$t_i = t_{1,i} + t_{2,i} + \ldots\ldots + t_{n,i}$$

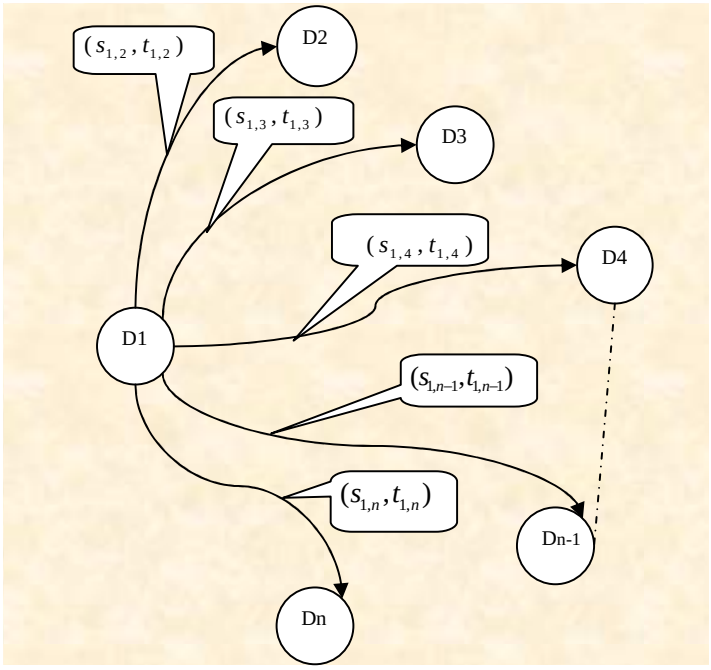

Figura 5.3: Nodul distribuie submulţimi

De exemplu: nodul D_i calculează

$$S_1 = s_{1,1} + s_{2,1} + \ldots\ldots\ldots + s_{n,1} \text{ (mod p)},$$

şi $$t_1 = t_{1,1} + t_{2,1} + \ldots\ldots + t_{n,1} \text{ (mod p)}.$$

Nodul (D_2) calculează

$$S_2 = s_{1,2} + s_{2,2} + \ldots\ldots\ldots + s_{n,2} \text{ (mod p)},$$

Şi $$t_2 = t_{1,2} + t_{2,2} + \ldots\ldots + t_{n,2} \text{ (mod p)}.$$

Şi nodul (D_n) calculează

$$S_n = s_{1,n} + s_{2,n} + \ldots\ldots\ldots + s_{n,n} \text{ (mod p)},$$

şi $$t_n = t_{1,n} + t_{2,n} + \ldots\ldots + t_{n,n} \text{ (mod p)}.$$

De asemeni D_i calculează $c_v = c_{1,v} . \ c_{2,v} \ldots\ldots c_{n,v}$ (mod p), pentru $v = 0, 1, 2, \ldots, t-1$.

Acum fiecare nod foloseşte schema de partajare a secretelor a lui Shamir pentru a găsi polinomul principal care are secretul principal:

$$S = s_{1,0} + \ s_{2,0} + \ldots\ldots + s_{n,0}$$

Verificarea componentei: fiecare nod D_i care a obţinut componenta principală (S_i, t_i) şi toate valorile angajamentului c_v pentru $v = 0, 1, 2, \ldots\ldots, t-1$ poate verifica dacă toate componentele principale S_j definesc într-adevăr un secret testând dacă:

$$g^{S_i} \ h^{t_i} = \prod_{v=1}^{t-1} c_v^{i^j} \ \text{(mod p)}$$

Reconstrucţia secretului: este la fel ca în schema lui Shamir.

5.2.1.2 Revocarea certificatului

Certificatul emis poate fi revocat de fiecare nod dacă acesta crede că certificatul nu deţine o cheie de utilizator validă. Nodul îşi poate revoca chiar şi cheia sa publică dacă crede că a fost compromisă.

În schema noastră [8] folosim două metode de revocare a certificatului: explicită şi implicită.

În schema de revocare explicită, un certificat eliberat poate fi revocat dacă nodul emite o declaraţie explicită de revocare. Declaraţia de revocare nu trebuie trimisă fiecărui

nod pentru că toate nodurile posedă o listă cu nodurile ce au nevoie de actualizarea certificatelor emise. Aşa că declaraţia este trimisă doar nodurilor care se actualizează regulat. După difuzare, revocarea certificatului ajunge şi la alte noduri, dar cu o întârziere a timpului de convergenţă în schimbul certificatului.

Metoda implicită de revocare a certificatului este bazată pe data de expirare al certificatului. Fiecare certificat include data la care a fost emis şi o perioadă de validitate (VP), care de obicei durează câteva zile. O operaţie importantă este desemnarea corectă a lungimii VP deoarece certificatul îşi pierde valabilitatea când timpul se termină.

În perioada de validitate a unui certificat este presupus că un nod poate stabili comunicaţia cu orice alt nod capabil să emită certificate. De asemenea, în timpul acestei perioade va avea loc un permanent schimb al actualizărilor certificatelor şi depozitele de certificate ale nodurilor vor fi actualizate. Dar, dacă o parte a certificatelor nu pot fi actualizate în depozitul local al nodului acestea pot fi recuperate cu ajutorul altor certificate actualizate disponibile.

5.2.1.3 Înnoirea certificatului

Certificatele au o dată de expirarea de aceea este necesară înnoirea lor. Când un nod D_i trebuie să-şi înnoiască certificatul, o coaliţie de *t* noduri vecine emite o înnoire la cererea certificatului. Fiecare nod al coaliţiei controlează ca certificatul să nu fie expirat sau revocat. Dacă a fost revocat, nodurile ignoră cererea. Dacă nu, cererea este admisă. Fiecare nod al coaliţiei emite un certificat parţial ce conţine o nouă dată de expirare şi îl trimite înapoi nodului D_i .

5.2.2 Analiza securităţii

În schema noastră [8] folosim autoritatea de certificare distribuită ce depinde de schema de partajare a secretului. Majoritatea modelelor autorităţii de certificare în MANET folosesc un arbitru ca să partajeze un secret între n participanţi (excluzând arbitrul). Felul în care secretul este împărţit face ca doar câţiva dintre participanţi să poată să îl reconstituie. Totuşi, este posibil ca participanţii să nu poată fi capabili să recupereze secretul dacă arbitrul sau alţi participanţi au o conduită maliţioasă. Acest fel de conduită poate fi împiedicată prin punerea în aplicare a unui protocol de securitate care permite ca

majoritatea destinatarilor componentelor să verifice majoritatea operaţiilor. Această abordare poate funcţiona dacă toţi participanţii (incluzând arbitrul) sunt oneşti.

În partajarea verificabilă a secretului (VSS) [14] obiectivul principal este de a rezista conduitei maliţioase a nodurilor. Acest comportament include transmiterea neonestă o componentelor către unul, câţiva sau toţi participanţii, care înaintează acele componente în timpul procesului de reconstrucţie. Utilizarea VSS necesită menţinerea canalelor private dintre nod şi fiecare participant individual, disponibile. Totuşi este clar că comunicarea pe canale private nu poate fi verificată public.

În schema noastră fiecare nod se poartă ca un arbitru şi doreşte să contribuie la crearea şi partajarea unui secret principal. Fiecare nod alege un secret aleatoriu, numit submulţime a secretului. Dacă este utilizat algoritmul lui Shamir de generare a componentei, submulţimea secretului poate fi partajată între noduri cu ajutorul submulţimilor componentei. Componenta principală poate fi creată prin combinarea submulţimilor fiecărui participant pentru ca, în final, secretul principal să poată fi reconstituit. Pentru a face asta este utilizat algoritmul lui Shamir de reconstruire a secretului, folosind t sau mai mult de t componente principale.

6. Concluzii finale şi lucrări viitoare

Cercetarea din această teză este concentrată pe securitatea reţelelor ad hoc mobile, în special pe două ramuri majore: protocolul de rutare şi autoritatea de certificare. Teza este împărţită în şase capitole: primele trei se ocupă de reţelele ad hoc în general şi de securitatea reţelelor ad hoc mobile în faţa majorităţii tipurilor de atacuri. Capitolele 4 şi 5 introduc schemele personale de îmbunătăţire a securităţii şi autentificării procesului de rutare şi modelele noastre de autorităţi de certificare pentru reţelele ad hoc mobile.

Pe parcursul acestei teze demonstrăm că securitatea reţelelor ad hoc mobile este un subiect de cercetare foarte bogat. Dar, mai este încă mult de muncă în acest domeniu, în special în ceea ce priveşte securitatea protocoalelor de rutare şi a autorităţii de certificare auto-organizate.

Rezumând contribuţia noastră, evidenţiem câteva probleme rămase deschise viitoarelor cercetări. O problemă comună primelor noastre şase scheme este autentificare reţelelor ad hoc mobile. În primele cinci scheme folosim funcţia hash şi numărul aleatoriu pentru a îmbunătăţi securitatea dintre nodurile sursă şi destinaţie, dar în a şasea schemă aplicăm funcţia puzzle, ca unealtă practică şi eficientă pentru a creşte securitatea protocolului de rutare la cerere în MANET. Schema a şaptea propune o tehnică de actualizare a cheii de autentificare comune pentru două noduri ce comunică între ele în MANET.

Ultimele două scheme aduc în discuţie altă problemă obişnuită, acea a eficientii sistemului de administrare a cheii. Toate abordările administrării cheii sunt supuse diverselor restricţii, cum ar fi disponibilitatea resurselor dispozitivelor mobile, lăţimea de bandă şi natura dinamică a reţelelor ad hoc mobile. Cercetarea administrării cheii merge în trei direcţii în concordanţă cu modelele de încredere, care sunt: centralizată, descentralizată şi distribuită total.

De asemenea subliniem nişte puncte ce pot fi explorate pe mai departe, cum ar fi tehnica de detecţie a intruşilor. Vom încerca să explorăm mai adânc în această zonă.

În final, am vrea să afirmăm că, în viitoarea generaţie a sistemelor de comunicaţie fără fir, va fi nevoie de desfăşurarea rapidă a dispozitivelor mobile independente. Scenariile reţelelor nu se pot baza pe conexiunea centralizată şi organizată, dar pot fi

concepute ca aplicaţii ale reţelelor ad hoc mobile. Deci, acest tip de reţea devine cea mai bună soluţie a diverselor probleme legate de reţele.

Referinţe

[1] Ahmad Alomari, Improvement Authentication of routing protocols for mobile ad hoc networks, International conference on advances in computer science and electronic engineering (ICACSEE-212), 2-3 Feb., 2012. International Journal of Advances in Computer Networks and Its Security – IJCNS, A Unit of UACEE Journals, Volume 2: Issue 1, 25 April, 2012.

[2] Ahmad Alomari, Security Authentication of AODV Protocols in MANETs. The 7th international conference on network and system security, NSS 2013,Lecture Notes in Computer Science (LNCS). Volume 7873, 2013, pp 621-627 (proceedings indexed ISI).

[3] Ahmad Alomari, Development in Authentication of AODV Protocols to Resist the Attacks **.**The 19th international conference on information and software technologies ICIST 2013, communications in computer and information science CCIS 403, pp. 334--344. Springer, Heidelberg (2013) (proceedings indexed ISI)

[4] Ahmad Alomari , Applying Puzzle Encryption In The On-Demand Routing Protocols In Mobile Ad Hoc Networks (Manets), Journal of Information System & Operations Management Vol. 6 No. 2 December 2012.

[5] Ahmad Alomari , Mutual Authentication and updating the Authentication Key in MANETS (submitted)

[6] Ahmad Alomari, Fully Distributed Certificate Authority Based on Polynomial over Elliptic Curve for MANET, 14th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2013) (proceedings index ISI)

[7] Ahmad Alomari, Implementing RSA algorithm fully distributed certificate authority in MANET, Journal of Information Systems & Operations Management; May2013, Vol. 7 Issue 1, p172.

[8] Ahmad Alomari , Fully self organized of Certificate Authority in MANETs by using (n, t, n) Secret Sharing Scheme (submitted)

[9] S. Capkun, J. Hubaux, and L. Buttyan, “Mobility Helps Peer-to-Peer Security,” IEEE Transactions on Mobile Computing, vol. 5, no. 1, pp. 43–51, 2006.

[10] S. Capkun, L. Buttyan, and J.-P. Hubaux, “Self-Organized Public-Key Management for Mobile Ad Hoc Networks,” IEEE Transactions on Mobile Computing, vol. 2, no. 1, pp. 52–64, 2003.

[11] M. Conti, Body, Personal and Local Ad Hoc Wireless Networks, in Book The Handbook of Ad Hoc Wireless Networks (Chapter 1), CRC Press LLC, 2003.

[12] A. R. Das, E. Charles, Perkins and E. M. Royer, “Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks”

[13] Y. Dong, A.-F. Sui, S. Yiu, V. O. Li, and L. C. Hui. Providing distributed certificate authority service in cluster-based mobile ad hoc networks. Elsevier, Computer Communications, May 2007.

[14] L. Harn and C. Lin, Strong (n,t,n) verifiable secret sharing scheme, Information Sciences 180 (2010) 3059–3064.

[15] C. Hedrick, Routing Information Protocol, RFC 1058, June 1988.

[16] S. Lee, B. Han, and M. Shin, "Robust Routing in Wireless Ad Hoc Networks," 2002 Int'l. Conf. Parallel Processing Wksps., Vancouver, Canada, Aug. 18–21, 2002.

[17] B. Lu and U.W.Pooch "Cooperative security-enforcement routing in mobile Ad Hoc networks" IEEE2002.

[18] H. Luo and S. Lu, Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks, Technical Report200030, UCLA Computer Science Department 2000[5] J-P. Hubaux, L. Buttyán and S. Capkun.

[19] C. R. Mala, S. Shetty, S. Padmashree., E. Elevarasi, "Wireless Ad hoc Mobile Networks", National Conference on Computing Communication and Technology, pp. 168-174, 2010.

[20] A. Mishra and M. K. Nadkarni, Security in Wireless Ad Hoc Networks, in Book The Handbook of Ad Hoc Wireless Networks (Chapter 30), CRC Press LLC, 2003.

[21] N. Mistry, D. C. Jinwala, Improving AODV Protocol against Blackhole Attacks, , IMECS 2010, Hong Kong.

[22] C. Perkins. and P. Bhagwat, Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Compute, 1994.

[23] A. Shamir, "How to Share a Secret," Comm. ACM, vol. 22, pp. 612-613, 1979. [58]

[24] Y. Zhang and W. Lee, Security in Mobile Ad-Hoc Networks, in Book Ad Hoc Networks Technologies and Protocols (Chapter 9), Springer, 2005.

[25] L. Zhou and Z. J. Haas, Securing Ad Hoc Networks. IEEE Networks, Volume 13, Issue 6 1999.