CONSTRUCŢII CRIPTOGRAFICE HIBRIDE, BAZATE PE TEHNICI SIMETRICE ŞI ASIMETRICE - APLICAŢII ÎN SISTEME DE CONDUCERE

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul Automatică
de către

ing. Bogdan Ioan Groza

Conducător ştiinţific:	prof.univ.dr.ing Toma-Leonida Dragomir
Referenţi ştiinţifici:	prof.univ.dr.ing. Victor-Valeriu Patriciu
	prof.univ.dr.ing. Liviu-Cristian Miclea
	conf.univ.dr.ing. Marius Minea

Ziua susţinerii tezei: 19.07.2008

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor

Universitatea „Politehnica" din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Rezultatele relevante ale unei activităţi de doctorat se află în general în lucrările publicate care conţin contribuţiile aduse. În cele mai multe cazuri o teză de doctorat este, în opinia mea, în primul rând un material de circulaţie naţională, internă, un raport de cercetare şi doar în al doilea rând un material universal necesar de citit. Sunt destul de rare cazurile în care o teză poate fi în sine o referinţă. Mai mult, o teză marchează un simplu moment evolutiv, doctoratul este o lucrare a tinereţii, iar noţiunea de doctorat devine din nefericire din ce în ce mai nesemnificativă - poate cândva am să dedic acestui subiect o lucrare ştiinţifică. Desigur, până la urmă nu cred că se poate da o definiţie pentru ceea ce înseamnă o teză de doctorat, şi există pe câţi autori pe atâtea teze, diferite ca structură şi manieră de abordare.

Cred că prezenta lucrare este binevenită şi utilă în special pe plan naţional deoarece nivelul naţional în domeniul securităţii informaţiei este destul de slab şi pe măsură ce acest domeniu creşte în relevanţă absenţa specialiştilor în această zonă poate fi o deficienţă semnificativă. Este regretul meu, că în ultimii patru ani în care am participat activ la conferinţe naţionale şi nu numai, am găsit foarte puţini colegi români care să aibă preocupări şi cunoştinţe reale în această zonă. Cred că valoarea contribuţiilor aduse de activitatea mea doctorală este greu de apreciat acum şi eu însumi voi putea face aprecieri din ce în ce mai obiective doar în viitor. Este necesar să te poţi detaşa de necesitatea imediată a unei lucrări pentru a-i aprecia valoarea.

Înainte de a încheia aceste câteva cuvinte doresc să mulţumesc persoanelor care m-au susţinut în acest drum. O listă a acestora ar fi poate greu de realizat, dar finalizarea acestui doctorat se datorează în mare măsură îndrumărilor pe care le-am primit din partea a două persoane: Toma-Leonida Dragomir şi Marius Minea. Le mulţumesc acestora pentru sprijinul acordat. Cred că am avut de la ambii destule de învăţat şi într-un mod util pentru mine, atitudinile celor doi fiind pe alocuri diferite au reuşit să îmi lărgească orizontul. Toma-Leonida Dragomir este un bun pedagog, de la el am câştigat mult în calitatea exprimărilor. Marius Minea este un om care ţinteşte sus şi are opinii corecte în ceea ce priveşte valoarea, datorită recomandărilor lui am ajuns să public şi în conferinţe decente şi am ajuns să îmi dau seama de lipsa de valoare a multor conferinţe şi publicaţii. De la amândoi am aflat multe despre ce înseamnă rigoarea ştiinţifică.

Timişoara, Aprilie 2008 Bogdan Groza

În memoria:

inginerului M. Gârleşteanu, biologului F. König şi a matematicianului L. Vuc

Groza, Bogdan Ioan

Construcţii criptografice hibride, bazate pe tehnici simetrice şi asimetrice – aplicaţii în sisteme de conducere

Teze de doctorat ale UPT, Seria 1, Nr. 11, Editura Politehnica, 2008, 132 pagini, 30 figuri, 5 tabele.

ISSN: 1842-5208

ISBN: 978-973-625-688-2

Cuvinte cheie: autentificare, criptografie, criptanaliză, protocol criptografic, securitate

Rezumat:
Pe fondul construcţiilor criptografice hibride între tehnicile simetrice şi asimetrice, lucrarea aduce contribuţii în zona protocoalelor criptografice de autentificare, a sistemelor criptografice cu chei publice şi a puzzleurilor criptografice, precum şi o aplicaţie a unui protocol de autentificare într-un scenariu de control la distanţă. Sunt dezvoltate câteva protocoale noi pentru asigurarea autenticităţii informaţiei folosind lanţuri one-way, aceste protocoale având caracteristici similare protocoalelor bazate pe chei publice dar folosind doar funcţii one-way fără trapă eficiente din punct de vedere computaţional. Tot în acest context este propusă o tehnică nouă pentru construcţia lanţurilor one-way care face posibilă generarea unor lanţuri nemărginite în practică, utile în scenarii de broadcast pe termen nelimitat. Lucrarea prezintă şi o generalizare a criptosistemului cu cheie publică RSA şi aplicarea acestei extensii întru cadru de criptare hibridă KEM-DEM. Sunt de asemenea prezentate rezultate experimentale pentru conducerea la distanţă, prin TCP/IP, a robotului X-80 folosind unul dintre protocoalele propuse. Concluzia lucrării este că implementarea criptografiei în sisteme informatice şi în particular de conducere este necesară şi fezabilă.

CUPRINS

Notaţii, abrevieri, acronime

1^k	- mulţimea stringurilor binare de lungime k
$\oplus$	- sau exclusiv (XOR), adunare modulo 2
$\|\|$	- concatenare
$\phi(n)$	- Funcţia Euler Phi, n este un întreg oarecare
$A <_P B$	- algoritmul A se reduce în timp polinomial la B
$A \Leftrightarrow B$	- algoritmul A este echivalent cu B, deci $A <_P B$ şi $B <_P A$
AES	- Advanced Encryption Standard
CIA	- triada obiectivelor de securitate: Confidenţialitate, Integritate, Disponibilitate (Confidentiality, Integrity, Availlability)
cmmdc	- cel mai mare divizor comun
cmmmc	- cel mai mic multiplu comun
CPA	- atac de tip mesaj ales
CCA1	- atac de tip criptotext ales
CCA2	- atac de tip criptotext ales adaptiv
CSA	- protocolul Chained Stream Authentication
DCS	- sistem de control distribuit (Distributed Control Systems)
DeMA	- protocolul Delayed Message Authentication
DeMA-QR	- protocolul Delayed Message Authentication cu lanţuri de reziduuri cvadratice
DeMA-DiCA	- protocolul Delayed Message Authentication - Direct Chain Authentication
DEM	- Mecanism de încapsulare a datelor (Data Encapsulation Mechanism)
DES	- Data Encryption Standard
DiMA	- protocolul Direct Message Authentication
DoS	- Blocare a Servicilor (Denial of Services)
$E_K(m)$	- Criptarea mesajului m cu cheia k
ExtRSA	- Extensia criptosistemului RSA, se referă la cazul când

	exponentul de criptare nu este obligatoriu prim la ordinul grupului
$f^{\eta}(k)$	- compoziţia funcţiei f cu ea însăşi de η ori
fps	- cadre pe secundă (frames per second)
IFP	- problema factorizării întregilor (Integer Factorization Problem)
IND	- imposibilitatea de a distinge (Indistinguishability)
GM	- criptosistemul Goldwasser-Micali
$H(m)$	- funcţie hash aplicată mesajului *m*
KDF(x)	- Key Derivation Function, funcţie de derivare a unei chei din valoarea *x*
KEM	- mecanism de încapsulare a cheii (Key Encapsulation Mechanism)
LAN	- reţea locală (Local Area Network)
$MAC_k(m)$	- cod de autentificare a mesajului *m* cu cheia *k* (Message Authentication Code)
MD5	- Message Digest (funcţie hash)
n	- se referă în genral la un modul compozit RSA de tipul $n = p \cdot q$
N	- mulţimea numerelor naturale
NIST	- National Institute of Standards and Technology
Nonce	- parametru variant în timp, valoare generată aleator
NM	- non-maleabilitate (Non-malleability)
NTLM	- NT LAN Manager, protocol folosit în sisteme de operare Windows
$o \leftarrow A(i)$	- desemnează un algoritm a cărui intrare este *o* şi ieşire *i*
$O(\cdot)$	- limita asimptotică superioară (complexitate)
ODVA	- Open DeviceNet Vendor Association
$ord_n(x_0)$	- ordinul lui x_0 în Z_n
PAIN	- tetrada obiectivelor de securitate: Confidenţialitate, Disponibilitate-Autenticitate, Integritate, Non-repudiere (Privacy, Availlability-Authentication, Integrity, Non-repudiation)
PID	- Proporţional, Integrator, Derivator

PKCS	- Public-Key Cryptography Standards (standarde în criptografie)
Pr[X]	- probabilitatea ca evenimentul *X* să se întâmple
PTP	- Probabilist în timp polinomial (algoritm)
PWM	- Pulse Width Modulation
QRP	- Quadratic Residuosity Problem
RC2, RC6	- funcţie de criptare simetrică proiectată de Ron Rivest, abrevierea provine de la "Ron's Code" sau "Rivest Cipher"
RIPEMD	- familia de funcţii hash RACE Integrity Primitives Evaluation Message Digest
ROM	- Modelul Oracolului Aleator (Random Oracle Model)
RSA	- Rivest-Shamir-Adleman (criptosistem)
RTT	- Round Trip Time
SCADA	- Supervisory Control and Data Acquisition
SHA	- familia de funcţii hash Secure Hash Algorithm
$Sig_A(m)$	- semnătura digitală a entităţii *A* asupra mesajului *m*
S-Key	- sistem de autentificare cu parole one-time
SSL	- Secure Sockets Layer
TCP/IP	- Transmission Control Protocol / Internet Protocol
TESLA	- protocolul Timed Efficient Stream Loss-Tolerant Authentication
Timed-DeMA-QR	- protocolul Delayed Message Authentication cu lanţuri de reziduuri cvadratice şi sincronizare temporală
$v(k)$	- funcţie (cantitate) neglijabilă
Z_n	- mulţimea resturilor modulo *n*
Z_n^*	- mulţimea resturilor modulo *n* relativ prime la *n*

Lista de tabele

Lista de figuri

Context

Sub prisma noţiunii de construcţii criptografice hibride între tehnicile simetrice şi asimetrice, lucrarea introduce câteva sisteme şi protocoale criptografice noi, rezultate din împletirea unor funcţii sau proprietăţi de natură simetrică şi asimetrică, în scopul obţinerii eficienţei şi atingerii unor obiective de securitate avansate. Zona de aplicabilitate este cea a sistemelor informatice de uz comun şi a sistemelor de conducere la distanţă. Există mai multe lucrări ale autorului care conţin astfel de aplicaţii şi către care se face referire pe parcursul materialului, iar în cadrul tezei este prezentată o aplicaţie care are ca subiect implementarea unui protocol de autentificare care este folosit pentru conducerea la distanţă printr-o reţea TCP/IP a robotului X-80. În ceea ce priveşte construcţiile criptografice propuse pe parcursul tezei, acestea reprezintă soluţii care pot fi încadrate în trei categorii: protocoale de autentificare, criptosisteme cu cheie publică şi puzzleuri criptografice. Marea parte a acestor construcţii intră sub incidenţa noţiunii de construcţie hibridă între tehnicile simetrice şi asimetrice. Astfel o parte din protocoalele de autentificare, folosesc funcţii simetrice, sau funcţii peste grupuri de întregi însă doar ca funcţii one-way fără a face apel la trapa acestora, şi în acelaşi timp, datorită utilizării sincronizării temporale sau a mecanismelor challenge-response, protocoalele ating proprietăţi asimetrice, şi anume faptul că nu folosesc secrete partajate. Rezultatul este desigur eficienţa, deoarece aceste protocoale vor avea cerinţe computaţionale scăzute comparativ cu criptosistemele cu cheie publică, şi în acelaşi timp atingerea unor obiective de securitate care le fac utile în scenarii din lumea reală. Pe de altă parte criptosistemul cu cheie publică propus, bazat pe extensia funcţiei RSA, face apel la funcţii simetrice pentru a anula anumite proprietăţi algebrice care pot conduce la atacuri şi a le spori rezistenţa în faţa adversarilor activi. Totodată criptosistemul este utilizat într-un cadru hibrid de tip KEM/DEM în care se utilizează mixt criptarea asimetrică cu cea simetrică pentru a spori eficienţa computaţională. Lucrarea include fundamente teoretice pentru problemele introduse, demonstraţii asupra securităţii soluţiilor propuse şi rezultate experimentale.

1 Introducere

1.1 Temă

Într-un secol în care informaţia este vitală asigurarea securităţii acesteia este o preocupare primordială deoarece este evident că informaţia are valoare doar atunci când integritatea, autenticitatea sau alte proprietăţi importante de securitate ale acesteia sunt asigurate. Existenţa unei vulnerabilităţi şi a unui potenţial adversar înseamnă un risc de securitate iar riscurile de securitate trebuie acoperite de garanţii de securitate. Este trist de remarcat că necesitatea securităţii este luată în calcul de cele mai multe ori doar atunci când lipsa securităţii a dus la potenţiale dezastre, aşa cum exemple practice arată. În acest context, creşterea numărului de lucrări, a gradului de informare, a soluţiilor în această zonă este cu atât mai mult bine-venită. Mai mult, formarea unor experţi în această zonă este o problemă critică pentru rezolvarea problemelor de securitate care încep să fie din ce în ce mai frecvente în viaţa de zi cu zi.

În cadrul securităţii criptografia joacă un rol special, deoarece ea oferă singura garanţie de securitate când obiectul manipulat este informaţia. Tehnicile criptografice sunt cu succes utilizate într-o arie largă de aplicaţii din telefonia mobilă, sisteme de operare, sisteme bancare, instituţii publice etc. Chiar fără să ştim, în scenarii cu care ne confruntăm zi de zi există multă criptografie. De exemplu, poate că nu mulţi utilizatori ştiu că sub ecranul de login de la sistemele de operare Windows, inclusiv XP, la introducerea unei parole se calculează două criptări simetrice cu funcţia DES, sau că în spatele fiecărui site a cărui adresă începe cu "https" se desfăşoară un adevărat arsenal de primitive criptografice, cunoscut sub numele de SSL, care include funcţii criptografice simetrice, asimetrice, hash-uri etc. Astfel prezenţa securităţii criptografice în sisteme informatice este o realitate constantă şi nicidecum un ideal. Mai mult decât atât domeniul securităţii informaţiei îmbracă o natură oarecum diferită faţă de alte domenii care face ca an de an, să fie cerute soluţii noi. Aceasta deoarece securitatea implică un scenariu de tip "break and fix" în care zi de zi soluţii sunt sparte şi apoi reparate. Aceasta conduce la un dinamism foarte ridicat, care nu lipseşte nici în criptografie, o dată la câţiva ani chiar şi standardele se modifică, probabil mult mai repede decât în alte domenii. Exemple grăitoare ar fi faptul că doar DES a fost singurul standard care a rezistat mai bine de două decenii, în rest funcţii hash de exemplu, precum MD5 şi SHA1, au fost atât de rapid demolate încât practica continuă să le folosească doar din inerţie, pentru că ele de abia au intrat puternic în uz, chiar dacă de fapt sunt nesigure. Alt exemplu grăitor este şi criptarea RSA care a îmbrăcat diverse schimbări prin tehnicile de padding propuse în ultimii ani. În acest context propunerea de noi soluţii în acest domeniu, care să se bazeze pe noi paradigme, sau să ofere noi proprietăţi este bine-venită. Aici încearcă şi această teză să îşi facă loc prin propunerea unor noi soluţii şi paradigme.

Noţiunea de construcţii criptografice hibride între tehnicile simetrice şi asimetrice introdusă în titlu are un caracter general şi reflectă o realitate constantă în criptografia ultimului deceniu: primitivele criptografice simetrice şi asimetrice ajung să se împletească în practică pentru a obţine soluţii eficiente care se bazează

până la urmă pe schimburi între caracteristicile cele mai bune ale acestora. Nu mai există doar soluţii pure, care să conţină un singur tip de primitivă criptografică şi doar caracteristicile de simetrie sau asimetrie ale acesteia. Vom mai reveni la această noţiune în preambulul din secţiunea 3. Putem reţine acum ca idee centrală faptul că tema acestei teze este propunerea unor soluţii moderne bazate pe funcţii criptografice simetrice şi asimetrice pentru a rezolva probleme de securitate vitale în sisteme informatice şi de control la distanţă.

Zona de aplicabilitate a propunerilor făcute este bineînţeles cea a sistemelor informatice. Ei i se adaugă însă şi o zonă oarecum neglijată de tehnicile criptografice şi anume cea a sistemelor de control la distanţă. Evoluţia ermetică a acestei zone, prin care problemele industriei au fost izolate de vederea publicului larg, au dus la inexistenţa securităţii de natură criptografică în această zonă. Acum însă lucrurile s-au schimbat şi totul devine uşor accesibil publicului odată cu explozia Internetului. Pentru aceasta în zona aplicării criptografiei în sistemele industriale devine relevantă desfăşurarea unei activităţi de cercetare care să aducă soluţii noi, precum şi publicarea unor materiale care cel puţin să atragă atenţia asupra necesităţilor din această zonă.

Nu în ultimul rînd, prezenta lucrare reprezintă o sinteză a contribuţiilor aduse de autor în contextul mai sus amintit. Rezultate ale celor peste 20 de lucrări publicate în acest domeniu în perioada celor patru ani de doctorat sunt sintetizate în acest material. Astfel, cu privire la contribuţiile aduse putem spune că sunt contribuţii actuale, apărute în publicaţii cu vizibilitate (marea parte a lor fiind indexate BDI), care au fost recenzate sau chiar citate de către persoane interesate şi specialişti din exterior. Zona aplicaţiilor în sisteme informatice şi de control este deschisă, autorul a realizat de-a lungul celor 4 ani diverse implementări ale protocoalelor propuse în scopul obţinerii unor rezultate experimentale.

1.2 Obiective

Obiectivul principal al acestei teze a fost aducerea unor contribuţii, a unor rezultate noi care pot fi relevante pentru domeniul securităţii şi al aplicaţiilor acestuia. Contribuţiile prezente în lucrare corespund la patru direcţii de cercetare care urmează să fie pe scurt descrise. Toate aceste contribuţii se desfăşoară sub ideea de construcţii hibride între tehnicile simetrice şi asimetrice anterior amintită.

Prima este cea a construcţiei protocoalelor de autentificare. Au fost aduse contribuţii atât în zona protocoalelor de autentificare a entităţilor cât şi în zona protocoalelor de autentificare a informaţiei. Prima zonă este cea care a deschis cercetarea doctorală prin aducerea unor extensii în cadrul schemei Lamport. Dar, datorită ineficienţei în practică a acestui mecanism zona fost extinsă către protocoalele de autentificare a informaţiei, zonă care s-a dovedit mult mai fertilă şi în care au fost fixate marea parte a contribuţiilor din această lucrare. Rezultatele aduse constau în propunerea câtorva protocoale de autentificare. Tot pe aceeaşi direcţie a fost efectuată şi analiza unor protocoale propuse de alţi autori şi au fost găsite noi vulnerabilităţi ale acestora.

Cel de-al doilea obiectiv a fost fixarea unor contribuţii în construcţia criptosistemelor cu cheie publică. Criptografia cu cheie publică devine vitală pentru asigurarea securităţii sistemelor informatice. Această zonă oferă însă şi cele mai multe probleme deschise şi nerezolvate. Obiectivul a fost studiul asupra criptosistemelor bazate pe problema factorizării întregilor, rezultatul obţinut fiind

construcţia unei scheme bazate pe extensia funcţiei RSA şi utilizarea ei în cadre KEM/DEM.

Datorită implicaţiilor puternice ale atacurilor DoS (Denial of Services) în siguranţa în funcţionare a sistemelor au fost abordate şi metode pentru construcţia puzzleurilor criptografice folosite la prevenirea atacurilor DoS. Atacurile asupra disponibilităţii sistemelor joacă un rol major. În special în cadrul sistemelor de conducere garantarea disponibilităţii dispozitivelor este strict necesară. Din păcate criptografia nu poate oferi decât o soluţie parţială, şi anume puzzleurile criptografice. Rezultatul obţinut constă în construcţia unui puzzle care încearcă să ofere câteva avantaje faţă de construcţiile deja cunoscute.

Nu în ultimul rând, a fost vizată aplicarea în sisteme informatice şi de control a soluţiilor criptografice. Zona aplicaţiilor sistemelor criptografice în sisteme informatice şi de control este o zonă în care se pot realiza multe contribuţii în special datorită a două aspecte: primul este acela că există foarte multe propuneri teoretice care nu au cunoscut implementări practice şi eficienţa lor poate fi discutabilă, al doilea este acela că există multe scenarii practice care aşteaptă rezolvări ce încă nu au fost aduse. În ceea ce priveşte prezenta lucrare contribuţia în această zonă este adaptarea unuia dintre protocoalele propuse pentru un scenariu de conducere automată şi extragerea unor rezultate experimentale cu privire la eficienţa acestui protocol.

1.3 Structură

S-a optat pentru o structura care să reflecte cât mai concis necesitatea şi relevanţa acestei cercetări, şi în special contribuţiile aduse de către autor. Lucrarea este structurată pe 6 capitole. În cadrul acestora a fost alocat un spaţiu redus, de circa 20%, reprezentat de capitolele 2 şi 3, pentru preambulul general necesar înţelegerii contribuţiilor şi un spaţiu mai larg, de circa 80%, reprezentat de capitolele 4, 5 şi 6, pentru contribuţiile şi concluziile autorului. Astfel, lucrarea este structurată după cum urmează.

Capitolele 2 şi 3 au rol de preambul, ceea ce se doreşte este crearea succintă a unei imagini cu privire la necesitatea şi importanţa criptografiei în zona sistemelor informatice şi de control la distanţă precum şi a evoluţiei criptografiei în ansamblu. Este bine cunoscut că securitatea este un aspect în general neglijat în sisteme, problemele de securitate fiind puse doar după ce se înregistrează pierderi majore din lipsa asigurării acestora. Se încearcă în acest sens crearea unei imagini mai clare a noţiunii de securitate, care să îi reflecte importanţa şi actualitatea, precum şi o enumerare a tehnicilor criptografice care pot fi utilizate în acest scop.

Capitolele 4 şi 5 conţin contribuţiile autorului. Sunt prezentate contribuţii în cele patru zone amintite anterior ca obiective de cercetare. Toate contribuţiile provin din cele 20 de lucrări publicate ca prim sau unic autor, marea parte a acestor lucrări sunt parte din fluxul principal de publicaţii indexate BDI. Capitolul 4 este dedicat în exclusivitate contribuţiilor teoretice, de fundamentare şi construcţie a unor protocoale şi soluţii noi. Totuşi linia între teorie şi practică este greu de trasat pentru capitolul 4 deoarece o parte din aceste soluţii au fost gândite din considerente practice iar expunerea puternic teoretizată se datorează dorinţei de a da un fundament cât mai solid problemelor în cauză. Capitolul 5 este dedicat exclusiv implementărilor practice şi rezultatelor experimentale. Pentru a stabili eficienţa protocoalelor propuse s-a trecut la implementarea acestora, mai exact a fost

implementat unul dintre protocoalele propuse în capitolul 4 şi au fost extrase rezultate experimentale cu privire la eficienţa acestuia.

Capitolul 6 prezintă concluziile desprinse în baza rezultatelor din capitolele anterioare. Autorul a încercat să fie cât se poate de obiectiv şi să păstreze aceste concluzii cât se poate de originale. Nu s-a dorit doar extragerea unor concluzii generale, care pot fi găsite şi în alte lucrări, ci aducerea unor concluzii care reflectă opinia autorului formată pe parcursul a mai bine de 3 ani de studiu doctoral.

2 Securitate criptografică pentru sisteme informatice şi de control

În momentul în care informaţia devine un bun necesar al vieţii de zi cu zi şi în care există adversari capabili de a produce distrugeri sau deturnări ale acesteia, asigurarea securităţii informaţiei devine o necesitate. Acest lucru este unanim recunoscut de experţii în domeniu şi neglijat doar de persoanele care au o imagine idilică asupra realităţii. Încercăm să creăm o imagine asupra securităţii orientată de la necesitate la obiective în următoarele secţiuni, urmând ca în capitolul următor să fie făcută o scurtă enumerare a funcţiilor criptografice existente şi utilizate pentru rezolvarea problemelor de securitate.

2.1 Necesitatea criptografiei în sisteme informatice şi de control

Dovadă clară a necesităţii criptografiei în sisteme informatice sunt în primul rând metodele criptografice care stau la baza unor sisteme la care facem apel zi de zi. Iată câteva exemple. Sub ecranul de login de Windows, după ce am introdus parola, se declanşează un mecanism de autentificare bazat pe parole care utilizează, în funcţie de setarea sistemului, două criptări folosind DES sau un hash MD5. Clientul de mail Thunderbird, în momentul în care se conectează la siteul facultăţii noastre, sau de fiecare dată când deschidem din Firefox serviciul de webmail, foloseşte mecanismul SSL, care are la bază criptarea cu cheie publică RSA şi criptarea cu cheie simetrică AES. Exemplele pot continua, am dat aceste două exemple doar pentru faptul că mulţi dintre noi fac apel la aceste două proceduri în fiecare zi de câteva zeci de ori, folosind astfel criptografie. Doar pentru faptul că criptografia este invizibilă publicului larg, nu trebuie să ne facă să îi subestimăm importanţa.

Intrând în era informaţiei, sistemele de control la distanţă, care erau izolate şi inaccesibile publicului larg, încep acum să fie răspândite pe scară largă şi să fie deschise către public. Mai mult, aceste sisteme au fost gândite pentru a maximiza şi optimiza funcţia căreia îi sunt destinate, fără a se pune accent pe securitate. Odată deschise însă căile de comunicare cu reţelele publice şi utilizarea unor protocoale standard, sistemele industriale au devenit expuse unor potenţiali adversari dornici să exploateze eventualele vulnerabilităţi ale sistemelor. Structura generală a unui sistem de conducere industrial este sugerată în figura 2.1. Este evident că astfel de structuri vor continua să stea la baza sistemelor industriale dar ceea ce în prezent s-a schimbat este gradul de conectare, concretizat în creşterea numărului de legături cu exteriorul al echipamentelor, precum şi utilizarea unor protocoale standard de comunicare eficiente atât ca şi preţ cât şi ca şi performanţă. Acest lucru a dus la extinderea imaginii sistemelor industriale către cea sugerată în figura 2.2 (inspirată din [41]) sistemele evoluând de la incinte izolate la complexe distribuite şi de la

conexiuni punct la punct la reţele complexe. Structura de control din figura 2.1 rămâne o parte vitală şi a sistemului industrial complex din figura 2.2 dar în plus, acest schimb de imagine a dus în mod natural şi la diversificarea ameninţărilor de securitate.

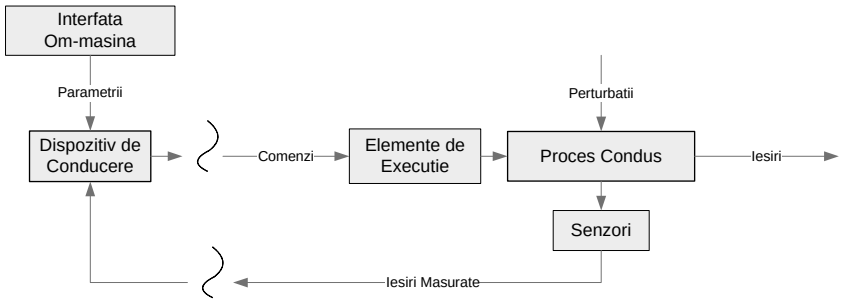


Figura 2.1. Sistem generic de control automat

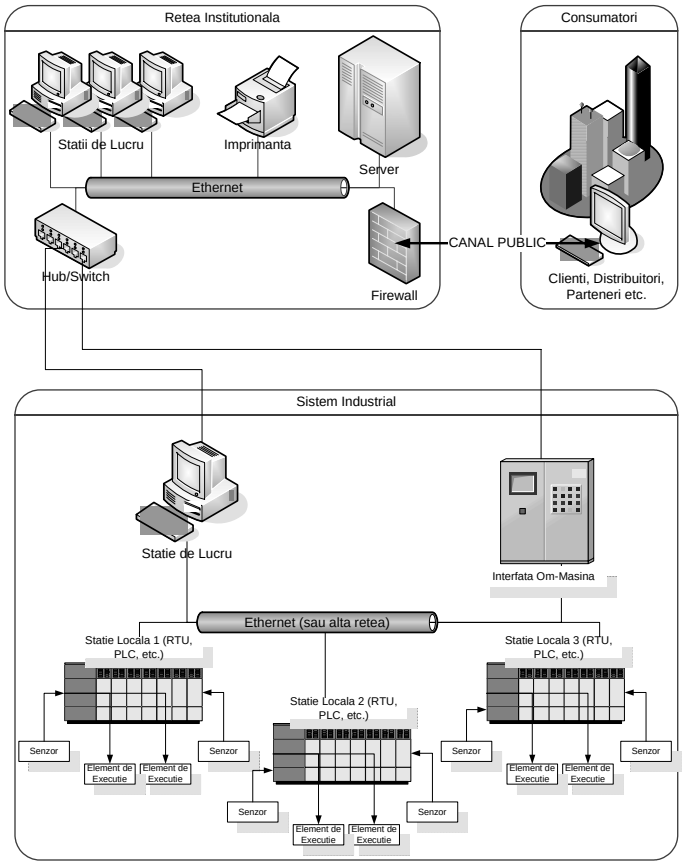


Figura 2.2. Structură de implementare contemporană a unui sistem de control industrial

Dacă în perioada 1982-2000 exista un echilibru între sursele incidentelor de securitate, aşa cum este sugerat în figura 2.3 a), acestea provenind în esenţă din trei direcţii distincte: externe, interne, accidentale; în perioada 2001-2003 pe lângă faptul că numărul de atacuri creşte semnificativ şi acest echilibru este pierdut, marea parte a problemelor de securitate începând să fie datorate factorilor externi, lucru sugerat în figura 2.3 b) [20].

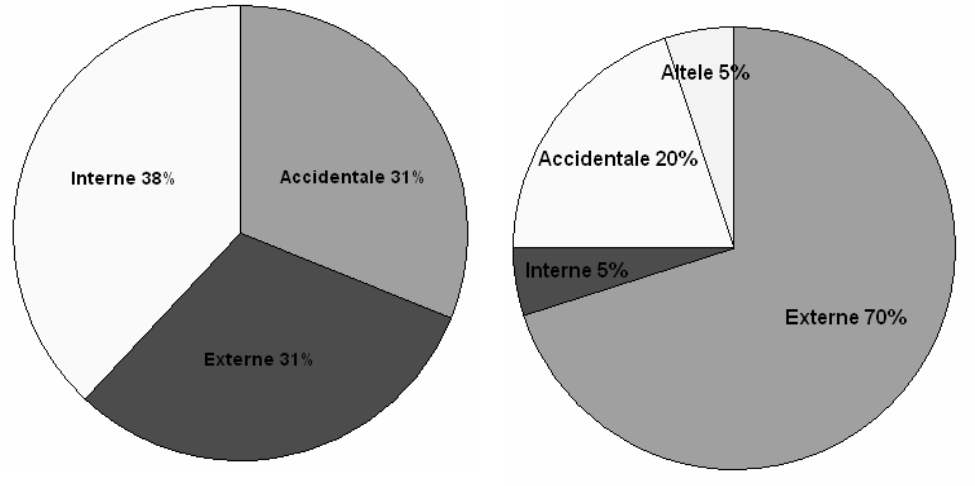
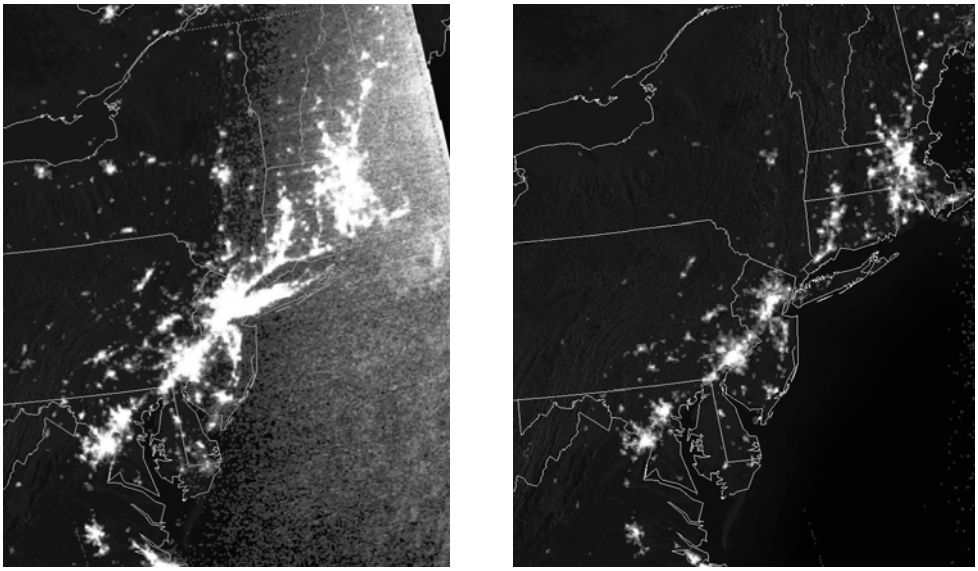


Figura 2.3. Tipuri de incidente de securitate a) în perioada 1982-2000 b) în perioada 2001-2003

Explicaţia acestui fapt vine exact din deschiderea naturală a sistemelor industriale către utilizarea reţelelor publice şi utilizarea componentelor standard impuse de piaţă. Câteva incidente raportate se găsesc în introducerea lucrării [30]. Black-out-ul din August 2003, figura 2.4 preluată din [124], este o dovadă clară a pagubelor pe care un astfel de incident poate să le aducă: o arie de 24.000 km^2 a fost afectată, 265 de centrale electrice au fost afectate din care 22 erau nucleare, 50 de milioane de oameni au fost afectaţi din care peste 21 milioane din New York. Pierderile totale sunt estimate la 6 miliarde de dolari. Una dintre cauzele căderii a fost o problemă în sistemele informatice care nu au declanşat semnalul de alarmă făcând astfel ca luarea de măsuri pentru a preveni căderea să întârzie până când a fost prea târziu. Chiar dacă acest incident nu este datorat unor probleme legate de securitatea criptografică, care este subiectul prezentei teze, incidentul este relevant pentru amploarea pagubelor care pot fi cauzate. Desigur, este doar o problemă de timp până când adversari, precum mişcările extremiste de exemplu, ar putea exploata vulnerabilităţi de securitate criptografică pentru a cauza pagube similare. Acest lucru este recunoscut în mod curent în special în Statele Unite în sectoarele de distribuţie a gazului şi electricităţii aşa cum lucrări de ultimă oră arată.

Riscurile şi ameninţările variază într-un spectru relativ larg [4]. În topul primelor 10 deficienţe de securitate ale reţelelor SCADA lucrarea (McAfee, 2007)

enumeră: politici de securitate inadecvate, sisteme slabe de control ale reţelelor, proasta configurare a sistemelor, comunicare wireless neadecvată, autentificare neadecvată la comunicare, lipsa mecanismelor de detecţie şi restricţionare a drepturilor de acces la sisteme de control, absenţa uneltelor pentru detecţia şi raportarea activităţilor anormale, utilizarea reţelei pentru trafic neautorizat, lipsa mecanismenlor de detecţie a erorilor ce pot conduce la epuizarea bufferelor, lipsa mecanismelor de control a schimbului de software. Multe dintre aceste ameninţări au ca soluţie utilizarea criptografiei, de exemplu ameninţările legate de autenticitatea comenzilor şi de controlul accesului. Iar potenţialele atacuri încep să se diversifice de la o aplicaţie la alta conducând spre o paletă relativ largă de obiective de securitate ce trebuie asigurate.



a) b)

Figura 2.4. Căderea de energie electrică din August, 2003: a) înainte b) după

Alte exemple relevante sunt în lucarea [73] unde între deficienţele de securitate în sisteme SCADA se enumeră următoarele: lipsa definiţiilor formale, lipsa unui buget de securitate incremental, absenţa expertizei necesare în domeniul securităţii, incapacitatea de a instala uşor tehnologia necesară, necesitatea educaţiei cu privire la bunele practici în vederea asigurării securităţii. Sigur toate aceste deficienţe nu sunt uşor de asigurat şi există şi recomandări concrete în vederea asigurării securităţii. De exemplu lucrarea [118] indică 21 de paşi pentru a asigura securitatea sistemelor SCADA, între aceştia mulţi au la bază utilizarea unor tehnici criptografice: de exemplu pasul 5 se referă la eliminarea protocoalelor customizate a căror securitate se bazează pe obscuritatea protocolului (singura soluţie care nu se bazează pe obscuritatea protocolului este criptografia) iar pasul 21 se referă la introducerea unor politici de securitate şi traininguri pentru ca personalul să fie

conştient în protejarea informaţiei senzitive (inclusiv parolele care sunt un ingredient criptografic). Nu în ultimul rând există şi produse program destinate asigurării securităţii în sisteme SCADA precum IntruShield de la McAfee (lucrarea [84] recomandă IntruShield în acest scop, altfel produsul este de uz general).

Din perspectiva securităţii informaţiei acolo unde există o vulnerabilitate şi un potenţial adversar există şi un risc, iar acoperirea acestor riscuri necesită garanţii. În acest context tehnicile criptografice joacă un rol special deoarece ele oferă singura garanţie atunci când obiectul manipulat este informaţia. Tehnicile criptografice sunt în mod uzual utilizate pentru a asigura securitatea informaţiei în sisteme bancare, sisteme medicale, telefonie etc. iar interesul pentru utilizarea tehnicilor criptografice în sisteme de control distribuite DCS (Distributed Control Systems) şi sisteme de achiziţie şi control SCADA (Supervisory Control and Data Acquisition) a crescut, acest fapt fiind demonstrat de cele mai recente lucrări care îndreaptă atenţia către acest domeniu [30], [33], [41], [73], [91], [104], [116], [118], [125]. Problema utilizării tehnicilor criptografice în astfel de medii se pune în contextul în care criptografia nu este uşor de adaptat şi implementat pentru astfel de scenarii, deoarece introduce întârzieri ce pot conduce la proasta funcţionare a sistemului şi necesită resurse de calcul sau comunicare care uneori nu sunt disponibile. Este bine cunoscut faptul că sistemele industriale au caracteristici diferite de cele ale sistemelor de calcul tradiţionale şi acest lucru ar putea ridica semne de întrebare cu privire la fezabilitatea implementării criptografiei. Totuşi, rezultate recente arată că implementarea criptografiei este posibilă chiar şi în medii cu resurse de calcul reduse. Un exemplu concret sunt reţelele ad-hoc de senzori, aceste reţele au putere de calcul şi abilitaţi de comunicare extrem de reduse. Faptul că primitivele criptografice au fost propuse şi utilizate cu succes în aceste medii [2], [29], [82], [81], [92], [98], [109] este un semn bun şi arată că este doar o chestiune de timp până când tehnicile criptografice vor fi introduse şi în sistemele de control la distanţă.

Putem concluziona că soluţiile de securitate bazate pe criptografie pentru sisteme industriale sunt necesare, există pe piaţă şi vor continua să apară, adoptarea lor pe scară largă fiind doar o chestiune de timp.

2.2 Obiective de securitate şi soluţii criptografice ale acestora

În ceea ce priveşte obiectivele de securitate care trebuie asigurate în sistem acestea variază într-un diapazon destul de larg. În ceea ce priveşte criptografia în general, şi nu securitatea în ansamblu, putem distinge patru obiective majore de securitate care sunt recunoscute de orice autor în domeniu şi care sunt prezente atât în sisteme informatice de uz comun precum şi în sisteme de control industrial: confidenţialitate, integritate, autenticitate, non-repudiere.

- **Confidenţialitatea**

Confidenţialitatea înseamnă asigurarea faptului că informaţia rămâne accesibilă doar părţilor autorizate în acest sens. Acesta este cel mai vechi obiectiv al criptologiei. În rândul necunoscătorilor este încă larg răspândită opinia că noţiunea de criptografie este sinonimă cu confidenţialitate. Sigur opinia este eronată pentru

că criptografia se ocupă şi de asigurarea multor alte obiective, ce vor fi enumerate în continuare, şi care nu au nici o legătura cu păstrarea secretă a informaţiei. În ceea ce priveşte asigurarea acestui obiectiv prin tehnici criptografice, el este îndeplinit cu ajutorul funcţiilor de criptare. În general datorită eficienţei se folosesc funcţii simetrice, dar scenarii practice conduc în general la orchestrarea acestora cu funcţii asimetrice.

- **Integritatea**

Integritatea se referă la asigurarea faptului că informaţia nu a fost alterată pe parcursul transmisiei sau de către un posibil adversar. Funcţiile criptografice utilizate în acest scop sunt funcţiile hash, sau codurile de autentificare MAC, care fac ca modificarea unui singur bit de informaţie să poată fi detectată. Menţionăm că în principiu este greşită utilizarea funcţiilor de criptare simetrice şi asimetrice în acest scop, instrumentul criptografic dedicat fiind funcţiile hash şi codurile MAC. Totuşi, funcţiile simetrice şi asimetrice contemporane pot asigura şi integritatea dar alegerea lor pentru acest scop trebuie făcută cu precauţie.

- **Autentificarea**

Autentificarea are două coordonate distincte: autentificarea entităţilor şi autentificarea informaţiei. Autentificarea entităţilor se referă la existenţa unei garanţii cu privire la identitatea unei anume entităţi. Autentificarea informaţiei se referă la determinarea sursei de provenienţă a informaţiei – în mod implicit aceasta garantează şi integritatea informaţiei deoarece dacă informaţia nu mai are integritate, deci un potenţial adversar a alterat-o, atunci nici sursa ei de provenienţă nu mai este aceeaşi. Însă doar integritatea informaţiei nu implică şi autenticitatea ei. Aceasta deoarece autentificarea este strâns legată de un factor temporal, este evident că o informaţie stocată poate fi supusă unui test de integritate pentru a se constata dacă a fost sau nu alterată dar nu poate fi supusă unui test de autenticitate dacă nu există o garanţie cu privire la momentul de timp la care entitatea de care este legată a depozitat-o (deoarece în acest caz informaţia putea fi replicată şi depusă de orice altă entitate). Autentificarea se realizează în general prin protocoale care pot avea la bază întreg arsenalul de funcţii criptografice: funcţii hash, MAC, criptări simetrice şi asimetrice, semnături digitale.

- **Non-repudierea**

Non-repudierea previne o entitate în a nega o acţiune întreprinsă (acţiune materializată desigur în transmisia unei informaţii). Aceasta înseamnă că dacă la un moment dat o entitate neagă ca ar fi emis o anume informaţie, entitatea care a primit informaţia respectivă poate demonstra unei părţi neutre că informaţia provine într-adevăr de la entitatea în cauză. Non-repudierea se realizează prin utilizarea semnăturilor digitale.

Este util de spus, în special din considerente istorice în domeniu, că în ultimii 20 de ani în securitatea informaţiei trei obiective au fost considerate ca fiind fundamentale, acestea formează aşa numita triadă CIA: confidenţialitate, integritate, disponibilitate („CIA” adică Confidentiality, Integrity, Availability) [122]. Recent însă, în 2002, Donn Parker a propus extinderea acestora la şase obiective

care poartă numele de Hexada Parkeriană: Confidenţialitate, Posesie sau Control, Integritate, Autenticitate, Disponibilitate, Utilitate (Confidentiality, Possession or Control, Integrity, Authenticity, Availability, and Utility) [123]. De asemenea, la fel de bine cunoscută şi vehiculată este tetrada PAIN care provine de la Confidenţialitate, Disponibilitate-Autenticitate, Integritate, Non-repudiere (Privacy, Availlability-Authentication, Integrity, Non-repudiation).

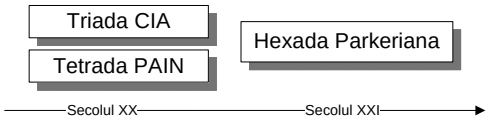


Figura 2.5. Evoluţia obiectivelor de securitate

Desigur că aceste obiective nu sunt o sinteză completă a obiectivelor de securitate existente în practică. Există desigur şi alte obiective, ceva mai particulare dar la fel de relevante, care trebuie să le amintim. Rămâne o problemă deschisă dacă obiectivele următoare pot sau nu să fie derivate din cele amintite anterior, util pentru această lucrare este să le enumerăm şi să le explicăm succint în cele ce urmează.

- **Actualitatea**

Actualitatea informaţiei se referă la asigurarea faptul că informaţia primită este actuală (proaspătă). Acest aspect poate fi interpretat în două moduri: pe de o parte se referă la faptul că informaţia poate expira după o anumită perioadă de timp, pe de altă parte se referă la faptul că un posibil adversar ar putea schimba ordinea în care pachetele cu informaţie ajung la destinaţie, diverse scenarii pot fi imaginate. Se realizează în general prin utilizarea parametriilor varianţi în timp: amprente temporale, numere aleatoare, contoare etc.

- **Anonimitatea**

Anonimitatea se referă la împiedicarea identificării identităţii unei entităţi care a solicitat un serviciu. Spre exemplu, poate fi extrem de util în tranzacţii bancare când nu se doreşte identificare persoanei care sau către care se face o plată, sau în servicii de e-mail pentru păstrarea anonimităţii expeditorului. Se realizează fie prin protocoale, fie prin funcţii criptografice adaptate acestui scop. De exemplu există un puternic segment de cercetare în zona funcţiilor de criptare cu renegare (deniable encryption), prin care se poate cripta informaţie al cărei conţinut poate fi schimbat la decriptare, făcând astfel renegabilă orice informaţie criptată, însă nu există în acest sens soluţii eficiente până în prezent.

- **Autorizarea**

Autorizarea se referă la controlul accesului şi la prevenirea intrării agenţilor neautorizaţi în sistem. Relaţia între obiectivul autentificării entităţilor şi controlul

accesului constă în aceea că cel din urmă obiectiv se construieşte în general pe primul (e normal să fie necesară o metodă de a autentifica entitatea înainte de a-i permite accesul) dar obiectivele sunt totuşi distincte. Aceasta deoarece autorizarea înseamnă utilizarea unui mecanism de autentificare şi a unei politici de securitate pentru a decide dreptul de acces al unor entităţi asupra unor resurse.

- **Disponibilitatea**

Disponibilitatea se referă la asigurarea faptului că un serviciu este accesibil atunci când un utilizator legitim îl solicită. Asigurarea acestui obiectiv presupune că o entitate neautorizată nu poate bloca accesul unei entităţi autorizate la serviciile sistemului. În acest caz însă nu intră în discuţie problemele legate de autorizarea accesului, anterior amintite, ci cele de disponibilitate a resusei în sine. Aceasta presupune a evita problemele de epuizare a resurselor sistemului din cauza utilizării nelegitime a acestora. Atacurile asupra acestui obiectiv sunt cele de tip Denial of Services (DoS) şi cauzează atât pagube economice dar şi de siguranţă în funcţionare. Una dintre măsurile criptografice în asigurarea acestui obiectiv este utilizarea puzzleurilor criptografice pentru protecţia în faţa atacurilor de tip DoS.

- **Protecţia părţilor terţe**

Protecţia părţilor terţe se referă la evitarea transmiterii pericolului asupra părţilor cu care există o legătură. De exemplu atacul asupra unei anume componente IT nu va defecta şi altă componenta, sau din punct de vedere economic: căderea unei componente datorită unei erori de manipulare nu va duce la discreditarea producătorului.

- **Revocarea**

Revocarea se referă la posibilitatea de a revoca un drept oferit. Cel mai relevant exemplu în legătură cu criptografia este posibilitatea de a revoca un certificat de cheie publică de către entitatea care l-a emis.

- **Trasabilitatea**

Trasabilitatea sau urmărirea unui sistem se referă la posibilitatea de a reconstrui istoricul funcţionării sistemului pe baza înregistrărilor, de exemplu înregistrarea comenzilor relevante, a persoanelor care le-au lansat etc. Obiectivul este relevant în determinarea cauzelor eventualelor problemelor de funcţionare, deci este utilizat în diagnosticare.

Nici această listă de obiective nu este completă, aproape fiecare carte în domeniu prezintă liste mai mult sau mai puţin complete, iar prezentarea lor exhaustivă poate fi în sine subiectul unei cărţi. Totuşi aceste obiective pot crea o imagine parţială asupra a ceea ce trebuie protejat în sisteme informatice şi de control şi asupra modului în care criptografia poate ajuta pentru aceasta. Nu în ultimul rând este important de subliniat faptul că un sistem industrial complex, precum cel prezentat în figura 2.2, necesită asigurarea acestor obiective. Aceeaşi este situaţia şi pentru sistemul de control din figura 2.1 care este parte integrantă a sistemului complex, pentru acest caz asigurarea obiectivului de autenticitate a

informaţiei vehiculate între dispozitivul de conducere şi procesul condus este vitală pentru buna funcţionare a sistemului complex. Este unanim recunoscut de experţii în domeniu, faptul că autenticitatea informaţiei este cel mai relevant obiectiv în sisteme de conducere, iar capitolul 5 al tezei va aduce câteva soluţii bazate pe criptografie pentru rezolvarea acestei probleme.

3 Funcţii şi protocoale criptografice

În scopul încadrării contribuţiilor din capitolele 4 şi 5 în peisajul contemporan al criptografiei, în prezentul capitol încercăm să construim o scurtă sinteză asupra celor mai relevante construcţii criptografice şi protocoale. Din acesta sinteză nu lipseşte formarea unui reper în ceea ce priveşte orientarea şi tendinţele către care se îndreaptă domeniul criptografiei.

3.1 Privire de ansamblu asupra funcţiilor criptografice

Pentru claritatea expunerii de pe parcursul tezei, în această secţiune dorim să facem o trecere în revistă asupra funcţiilor criptografice existente. Scopul secţiunii este doar introducerea unor definiţii formale asupra acestora, pentru detalii şi explicaţii suplimentare poate fi consultată orice carte de specialitate în domeniu cum ar fi [83], [85], [105] sau chiar cartea autorului [45]. Prin sistem criptografic, sau simplu criptosistem, înţelegem un ansamblu format din trei algoritmi: un algoritm de generare a cheilor (cheie de criptare şi cheie de decriptare), un algoritm de criptare şi un algoritm de decriptare – acest lucru este sugerat în figura 3.1. Se foloseşte frecvent, fără a pierde din semnificaţii, termenul de funcţie în locul celui de algoritm, de exemplu spunem funcţie de criptare sau funcţie de decriptare cu referire la algoritmii de criptare şi decriptare. Noţiunea de criptosistem aşa cum a fost anterior definită nu este deloc rigidă ea putând de fapt să surprindă toate primitivele criptografice ce vor fi descrise în continuare, fiind un simplu exerciţiu de imaginaţie a gândi scenariile pentru care algoritmul de generare a cheii poate să lipsească sau algoritmii de criptare sau decriptare îndeplinesc alte roluri decât criptarea sau decriptarea efectivă a informaţiei.

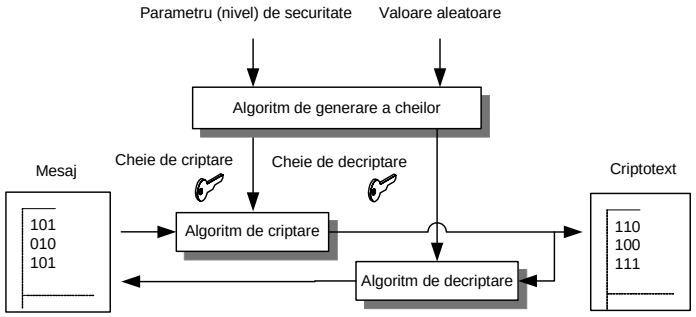


Figura 3.1. Sistem criptografic

➢ **Funcţie one-way**

O funcţie greu de inversat nu trebuie confundată cu o funcţie neinversabilă. Din punct de vedere matematic o funcţie neinversabilă este o funcţie care nu este bijectivă, în timp ce o funcţie greu de inversat este o funcţie a cărei inversă nu poate fi calculată în mod eficient. Această noţiune poate fi uşor formalizată dacă considerăm că o funcţie este one-way dacă orice algoritm probabilist în timp polinomial ar putea să o inverseze doar cu o probabilitate neglijabilă. În general se merge însă mult mai departe de atât şi se impune imposibilitatea găsirii unei coliziuni în imaginea acestei funcţii, condiţie care este chiar mai puternică decât aceea de a inversa efectiv funcţia. Acest aspect este sintetizat în următoarea definiţie.

Definiţia 3.1. (Funcţie one-way (puternică one-way)) O funcţie $f : D_f \to C_f$ se numeşte one-way dacă:

1) Există un algoritm PTP (Probabilist în Timp Polinomial) care calculează $y = f(x)$ pentru aproape orice valoare $x \in D_f$.

2) Orice algoritm PTP care primeşte ca intrare pe $y = f(x)$ returnează cu o probabilitate neglijabilă o valoare z astfel încât $f(z) = y$, adică pentru o valoare k numită parametru de securitate avem $\Pr\left[x \xleftarrow{\Theta} D_f, y \leftarrow f(x), z \leftarrow A(1^k, y) : f(z) = y\right] \le v_A(k)$ (simbolul $\xleftarrow{\Theta}$ denotă că valoarea a fost aleasă aleator).

Aici, prin $v_A(k)$ desemnăm o funcţie neglijabilă, adică o funcţie care este mai mică decât inversul oricărui polinom într-un punct începând de la o anumită valoare.

Toate funcţiile criptografice introduse în cele ce urmează au la bază o astfel de funcţie one-way.

➢ **Funcţii hash**

O funcţie hash este o funcţie one-way care primeşte ca intrare mesaje de dimensiune variabilă şi returnează un mesaj de lungime fixă din care mesajul iniţial nu poate fi recuperat. Funcţiile hash nu folosesc nici un fel de cheie, şi le vom nota cu $H(m)$ - funcţie hash aplicată mesajului m. În general faţă de o funcţie hash se impun următoarele cerinţe de securitate:

1) Rezistenţă primară a imaginii (sau target collision resistance): fiind dat $H(x)$ nu se poate găsi x.
2) Rezistenţă secundară a imaginii: fiind dat $x, H(x)$ nu se poate găsi $x' \neq x$ astfel încât $H(x) = H(x')$.
3) Rezistenţă la coliziune: nu se poate găsi o pereche x, x' astfel încât $H(x) = H(x')$.

➢ **Coduri MAC**

Codurile de autentificare a mesajelor, MAC (Message Authentication Codes), le notăm cu $MAC_K(m)$ - cod de autentificare a mesajului m calculat cu cheia k. Indiferent de dimensiunea în biţi a mesajului dimensiunea ieşirii funcţiei este constantă (de obicei dimensiunea cheii este egală cu dimensiunea ieşirii funcţiei). Din punct de vedere formal un cod de autentificare al informaţiei îl definim după cum urmează:

Definiţia 3.2. (Cod de autentificare a mesajelor) Un cod de autentificare a mesajelor *MAC* este un ansamblu de trei algoritmi PTP $\{MAC.Gen(1^k), MAC.Tag(m,K), MAC.Ver(m,K,\tau)\}$ având proprietăţile:

1) $MAC.Gen(1^k)$: este un algoritm PTP care la intrarea $1^k, k \in Z_{x>0}$ returnează o cheie secretă K, i.e. $K \leftarrow MAC.Gen(1^k)$, uniform aleasă din spaţiul cheilor, i.e. $K \xleftarrow{\Theta} Keys(MAC)$.
2) $MAC.Tag(m,K)$: este un algoritm PTP care primeşte la intrare un mesaj m din mulţimea mesajelor posibile, o cheie simetrică K şi returnează valoarea codului de autentificare, denumit generic *tag* (etichetă) τ, i.e. $\tau \leftarrow MAC.Tag(m,K)$.
3) $MAC.Ver(m,K,\tau)$: este un algoritm PTP care ia ca intrare un tag τ, o cheie secretă K şi un mesaj m, şi returnează un bit b de valoare 1 dacă tripletul de intrare este valid şi 0 în caz contrar, i.e. $b \leftarrow MAC.Ver(m,K,\tau), b \in \{0,1\}$.

Următoarea definiţie stabileşte securitatea unui MAC şi este frecvent folosită în demonstraţii de securitate pentru protocoalele care folosesc MAC-uri.

Definiţia 3.3. Un MAC este o familie de funcţii one-way $\{f_K(x)\}$ determinate de o cheie K pentru care:

1) Există un algoritm PTP care pentru orice x calculează $f_K(x)$.
2) Pentru orice algoritm PTP care primeşte la intrare un set de mesaje $m_i, i = \overline{1,q}$ şi valoarea lui f_K calculată asupra acestor mesaje, probabilitatea ca algoritmul să returneze o pereche (m', a) astfel încât $m' \neq m_i, \forall i = \overline{1,q}$ şi $f_K(m') = a$ este neglijabilă.

➢ **Criptări simetrice**

Criptarea simetrică a mesajului m cu cheia secretă K o vom nota cu $E_K(m)$, E reprezintă funcţia de criptare care este întotdeauna cunoscută, iar securitatea criptosistemului (a cifrului) depinde de păstrarea secretă a cheii K. A cripta înseamnă a aplica transformarea E_K asupra mesajului lui m pentru a obţine criptotextul $c = E_K(m)$ iar a decripta înseamnă a aplica transformarea $E_{K^{-1}}$ (adeseori notată şi ca D_K) asupra criptotextului c pentru a obţine mesajul m.

Pentru algoritmii simetrici, prin convenţie, presupunem $K = K^{-1}$ prin aceasta înţelegând că putem calcula uşor cheia de decriptare din cea de criptare, subliniem de asemenea faptul că aceasta nu înseamnă că aceste chei sunt identice (şi nici nu sunt în cele mai multe cazuri practice) ci doar că una este uşor de dedus din cealaltă (prin antiteză, pentru algoritmii asimetrici ce vor fi introduşi în paragraful următor, presupunem $K \neq K^{-1}$ prin care înţelegem că nu este fezabil a calcula cheia de decriptare din cea de criptare).

Definiţia 3.4. (Criptare cu cheie simetrică) Un sistem de criptare cu cheie simetrică este un ansamblu de trei algoritmi PTP $\{Sym.Gen(1^k),\ Sym.Enc(m,K),\ Sym.Dec(c,K)\}$ având proprietăţile:

1) $Sym.Gen(1^k)$: este un algoritm PTP care la intrarea $1^k, k \in Z_{x>0}$ returnează o cheie secretă K, i.e. $K \leftarrow Sym.Gen(1^k)$, aleasă uniform din spaţiul cheilor.
2) $Sym.Enc(m,K)$: este un algoritm PTP care primeşte la intrare un mesaj m din mulţimea mesajelor posibile, o cheie simetrică K şi returnează criptotextul c dacă mesajul este valid respectiv $\perp$ în caz contrar, i.e. $c \leftarrow Sym.Enc(m,K), c \in \{0,1\}^k \cup \{\perp\}$.
3) $Sym.Dec(c,K)$: este un algoritm PTP care ia ca intrare un criptotext c din mulţimea criptotextelor, o cheie secretă K şi returnează un mesaj m în cazul în care criptotextul este valid respectiv $\perp$ în caz contrar, i.e. $m \leftarrow Sym.Dec(c,K), m \in \{0,1\}^k \cup \{\perp\}$.

- **Criptări asimetrice**

Criptarea asimetrică, o notăm similar cu cea simetrică, pentru claritate însă schimbăm cheia K cu PK, astfel avem $c = E_{PK}(m)$ (aceasta semnificând criptarea cu cheia PK a mesajului. Decriptarea se face folosind cheia privată ca $m = D_{SK}(c)$ (în general se foloseşte noţiunea de cheie privată şi nu secretă). Criptarea cu cheie asimetrică comparativ cu cea cu cheie simetrică are ca dezavantaj viteza şi prezintă două avantaje majore: i) nu necesită schimbul prealabil de chei secrete, deci comunicaţia poate fi efectuată şi pe un canal nesigur fără să existe secrete partajate şi ii) numărul de chei partajate la comunicarea între n entităţi este minim (o cheie publică şi o cheie privată pentru fiecare entitate). Din punct de vedere formal un criptosistem cu cheie publică îl definim după cum urmează:

Definiţia 3.5. (Criptare cu cheie publică). Un sistem de criptare cu cheie publică este un ansamblu de trei algoritmi PTP $\{PKE.Gen(1^k),\ PKE.Enc(m,PK),\ PKE.Dec(c,SK)\}$ având proprietăţile:

1) $PKE.Gen(1^k)$: este un algoritm PTP care la intrarea $1^k, k \in Z_{x>0}$ returnează o pereche cheie secretă, cheie publică (PK,SK), i.e. $(PK,SK) \leftarrow PKE.Gen(1^k)$.

2) $PKE.Enc(m, PK)$: este un algoritm PTP care primeşte la intrare un mesaj m din mulţimea mesajelor posibile, o cheie publică PK şi returnează criptotextul c, i.e. $c \leftarrow PKE.Enc(m, PK)$.

3) $PKE.Dec(c, SK)$: este un algoritm PTP care ia ca intrare un criptotext c din mulţimea criptotextelor, o cheie secretă SK şi returnează un mesaj m, i.e. $m \leftarrow PKE.Dec(c, SK)$.

Se impune ca pentru fiecare pereche cheie publică, cheie privată $(PK, SK) \leftarrow PKE.Gen(1^k)$ şi pentru orice mesaj m să fie valabil $m \leftarrow PKE.Dec(PKE.Enc(m, PK), SK)$. Aceasta înseamnă că criptosistemul furnizează rezultate corecte şi întotdeauna decriptarea criptării unui mesaj returnează acel mesaj.

- **Semnături digitale**

Semnăturile digitale reprezintă echivalentul electronic al semnăturilor de mână, acest concept fiind în mare introdus ca proprietate adiţională a criptosistemelor cu cheie publică de către Diffie şi Hellman în 1976, în absenţa unei scheme criptografice pentru acest scop. Obiectivul principal de securitate pe care îl asigură semnăturile digitale îl reprezintă non-repudierea, şi anume faptul că o entitate odată ce a semnat o informaţie nu poate nega că a emis acea informaţie şi orice altă entitate neutră poate verifica acest lucru. Semnăturile digitale reprezintă o valoare numerică care leagă conţinutul unui mesaj de identitatea unei entităţi. Definiţia formală a unei scheme de semnătura digitală este următoarea:

Definiţia 3.6. (Semnături digitale). Un sistem de semnătură digitală este un ansamblu de trei algoritmi PTP $\{Sig.Gen(1^k), Sig.Sign(m, SK), Sig.Ver(m, S, PK)\}$ având proprietăţile:

1) $Sig.Gen(1^k)$: este un algoritm PTP care la intrarea $1^k, k \in Z_{x>0}$ returnează o pereche cheie secretă, cheie publică (PK, SK), i.e. $(PK, SK) \leftarrow Sig.Gen(1^k)$.

2) $Sig.Sign(m, SK)$: este un algoritm PTP care primeşte la intrare un mesaj m din mulţimea mesajelor posibile, o cheie secretă SK şi returnează semnătura asupra mesajului s, i.e. $s \leftarrow Sig.Sign(m, SK)$.

3) $Sig.Ver(s, m, PK)$: este un algoritm PTP care ia ca intrare un mesaj m, semnătura acestuia s şi o cheie publică PK, şi returnează un bit b care este 1 dacă semnătura corespunde mesajului şi 0 în caz contrar, i.e. $b \leftarrow Sig.Ver(m, s, PK), b \in \{0,1\}$.

3.2 Evoluţie, stadiu curent şi orientare în criptografie

3.2.1 Lipsa securităţii în sisteme clasice

În trecut sistemele criptografice erau construite pentru a atinge obiective de securitate rudimentare precum confidenţialitatea de tip totul sau nimic („all or nothing secrecy") prin aceasta înţelegând că un adversar poate afla fie totul despre mesajul criptat fie nimic, sau altfel spus dacă nu a aflat totul despre mesajul criptat e ca şi când nu ar fi aflat nimic – ceea ce presupune practic inversarea totală a funcţiei one-way pentru a sparge criptosistemul. Un astfel de deziderat de securitate este însă insuficient în practică pentru că folosirea schemelor criptografice în variantă „text-book" face posibilă aflarea unor informaţii parţiale despre textul criptat şi este evident că şi acest lucru poate duce la pierderea securităţii. De exemplu criptarea RSA nu modifică simbolul Jacobi al mesajului criptat, astfel, un adversar poate oricând face distincţie între criptările a două mesaje cu simboluri Jacobi diferite.

Totodată, în trecut securitatea era gândită în faţa unor adversari pasivi care în principiu analizau criptotextul în vederea găsirii mesajului sau a cheii. În lumea reală adversarii nu sunt pasivi ci activi, având acces la maşinile de criptare şi decriptare. În faţa atacurilor active, sistemele în variantă "text-book" sunt nesigure. Astfel, cărţile aparţinând secolului trecut din domeniul criptografiei, precum [85], [105], care nu ţineau cont de aceste probleme de securitate, prezintă criptosisteme clasice care nu sunt sigure. Orice carte contemporană poate fi consultată însă pentru variante sigure ale algoritmilor clasici [83] şi următoarele secţiuni ale tezei sunt dedicate unui studiu sumar asupra acestei probleme. De asemenea soluţiile introduse de prezenta teză răspund unor obiective avansate de securitate precum cele introduse ca răspuns în faţa lipsei de securitate în variantele clasice.

3.2.2 Noţiuni avansate de securitate

Datorită ineficienţei securităţii de tip totul sau nimic criptosistemele contemporane trebuie să atingă obiective de securitate avansate precum ne-distingerea criptotextelor IND (indistinguishability of encryptions) şi non-maleabilitatea criptotextelor NM (non-malleability). Noţiunile de IND şi NM apar cu preponderenţă tratate în cadrul criptosistemelor cu cheie publică, dar ele sunt noţiuni valide şi în cazul altor criptosisteme. Prin IND, noţiune introdusă de Goldwasser şi Micali [43], se înţelege faptul că un adversar nu poate afla nici un fel de informaţie cu privire la un mesaj având doar criptotextul aferent – în mod ideal aceasta înseamnă că ceea ce un adversar ştie având criptotextul, ştie şi fără criptotext. Prin NM, noţiune introdusă de [27], se înţelege faptul că un adversar nu poate construi un criptotext având un criptotext dat la care nu cunoaşte mesajul aferent astfel încât între mesajele aferente să existe o relaţie cunoscută de către adversar. Noţiunea de IND a apărut sub diverse forme în literatura de specialitate, dintre acestea amintim: securitate semantică şi securitate polinomială (toate noţiunile au aceeaşi semnificaţie în securitate).

Mai mult, toate aceste obiective trebuie să fie atinse în prezenţa unor adversari activi care pot fi împărţiţi în trei categorii:

1) CPA (chosen plaintext attack) desemnează adversari care au acces la maşina de criptare. Este evident că orice criptosistem cu cheie publică

trebuie să fie rezistent CPA deoarece orice adversar are acces la maşina de criptare cheia de criptare fiind în mod evident publică – în acest sens nu se discută niciodată de rezistenţa CPA a unui criptosistem cu cheie publică, aceasta fiind o proprietate elementară pe care trebuie să o îndeplinească.

2) CCA1 (non-adaptive chosen ciphertext attack) desemnează adversari care au acces neadaptiv la maşina de decriptare. Mai exact adversarul are acces la maşina de decriptare până la momentul la care primeşte valoarea criptotextului care trebuie atacat, moment la care pierde accesul la maşina de decriptare.
3) CCA2 (adaptive chosen ciphertex attack) desemnează adversarii care au acces adaptiv la maşina de decriptare, adică accesul la maşină rămâne valabil chiar şi după primirea valorii criptotextului care trebuie atacat. Deoarece atacul CCA1 este considerat oarecum perimat, adeseori în literatura de specialitate se vorbeşte doar de atac CCA prin acesta înţelegându-se de fapt atacul de tip CCA2.

Grupând cele 3 obiective de securitate {IND, NM} cu cele 3 tipuri de adversari {CPA, CCA1, CCA2} obţinem 6 noţiuni de securitate în cazul criptosistemelor, acestea sunt: IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, NM-CCA2. Lucrarea [9] este cea care a oferit prima abordare unitară a acestor noţiuni de securitate, în particular pentru criptosistemele cu cheie publică, definind legăturile între ele. În figura 3.2. sunt sintetizate aceste legături demonstrate în [9].

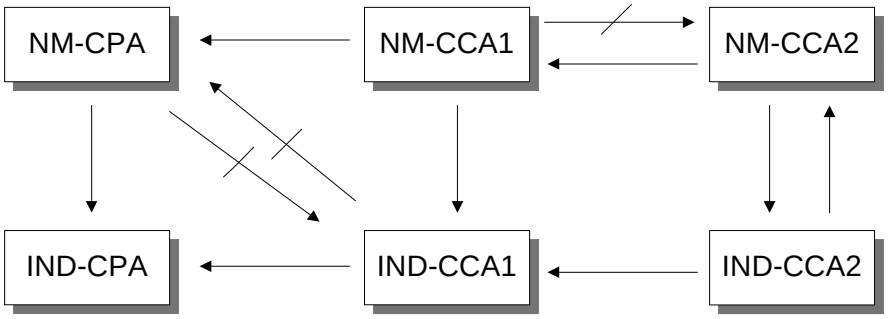


Figura 3.2. Relaţii între noţiuni de securitate

Aici simbolul $A \leftarrow B$ indică faptul că asigurarea obiectivului *B* garantează asigurarea obiectivului *A*. Cel mai esenţial aspect în aceste relaţii este echivalenţa IND-CCA2 $\Leftrightarrow$ NM-CCA2 care înseamnă că un criptosistem pentru care un adversar cu acces adaptiv la maşina de criptare nu poate asocia un criptotext unui mesaj, chiar dacă i se oferă mesajul original şi un alt mesaj, cu o probabilitate mai mare de ½ atunci acest criptosistem este şi non-maleabil, adică un adversar având un

criptotext nu poate construi un alt criptotext astfel încât mesajele aferente să aibă vreo legătură cunoscută de adversar.

Din punct de vedere formal, rezistenţa IND-CCA2 poate fi definită după cum urmează:

Definiţia 3.7. (Rezistenţa IND-CPA, IND-CCA1, IND-CCA2) Fie criptosistemul $\Xi = \{K, E, D\}$, $A = \{A_1, A_2\}$ un adversar al criptosistemului $at \in \{cpa, cca1, cca2\}$ şi k un parametru de securitate (care reprezintă nivelul de securitate). Definim avantajul IND al adversarului împotriva criptosistemului ca fiind:

$$Adv_{A,\Xi}^{ind-atl}(k) = 2 \cdot \Pr\left[\begin{array}{l}(PK, SK) \leftarrow K(1^k), (x_0, x_1, s) \leftarrow A_1^{O_1}(PK), \\ b \leftarrow (0,1), y \leftarrow E_{PK}(x_b) : A_2^{O_2}(x_0, x_1, s, y) = b\end{array}\right] - 1$$

Având următoarele instanţe pentru *at*, O_1 şi O_2:

- Dacă *at=cpa* atunci $O_1(\cdot) = \perp$ şi $O_2(\cdot) = \perp$
- Dacă *at=cca1* atunci $O_1(\cdot) = D_{sk}(\cdot)$ şi $O_2(\cdot) = \perp$
- Dacă *at=cca2* atunci $O_1(\cdot) = D_{sk}(\cdot)$ şi $O_2(\cdot) = D_{sk}(\cdot)$

Datorită echivalenţei IND-CCA2 $\Leftrightarrow$ NM-CCA2 nu introducem definiţia pentru proprietatea NM (practic pentru criptosisteme odată demonstrată rezistenţa IND-CCA2 aceasta implică şi rezistenţă NM-CCA2).

3.2.3 Soluţii sigure dar ineficiente

Există multe propuneri care ar putea fi discutate în acest context, dar poate că cele mai relevante sunt propunerile lui Goldwasser şi Micali care ca pionieri în criptografie au reuşit să propună o schemă de criptare cu cheie publică care să dispună de proprietăţi de securitate necunoscute până atunci (proprietatea IND) precum şi o semnătură digitală a cărei securitate este demonstrabilă ca fiind echivalentă cu factorizarea (semnătura bazată pe permutări Claw-Free) [42], [43]. Nu este relevant să detaliem toate aceste propuneri, scopul este doar de a induce ideea lipsei de eficienţă şi pentru aceasta vom prezenta succint doar schema de criptare asimetrică introdusă de Goldwasser şi Micali.

Acest criptosistem este esenţial ca şi concept şi principii introduse în securitate. Sistemul este un sistem criptografic non-determinist, ceea ce înseamnă că rezultatul are o valoare aleatoare, acelaşi mesaj criptat de mai multe ori va rezulta în criptotexte diferite. Principiul pe care se bazează criptosistemul este imposibilitatea de a distinge reziduurile de non-reziduurile cvadratice din Z_n^* fără a cunoaşte factorizarea lui n - acest lucru mai se numeşte şi problema reziduurilor cvadratice (QRP – Quadratic Residuosity Problem). Descrierea algoritmului de criptare asimetrică Goldwasser-Micali este următoarea:

$GM.Gen\left(1^k\right)$: Generează două numere prime aleatoare *p, q* şi calculează $n=pq$. Alege $y \in Z_n^*$ astfel încât y este un pseudo-reziduu cvadratic, i.e. $y \in \widetilde{Q}_n$. Cheia publică este $PK \leftarrow (n, y)$ iar cea privată este $SK \leftarrow (p, q)$.

$GM.Enc(m, PK)$: Se reprezintă mesajul m în binar ca $m = m_0 m_1 m_2 ... m_t$. Având cheia publică a entităţii aferente $PK \leftarrow (n, y)$, pentru $i = \overline{0,t}$ alege un întreg aleatoriu $x \in Z_n^*$ şi dacă $m_i = 1$ atunci $c_i = yx^2 \bmod n$ altfel dacă $m_i = 0$ atunci $c_i = x^2 \bmod n$. Mesajul criptat este $(c_0 c_1 c_2 ... c_t)$.

$GM.Dec(c, SK)$: Pentru $i = \overline{1,t}$ calculează simbolul Legendre $\left(\frac{c_i}{p}\right)$ şi dacă $\left(\frac{c_i}{p}\right) = 1$ atunci $m_i = 0$ altfel $m_i = 1$.

Securitatea schemei este echivalentă cu problema calculului apartenenţei la mulţimea reziduurilor cvadratice, deci $GM.Dec \Leftrightarrow QRP$ şi criptosistemul este rezistent IND în faţa unor adversari pasivi, lucru elementar de demonstrat. Se poate constata însă uşor că această schemă induce pentru fiecare bit din mesaj o valoare criptată de dimensiunea modulului. Cum un modul RSA are în general dimensiunea câtorva mii de biţi este evident că mesajul criptat va creşte în dimensiune de câteva mii de ori – cea ce înseamnă ineficienţă. Ineficiente au fost multe dintre primele soluţii propuse pentru a atinge obiective de securitate avansate. Soluţii precum cele din secţiunea 3.3 ating obiective de securitate chiar mai solide de atât şi în acelaşi timp nu fac ca mesajul să crească doar cu câteva zeci de biţi.

3.3 Soluţii eficiente bazate pe hibridizarea tehnicilor simetrice şi asimetrice

Subiectul este suficient de amplu şi de interesant pentru a-i dedica o teză în sine. Încercăm însă să expunem pe scurt considerentele care duc la aceste idei şi necesitatea lor.

Principiul de bază este următorul: primitivele asimetrice, fiind bazate pe proprietăţi exotice provenite din teoria numerelor sunt extrem de flexibile putând fi utilizate în scenarii cât se poate de spontane (exemplu clasic, cazul în care se doreşte comunicarea pe un canal nesigur în absenţa unui secret partajat prealabil, acest lucru fiind de fapt şi începutul criptografiei cu cheie publică). Aceleaşi proprietăţi exotice care fac aceste funcţii asimetrice indispensabile, le fac şi uşor de atacat (exemplul clasic, atacurile CCA2 asupra RSA şi ElGamal). În acest sens introducerea în criptosisteme asimetrice a unor primitive simetrice reprezintă o necesitate tocmai pentru a anula anumite proprietăţi algebrice ce pot fi exploatate de adversari şi păstra doar acele proprietăţi care sunt strict necesare. În acest caz vorbim de o combinare a tehnicilor simetrice şi asimetrice la nivel de funcţie

criptografică, câteva astfel de exemple vor fi discutate în secţiuniile următoare. Un exemplu imediat este criptarea RSA aşa cum este ea implementată după standardul PKCS în .NET sau Java, şi care foloseşte sub funcţia RSA o funcţie hash.

A doua situaţie de combinare este utilizarea tehnicilor simetrice pentru a obţine soluţii cu caracteristici asimetrice. Chiar dacă pare contradictoriu acest deziderat, justificarea lui este următoarea: pe de o parte funcţiile din criptografia asimetrică sunt bazate pe elemente de teoria numerelor care le fac mult mai intense din punct de vedere computaţional şi care conduc la întrebări fără răspuns legate de securitatea schemei în cauză (de exemplu întrebarea al cărei răspuns încă nu este cunoscut, dacă RSA este sau nu echivalent cu problema factorizării). Această direcţie este chiar mai veche decât prima şi a fost deschisă probabil de introducerea conceptului de semnătură digitală one-time, semnături care folosesc exclusiv funcţii simetrice pentru a obţine o proprietate asimetrică (generarea cu o cheie şi verificarea cu o altă cheie) [13], [14], [32], [86]. În epoca contemporană însă lucrurile merg mult mai departe de atât şi apar în peisajul criptografiei construcţii, precum bine-cunoscutul protocol TESLA [95], care utilizează doar funcţii simetrice şi reuşesc să ofere proprietăţi tipice schemelor asimetrice, cum ar fi verificarea de mai multe entităţi cu aceeaşi cheie a autenticităţii unei informaţii.

3.3.1 Funcţii eficiente

Această secţiune prezintă câteva soluţii pentru a transforma criptosistemele cu cheie publică în criptosisteme rezistente CCA2. Propunerile generice ale lui Bellare si Rogaway sunt primele propuneri cu securitate demonstrată. Aceste propuneri, pe lângă importanţa istorică, fiind primele propuneri cu securitate demonstrată, sunt esenţiale deoarece şi propuneri ulterioare (de exemplu RSA-KEM din [111]) chiar dacă mult mai elaborate şi aparent mai complicate sunt extrem de apropiate ca tehnică constructivă de mecanismele propuse de Bellare şi Rogaway. În [5] a fost introdusă prima euristică în baza căreia se poate demonstra securitatea în faţa adversarilor activi. Metoda are la bază utilizarea unui model numit modelul oracolului aleator ROM (Random Oracle Model) şi are ca bază simularea comportamentului funcţiilor hash ca funcţii perfect aleatoare (altfel spus modelul presupune că un potenţial adversar nu poate face diferenţa între ieşirea unei funcţii aleatoare şi ieşirea unei funcţii hash). Modelul ROM a adus şi criticism din partea unor nume puternice în domeniu [21] dar în cele din urmă este singura metodă la momentul actual de a demonstra securitatea unui criptosistem. Respingerea acestui model are la bază faptul că desigur, în cele din urmă, o funcţie hash nu este o funcţie aleatoare, deci Oracole Aleatoare nu există în lumea reală, dar în cele din urmă modelul este acceptat ca fiind cel puţin un compromis, şi reprezintă cel puţin un test pentru criptosisteme, adică un criptosistem care nu face faţă în ROM nu trebuie pus sub nici o formă în practică în timp ce un criptosistem care rezistă în ROM are şanse bune ca şi în practică să nu poată fi eficient atacat. Totuşi trebuie menţionat că există criptosisteme care pot fi demonstrate ca fiind sigure în ROM şi totuşi pot fi sparte – deci problema de bază a modelului ROM este incompletitudinea. Două tehnici de criptare sunt introduse în [5]: criptarea $E(x) = f(r) \,||\, G(r) \oplus x$ care are securitate IND în faţa adversarilor CCA1 şi criptarea $E(x) = f(r) \,||\, G(r) \oplus x \,||\, H(rx)$ are securitate IND şi NM în faţa adversarilor CCA2 (la data publicării lucrării echivalenţa IND-CCA2 $\Leftrightarrow$ NM-CCA2 nu era încă demonstrată aşa că în [5] se găsesc demonstraţii separate pentru cele două proprietăţi de securitate). Aici G este un generator de numere aleatoare, H este o

funcţie hash, iar f este o funcţie de criptare oarecare (în practică G şi H pot fi implementate cu o funcţie hash datorită comportamentului funcţiilor hash similar cu al funcţiilor aleatoare). Notaţia $\oplus$ reprezintă funcţia booleană XOR iar notaţia || desemnează concatenare.

Nu este de mirare, că tot Bellare şi Rogaway introduc prima tehnică de criptare pe bază de RSA care este rezistentă CCA2. Aceasta este binecunoscuta tehnică de formatare (padding) a mesajului denumită OAEP (Optimal Asymmetric Encryption Padding) tehnică folosită sub funcţia RSA, ansamblu cunoscut sub numele de RSA-OAEP [6]. În primul rând se fac importante câteva menţiuni istorice cu privire la OAEP. Bellare şi Rogaway au introdus OAEP în [6] susţinând că OAEP este o tehnică de padding care funcţionează pentru orice funcţie one-way trapdoor făcând-o rezistentă în faţa atacurilor CCA2. Ulterior Shoup [110] demonstrează că OAEP nu poate să garanteze acest lucru pentru orice funcţie, aducând un contra-exemplu cu o funcţie XOR-maleabilă. Deficienţa descoperită de Shoup ridică mari semne de întrebare cu privire la RSA-OAEP şi sunt Fujisaki, Okamoto, Pointcheval şi Stern [40] cei care reuşesc să demonstreze că RSA-OAEP este rezistentă CCA2. În concluzie RSA-OAEP este un mecanism eficient şi sigur de criptare folosind RSA. Continuăm cu descrierea acestui criptosistem. Funcţia de criptare *f-OAEP* defineşte criptarea ca: $E(x) = f\left(x \oplus G(r) \,||\, r \oplus H\left(x \oplus G(r)\right)\right)$. Se poate observa că este vorba de fapt de includerea unei reţele Feistel sub o funcţie one-way cu trapă.

OAEP este deci o tehnică de padding a cărei securitate este demonstrată la nivelul utilizării funcţiei cu trapă RSA. Desigur există însă şi alte funcţii cu trapă decât RSA-ul. În acest context dezvoltarea unor metode cât mai variate de padding a devenit necesară. Probabil cea mai bună propunere este cea a lui Fujisaki şi Okamoto. Pe scurt propunerea acestora este următoarea: criptarea $E(x) = E_{pk}\left((x \,||\, r) \,||\, H(x,r)\right)$, unde r este o valoare aleatoare, H este o funcţie hash. În modelul ROM Fujisaki şi Okamoto au demonstrat că o astfel de criptare este rezistentă IND-CCA2 cu condiţia ca funcţia de criptare să fie sigură IND-CPA. Între exemplele oferite de autori se află aplicarea unui astfel de padding asupra schemelor Blum-Goldwasser, ElGamal şi Okamoto-Uchiyama [39]. Această schemă de padding mai este cunoscută şi sub numele de Enhanced Probabilistic Encryption.

Trebuie amintite şi criptosistemele de criptare hibridă rezistente IND/NM-CCA2 introduse de Shoup şi Cramer în [25]. O soluţie analoagă pentru construcţia de criptosisteme hibride rezistente CCA2 este în [76] (mecanismul de încapsulare propus în [76] a fost identificat ca fiind nesigur în [69]).

3.3.2 Protocoale eficiente

În contextul protocoalelor, noţiunea de eficienţă a intrat în calcul relativ recent. Există multe protocoale utilizate în practică a căror eficienţă computaţională şi nivel de securitate lasă de dorit, un bun exemplu este NTLM (NT LAN Manager) unul dintre cele mai frecvent folosite protocoale de autentificare prezent în sistemele de operare Windows, toate generaţiile, inclusiv XP şi Vista [64]. Problema eficienţei a fost luată în calcul de îndată ce s-a pus problema utilizării protocoalelor în medii constrânse unde se doreau proprietăţi avansate de securitate şi consumuri scăzute de resurse. În acest sens sunt extrem de relevante protocoalele de autentificare a informaţiei din familia protocolului TESLA propuse pentru utilizare inclusiv în reţele de senzori. Datorită legăturii solide cu protocoalele aduse ca şi contribuţii în această teză vom aduce aici o scurtă trecere în revistă a acestuia precum şi a protocolului

CSA care utilizează de asemenea lanţuri one-way dar într-o manieră diferită. Trebuie subliniat că toate aceste protocoale corespund de asemenea conceptului de construcţie hibridă între tehnicile simetrice şi asimetrice deoarece folosesc funcţii criptografice simetrice şi totuşi ating o proprietate asimetrică, anume absenţa secretelor partajate.

- **Protocolul TESLA**

Protocolul TESLA Timed Efficient Stream Loss-tolerant Authentication este un protocol de autentificare a informaţiei bazat pe lanţuri one-way si sincronizare temporală slabă. Propunerea îi aparţine lui Perrig et al. şi există mai multe variante ale acestui protocol precum şi câteva propuneri de utilizare în diverse zone [92], [93], [94], [95], [96]. Toate propunerile făcute de Perrig se bazează pe necesitatea unei sincronizări temporale între emiţători şi receptori. Subliniem în acest sens condiţia de securitate pe care se bazează toate schemele propuse de Perrig et al. (valabilă pentru toate protocoalele din [92], [93], [94], [95], [96]): un pachet P_i este recepţionat în condiţii de siguranţă dacă şi numai dacă receptorul poate decide, bazat pe sincronizarea temporală, dacă emiţătorul nu a trimis deja pachetul P_j care face publică cheia de autentificare a pachetului P_i.

Schema de bază fără toleranţă la pachete pierdute propusă în [93] nu face uz de elementele unui lanţ one-way, propunerea fiind apropiată ca natură de protocolul Guy Fawkes [1]. Principiul pe care schema funcţionează este următorul: emiţătorul trimite o garanţie pentru o cheie generată aleator şi păstrată secretă, această cheie este utilizată pentru a calcula un MAC asupra pachetului P_i, şi va fi făcută publică în pachetul P_{i+1} când receptorul va putea verifica MAC-ul din pachetul P_i. Dacă MAC-ul este verificat cu succes atunci P_i este autentic. Această schemă poate fi spartă dacă un adversar intră în posesia pachetului P_{i+1} înainte ca receptorul să fi primit P_i - în acest caz adversarul este în posesia cheii K_i secrete cu care a fost calculat MAC-ul şi traficul poate fi fraudat pentru fiecare pachet nou venit. Pentru a înlătura acest atac emiţătorul trebuie să verifice condiţia de securitate folosind sincronizarea temporală. Pentru aceasta, receptorul trebuie să cunoască programul de transmitere al pachetelor iar cel mai simplu mod de a realiza acest lucru este prin utilizarea unei rate de transfer constante.

Introducerea lanţurilor one-way este soluţia pentru a face schema tolerantă la pachete pierdute. De ce lanţurile one-way oferă o soluţie în acest sens? Explicaţia este simplă: dacă fiecare cheie de sesiune este element al unui lanţ one-way atunci chiar dacă un număr oarecare de chei este pierdut ele pot fi recuperate din cheia curentă prin compoziţia succesivă a funcţiei one-way; în timp ce din cheia curentă nu pot fi aflate cheile următoare deoarece lanţul este ireversibil. O astfel de schemă este tolerantă în principiu la orice număr de pachete pierdute. În detaliu, ideea este de a alege aleator o valoare K_n şi de a genera secvenţa de n valori $K_i = F^{n-i}(K_n)$, $i = \overline{0,n}$ această secvenţă (lanţ one-way) urmând să se numească lanţ de chei. Restul schemei rămâne nemodificată. Este de asemenea important de observat că acum se poate renunţa la garantarea cheii în fiecare pachet, i.e. valoarea lui $F(K_i)$, deoarece oricare valoare a lanţului este în fapt o garanţie pentru întregul lanţ (de fapt este suficientă garantarea lui $K_0 = F^n(K_n)$ pentru autenticitatea întregului lanţ).

Creşterea ratei de transfer poate fi de asemenea realizată. Condiţia de securitate aduce şi o limitare: pachetul P_{i+1} poate fi trimis doar la momentul la care toţi receptorii au primit deja P_i - aceasta, în mod evident, duce la limitarea ratei de transfer. Problema poate fi elegant soluţionată prin divulgarea cheii K_i nu în pachetul P_{i+1} ci în pachetul P_{i+d} unde d este un parametru fixat la începutul sesiunilor de comunicare ce reprezintă întârzierea autentificării. Este important de reţinut: alegerea lui d nu afectează securitatea schemei propuse ci doar utilizarea ei. Astfel alegerea unei valori mari pentru d duce la întârzieri mari de autentificare, in timp ce alegerea de valori mici pentru d duce la pierderea unor pachetelor pentru receptorii la care viteza de transfer este scăzuta şi care nu primesc pachetele în timp util.

Schemele anterioare impuneau ca emiţătorul sa trimită pachete la momente fixe de timp, lucru care poate duce la limitarea scalabilităţii soluţiei. Pentru aceasta protocolul poate fi modificat pentru a permite trimiterea unui număr arbitrar de pachete în fiecare interval de timp, ceea ce înseamnă rate dinamice pentru pachete, astfel aceeaşi cheie K_i este utilizată pentru a garanta autenticitatea tuturor pachetelor trimise într-un anumit interval de timp de index i. În acest caz indexul i al intervalului de care aparţine pachetul primit la momentul t poate fi calculat ca: $i = \left\lfloor \frac{t - T_0}{T_\Delta} \right\rfloor$. În relaţie T_0 este momentul de început al primului interval iar T_Δ este durata comună a intervalelor.

Problema autentificării imediate în TESLA este abordată abia în [95]. Faptul că autentificarea este obţinută cu întârziere, deoarece cheia utilizată pentru MAC este făcută publică doar într-o sesiune ulterioară, duce la faptul că receptorul trebuie să stocheze pachetele în vederea autentificării. Pentru a evita acest fapt o soluţie de autentificare imediată poate fi uşor construită prin garantarea mesajului înainte de a fi trimis: în acest caz în momentul în care mesajul este primit autenticitatea sa poate fi direct verificată pe baza autenticităţii garanţiei anterioare. Desigur însă dezavantajul acestei abordări este faptul că mesajul trebuie cunoscut cu o sesiune înainte de a fi efectiv transmis.

- **Protocolul CSA**

Propunerea Chained Stream Authentication este un protocol de autentificare bazat pe lanţuri one-way fără sincronizare temporală şi care are la bază un mecanism de tip challenge-response. Astfel în loc de a necesita o sincronizare temporală pentru a decide dacă intervalul de timp în care a ajuns pachetul este corect, acest protocol necesită o confirmare care la rândul ei este tot un element al unui lanţ de tip one-way. Rădăcini mai vechi ale acestui protocol de autentificare pot fi de asemenea găsite în protocolul Guy Fawkes [1]. În [10] se găsesc trei propuneri distincte de autentificare CSA ce vor fi descrise în continuare.

Protocolul Interactive CSA I-CSA este o variantă a protocolului CSA dedicată schimbului de informaţie între două entităţi [10]. Protocolul DeMA [50] propus de autor în capitolul 4 este apropiat ca natură de CSA, diferenţa este la nivelul funcţiei one-way pentru construcţia lanţului, care la DeMA este o funcţie one-way oarecare lucru care va face posibilă utilizarea funcţiei putere discretă în varianta DeMa-QR, precum şi la nivelul iniţializării lanţurilor one-way care la DeMA se realizează prin

protocolul introdus de autor şi denumit DiMA (ansamblu final denumit DeMA/DiCA). Pentru protocolul CSA iniţializarea lanţurilor one-way se face folosind o semnătura digitală aplicată după cum urmează:

Sesiunea de iniţializare la I-CSA

$$B \to A:\ h^k(\beta), SN, Sig\left(SK_B, h^k(\beta), SN\right)$$

$$A \to B:\ A_1, MAC_{h^{k-1}(\alpha)}(A_1), h^k(\alpha), SN, Sig\left(SK_A, h^k(\alpha), SN\right)$$

Notaţiile au următoarea semnificaţie: *A* şi *B* sunt cele două entităţi, *Sig* este o semnătură digitală, *h* este o funcţie hash, *k* este dimensiunea lanţului one-way, α, β sunt valori de iniţializare ale lanţurilor one-way, *SN* este un nonce iar A_1 este mesajul din sesiunea 1. Sesiunea de iniţializare conţine o modificare faţă de o propunerea iniţială pentru a evita un atac de tip man-in-the-middle propus în [95]. Schimbul de mesaje la CSA se desfăşoară prin mecanismul descris în continuare:

Sesiunea de comunicare *i* la I-CSA

$$B \to A:\ h^{k-i+1}(\beta)$$

$$A \to B:\ A_i, MAC_{h^{k-i}(\alpha)}(A_i), h^{k-i+1}(\alpha)$$

Dorinţa autorilor din [10] a fost şi de a construi un protocol dedicat autentificării broadcastului de informaţie, pentru aceasta a fost propusă varianta N-party I-CSA. Desigur că o primă variantă în atingerea acestui deziderat ar fi aplicarea directă a protocolului I-CSA între un emiţător si mai mulţi receptori cu un singur lanţ de partea emiţătorului. Din păcate această soluţie înregistrează o deficienţă majoră: în cazul în care I-CSA este aplicat între un emiţător si mai mulţi receptori, comunicaţia nu poate continua până când nu s-a primit un răspuns din partea tuturor receptorilor. Altfel spus dacă unul dintre receptori dispare de pe scenă, comunicaţia încetează pentru toţi ceilalţi receptori şi mai mult dacă răspunsurile unui receptor vin cu întârzieri comunicaţia cu ceilalţi receptori se va desfăşura cu aceleaşi întârzieri.

Soluţia propusă pentru eliminarea acestui dezavantaj este ceea ce autorii numesc „autentificare întârziată" („delayed authentication"): prin aceasta sunt întâi trimise pachetele de informaţie pentru a fi autentificate mai târziu în momentul în care toţi receptorii sunt capabili să răspundă şi cheile de autentificare pot fi trimise. Rămâne însă o întrebare: cine este dispus să stocheze informaţie neautentificată, în ideea că la un moment dat cineva o va autentifica? Această vulnerabilitate poate duce la atacuri de tip Denial of Sevices prin epuizarea spaţiului de stocare a informaţiei. Suntem sceptici în aplicabilitatea practică a acestei propuneri, de altfel partea experimentală din [10] evită implementarea acestei soluţii de autentificare.

Varianta Timed Chained Stream Authentication T-CSA este un protocol de autentificare bazat pe lanţuri one-way şi sincronizare temporală, este uşor de intuit că această variantă este similară cu TESLA. În propunerea din [10] nu există nici o diferenţă semnificativă între T-CSA şi TESLA drept pentru care prezentarea detaliilor nu mai este utilă. Suplimentar în [10] este propusă o aplicaţie a acestui protocol în broadcastul autentic de informaţie audio. Aplicaţia MAT (Multicast Authentication Tool) este construită peste RAT (Robust Audio Tool) pentru a oferi transmisie autentică de informaţie audio folosind T-CSA.

3.4 Considerente de implementare

Datorită relevanţei primitivelor criptografice orice limbaj modern dispune de biblioteci cu implementări ale acestora. De exemplu mediile .NET şi Java dispun de implementări ale funcţiilor criptografice cum ar fi funcţii hash, criptări simetrice, coduri MAC, criptări asimetrice semnături digitale. Încercăm în acest capitol să facem doar o scurtă enumerare a funcţiilor frecvent utilizate în practică şi le vom puncta pe cele care pot fi regăsite în .NET – deoarece acest mediu a fost utilizat pentru dezvoltarea aplicativă din capitolul 5. Nu se va face o enumerare similară pentru funcţiile din Java deoarece aici arsenalul de funcţii criptografice este mult mai mare fiind vorba de un limbaj open source pentru care există comunităţi masive de programatori. În principiu în Java există toate funcţiile care există în .NET şi în plus multe altele.

- **Funcţii Hash**

Cea mai utilizată gamă de funcţii hash este familia SHA (Secure Hash Algorithm) [36] pentru care dimensiunea ieşirii este 224, 256, 384, 512 biţi indiferent de dimensiunea datelor de intrare. Acestea se mai numesc generic şi familia SHA-2, varianta mai veche de funcţie din familia SHA, şi încă folosită, este SHA-1. Menţionăm că atacuri recent publicate [119] arată că funcţiile hash MD5 şi SHA-1 nu pot garanta rezistenţa secundară a imaginii dar în ciuda acestui fapt ele sunt încă folosite în multe aplicaţii. O discuţie relevantă cu privire la ce implicaţii are pierderea rezistenţei secundare a imaginii (de remarcat că nu orice categorie de aplicaţie este pusă în pericol de aceasta) poate fi găsită în [80]. Atacuri asupra SHA au fost anunţate pentru prima oară în [120], un articol non-tehnic al lui Schneier cu privire la atacurile asupra funcţiilor hash este în [106]. Pentru soluţii contemporane se recomandă folosirea SHA-256 sau alte funcţii din familia SHA-2 mai puternic şi nicidecum a MD5 sau SHA-1. Recent a fost lansat şi concursul pentru SHA-3. În .NET se găsesc implementate funcţiile: MD5, SHA1, SHA256, SHA384, SHA 512.

- **Coduri MAC**

Codurile MAC se construiesc pe baza unei funcţii hash, frecvent se foloseşte MD5 sau SHA1, cu toate că ambele au un nivel de securitate destul de scăzut, drept pentru care se recomandă folosirea funcţiilor hash din familia SHA-2. Rolul codurilor MAC este de a testa autenticitatea unei informaţii, deci pentru a verifica sursa de provenienţă a informaţiei, implicând astfel şi o garanţie asupra integrităţii. În

practică se folosesc construcţiile, unanim recunoscute ca eficienţă şi securitate, HMAC şi NMAC propuse de Bellare, Canetti şi Krawczyk în lucrarea [7]. În .NET se găsesc implementate coduri MAC de tip HMAC disponibile cu toate variantele de hash anterior amintite.

- **Funcţii de criptare simetrică**

În prezent există o mare varietate de algoritmi simetrici de criptare. Standardul valabil până în 2001 a fost DES (Data Encryption Standard) [35], la momentul de faţă acest algoritm neoferind securitate (poate fi spart în câteva ore pe un calculator actual). DES este un cod construit pe reţea Feistel care transformă mesaje de 64 de biţi în criptotext de 64 de biţi, cheia DES are însă doar 56 de biţi (deci nivelul de securitate este de 56 de biţi şi nu de 64). În prezent DES supravieţuieşte sub forma 3DES (recomandat încă din 1999), care constă în aplicarea transformării DES de 3 ori şi este pe 128 de biţi oferind un nivel de securitate suficient de bun în zilele de azi. La nivelul anilor 2001 DES nu mai oferă securitatea necesară (de fapt încă din anii 90 sunt consemnate atacuri soldate cu succes asupra DES), pentru care, pe bază de concurs se alege un nou standard AES (Advanced Encryption Standard). Standardul curent este candidatul la AES numit Rijndael [37] ales din cei 5 finalişti: Rijndael, Serpent, Twofish, RC6 şi MARS. AES este un cod bloc disponibil în trei variante de dimensiuni pentru cheie 128, 192, 256; chiar şi cheia de 128 de biţi este considerată destul de sigură pentru cerinţele din ziua de azi. Necesită doar 10-14 runde în funcţie de dimensiunea cheii, este sigur şi este cel mai rapid dintre candidaţi. Deoarece AES este mai rapid decât alte coduri simetrice, chiar şi decât 3DES, şi oferă acelaşi nivel de securitate nu există nici un motiv de a utiliza altceva decât AES în arhitecturi de securitate contemporane. Totuşi trebuie să precizăm că singura suspiciune cu privire la securitatea AES-ului este faptul că foloseşte un design destul de non-conformist, spre deosebire de sistemele simetrice clasice, care se construiesc pe reţea Feistel, şi acest design nu a fost sub atenţia comunităţii criptologilor decât în ultimii ani, de la propunerea AES-ului. În principiu o reţea Feistel este o reţea bine studiată şi un cifru construit pe reţea Feistel este puţin probabil să aducă surprize în ceea ce priveşte securitatea. Acesta este singurul dezavantaj al AES, faptul că are un design exotic comparativ cu toate celelalte criptosisteme simetrice, transformarea Rijndael fiind echivalentă cu o ecuaţie algebrică faţă de care există suspiciunea că ar putea duce în viitor la o serie de atacuri. Descrierea criptosistemului se găseşte în [37]. În .NET se găsesc implementate variantele RC2, DES, 3DES şi Rijndael (AES).

- **Funcţii de criptare asimetrică**

În prezent există o paletă largă de funcţii de criptare cu cheie publică, numită şi criptare asimetrică. Acestea pot fi împărţite în două mari categorii după cele două mari probleme pe care se bazează (această clasificare nu este exhaustivă deoarece există şi alte probleme pe care se pot construi criptosisteme cu cheie publică, doar că acestea nu prezintă eficienţa necesară în practică): a) criptosisteme cu cheie publică bazate pe dificultatea factorizării întregilor (au ca punct de plecare algoritmul RSA [101]), b) criptosisteme cu cheie publică bazate pe dificultatea logaritmului discret (au ca punct de plecare schimbul de cheie Diffie-Helmann [26] şi semnătura digitală ElGamal [31] precum şi extensiile acestora peste grupurile formate de curbe eliptice propuse de Koblitz şi Miller [87]). În .NET este implementată criptarea RSA.

- **Semnături digitale**

În principiu, orice algoritm asimetric poate fi utilizat pentru crearea unei semnături digitale prin inversarea rolului cheii publice cu cea privată, iar primele propuneri de semnături digitale se găsesc în lucrările lui Rivest, Rabin şi ElGamal [99], [101], [31]. Subliniem că şi folosind algoritmi simetrici se pot crea semnături digitale de tip one-time dar acestea sunt rar utilizate în practică şi nu sunt relevante în contextul prezentei lucrări. În .NET sunt disponibile semnăturile RSA şi DSA.

- **Protocoale criptografice**

Există diverse biblioteci disponibile pentru implementarea unor protocoale standard precum SSL, Kerberos, NTLM etc. În medii de programare precum .NET sau Java nu există însă implementări pentru acestea. De asemenea cantitatea de propuneri de protocoale din lucrările de specialitate este enormă şi în practică se găsesc implementate doar o mică parte din acestea.

4 Contribuţii în criptografie

4.1 Autentificarea entităţilor

Protocoalele de autentificare a entităţilor, sau de identificare, sunt cele mai frecvent utilizate protocoale de securitate, ele fiind folosite în diverse aplicaţii practice din zone precum telefonia mobilă, industria bancară, sisteme de operare etc. Multe operaţiuni simple efectuate zi de zi au în spate un mecanism de identificare, câteva exemple sunt extragerea banilor dintr-un bancomat, introducerea cardului în telefonul mobil, accesul la un sistem de operare, accesarea unei imprimante legate la un alt computer din reţea etc. Procesul de autentificare al unei entităţi este în general legat de existenţa unui secret, pe care în limbaj comun îl numim parolă iar din punct de vedere ştiinţific poartă numele de cheie secretă sau privată, care se foloseşte pentru a demonstra identitatea entităţii care îl cunoaşte. Din punct de vedere al construcţiilor criptografice care stau la baza lor există trei mari clase de protocoale de autentificare:

1) Autentificări bazate pe parole (sau autentificări slabe) – sunt autentificările bazate pe parole (password-based) şi care oferă cel mai scăzut nivel de securitate.

2) Autentificări bazate pe parole de unică folosinţă (one-time passwords) – sunt o extensie naturală a autentificărilor bazate pe parole bazate pe restricţionarea validităţii unei parole la o singură utilizare.

3) Autentificări challenge-response (sau autentificări puternice) – sunt autentificări bazate pe primitive criptografice simetrice sau asimetrice în care autentificarea se face într-o manieră interactivă prin adresarea unei provocări (challenge) la care entitatea în cauză trebuie să ofere răspunsul corect (response).

4) Autentificări zero-knowledge – sunt cele mai solide protocoale de autentificare din punct de vedere al criptografiei, dar în general au şi cerinţe computaţionale (folosesc funcţii criptografice intense din punct de vedere computaţional) şi de comunicare (necesită un număr mai mare de runde) mai ridicate. Avantajul care îl prezintă în faţa tehnicilor de autentificare challenge-response este faptul că fiecare autentificare nu aduce nici o informaţie suplimentară asupra secretului în baza căruia se face autentificarea (prin absenţa acestei proprietăţi autentificările challenge-response fac posibilă iniţierea unor atacuri de tip chosen-ciphertext).

În cele ce urmează va fi urmărită autentificarea bazată pe parole one-time folosind schema Lamport care va fi extinsă prin contribuţiile autorului şi de asemenea această schemă va constitui baza unor protocoale eficiente de autentificare a informaţiei descrise începând din secţiunea 4.5. Schema Lamport poate de asemenea fi văzută ca un protocol de tip challenge-response. Contribuţia autorului este de a propune utilizarea funcţiilor peste grupuri de întregi utilizate în

criptografia asimetrică şi ilustrarea avantajelor dar şi a dezavantajelor introduse raportat la funcţiile one-way din criptografia simetrică.

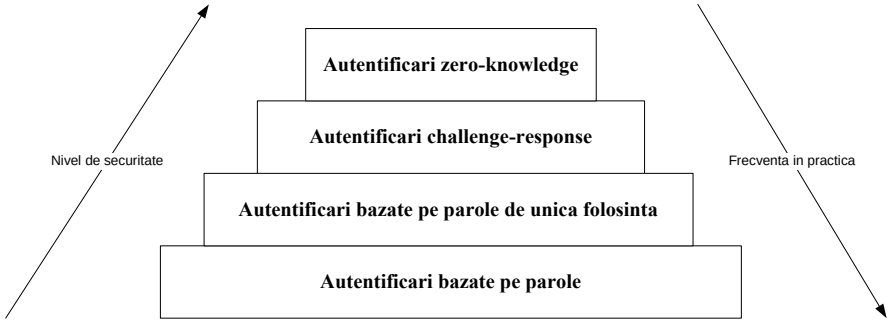


Figura 4.1. Ierarhie a mecanismelor de autentificare a entităţilor

4.1.1 Extensii ale schemei Lamport folosind funcţia putere discretă

Cu toate că în viaţa de zi cu zi parolele sunt frecvent utilizate, autentificarea bazată pe parole prezintă câteva dezavantaje, cel mai important dintre acestea fiind faptul că o parolă interceptată poate fi folosită în mod fraudulos de un potenţial adversar. Pentru a elimina aceste dezavantaje conceptul de parolă de unică folosinţă (one-time passwords) este o bună alternativă. Aceste parole sunt valabile doar pentru o singură utilizare şi în cazul în care o parolă deja folosită este interceptată ea nu poate fi folosită ulterior de un posibil adversar nemaifiind valabilă. Există multe propuneri în acest sens, unele s-au dovedit a fi nesigure [117] iar altele [23] au rămas încă fără vulnerabilităţi cunoscute.

Schema propusă de Lamport este prima soluţie care răspunde la această necesitate. Mai mult decât atât, schema propusă de Lamport posedă şi avantajul că necesită stocarea de secrete doar de partea utilizatorului şi nu şi a sistemului. Autentificarea folosind schema Lamport reprezintă astfel o îmbunătăţire substanţială a autentificărilor slabe deoarece nu utilizează secrete partajate de doi utilizatori şi mai mult decât atât, pierderea unei parole nu duce la impersonarea entităţii în cauză. Pentru autentificarea unui utilizator către un sistem cu schema Lamport se calculează secvenţa $\{x, f(x), f^1(x), f^2(x), ..., f^{N_A}(x)\}$, unde x reprezintă o valoare secretă aleasă de utilizator, f este o funcţie one-way prestabilită iar N_A este numărul maxim de autentificări care poate fi efectuat. Această secvenţă mai poartă şi numele de lanţ one-way (one-way chain) şi fiecare valoare poate fi utilizată ca o parolă. Astfel, într-o fază de iniţializare valoarea lui $f^{N_A}(x)$ este făcută cunoscută sistemului, iar apoi pentru a i-a autentificare $f^{N_A-i}(x)$ este utilizată drept parolă, urmând ca sistemul să verifice că într-adevăr $f(f^{N_A-i}(x)) = f^{N_A-i+1}(x)$ unde $f^{N_A-i+1}(x)$ este ultima parolă corectă primită prin calcularea funcţiei f asupra parolei nou primite. Figura 4.2 ilustrează acest concept. O implementare practică a acestui

sistem de autentificare a fost făcută de Haller în sistemul S-Key [66], [67]. Trebuie amintit însă că acest sistem nu este sigur [88], [89].

- **Utilizarea funcţiei putere discretă**

Desigur, pentru implementarea unui astfel de sistem de autentificare este în primul rând necesară o funcţie one-way. Datorită simplităţii lor, funcţiile hash criptografice sunt în general utilizate în acest scop. Totuşi utilizarea funcţiilor hash prezintă şi un dezavantaj: numărul de autentificări care se poate face este fix, astfel după epuizarea celor N_A parole (elemente din lanţul one-way) nu mai pot fi generate noi parole (secvenţa fiind one-way). Desigur că în practică acest lucru poate fi extrem de dezavantajos deoarece numărul de autentificări este greu de prezis. Mult mai convenabil în practică este însă alegerea unui margini superioare B_{U,N_A} pentru numărul de autentificări care vor fi făcute, o astfel de margine superioară fiind uşor de estimat ca $B_{U,N_A} < \frac{T}{T_A}$ unde T este orizontul de timp în care va funcţiona protocolul de autentificare iar T_A este durata unei autentificări. O valoare mare a marginii superioare B_{U,N_A} devine însă un dezavantaj pentru funcţiile hash deoarece este necesară efectuarea a B_{U,N_A} compoziţii a funcţiei hash pentru obţinerea parolelor.

Prin utilizarea ridicării la putere în Z_n acest dezavantaj însă dispare. Astfel putem folosi funcţia putere discretă:

$$f(x) = x^{\varepsilon} \bmod n, \varepsilon \in Z_n \qquad (4.1)$$

unde $n = p \cdot q$ este produsul a două numere prime iar ε este un exponent întreg. Compoziţia succesivă a acestei funcţii este uşor de calculat deoarece în Z_n exponenţii pot fi reduşi modulo $\phi(n)$ [1] şi următoarea relaţie poate fi folosită:

$$f^{\eta}(x) = x^{\varepsilon^{\eta} \bmod \phi(n)} \bmod n \qquad (4.2)$$

Astfel, în mod analog funcţiilor hash, această funcţie poate fi folosită în protocolul de autentificare. Rezultatul este sintetizat în figura 4.3.

---

[1] $\phi(n)$ este funcţia Euler phi, şi poate fi calculată dacă şi numai dacă se cunoaşte factorizarea $n = \prod_{i=1}^{r} p_i^{e_i}$ în baza relaţiei $\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)...\left(1 - \frac{1}{p_r}\right)$.

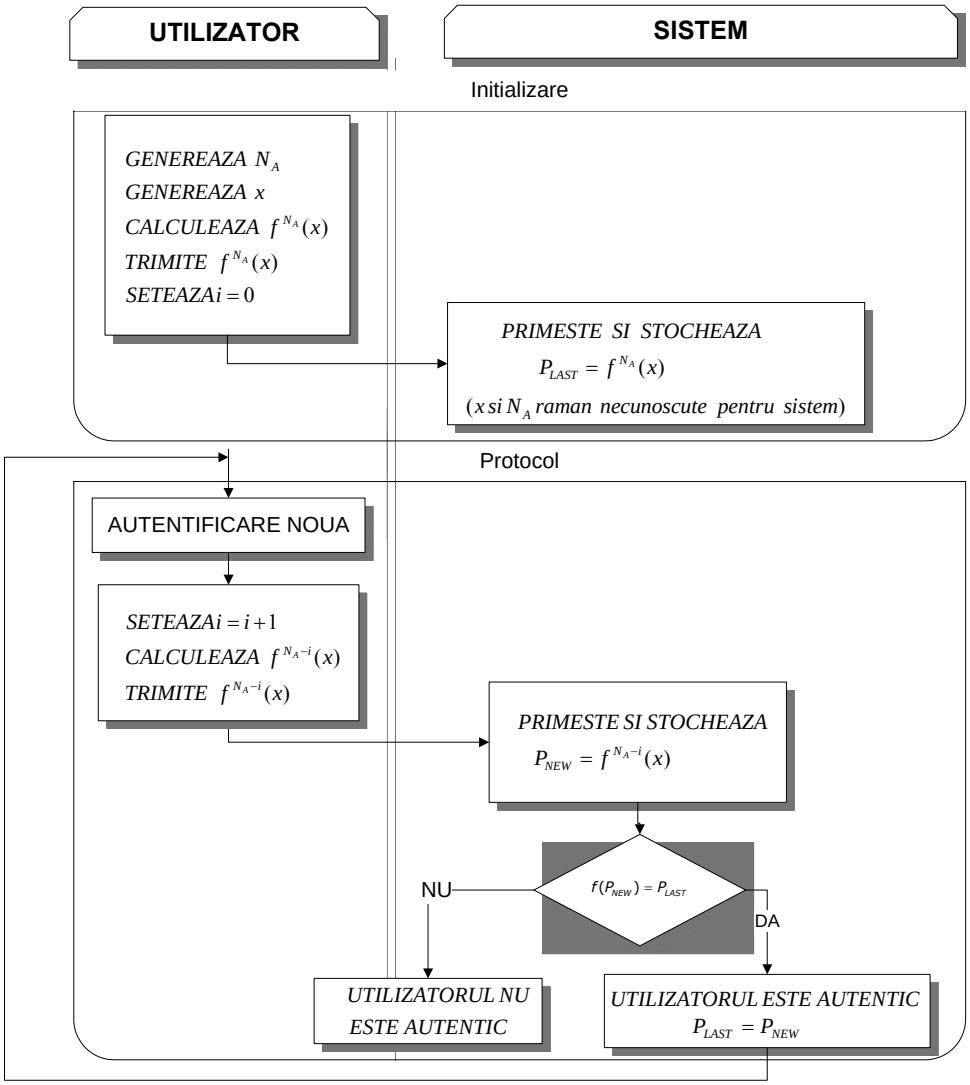


Figura 4.2. Autentificare cu schema Lamport

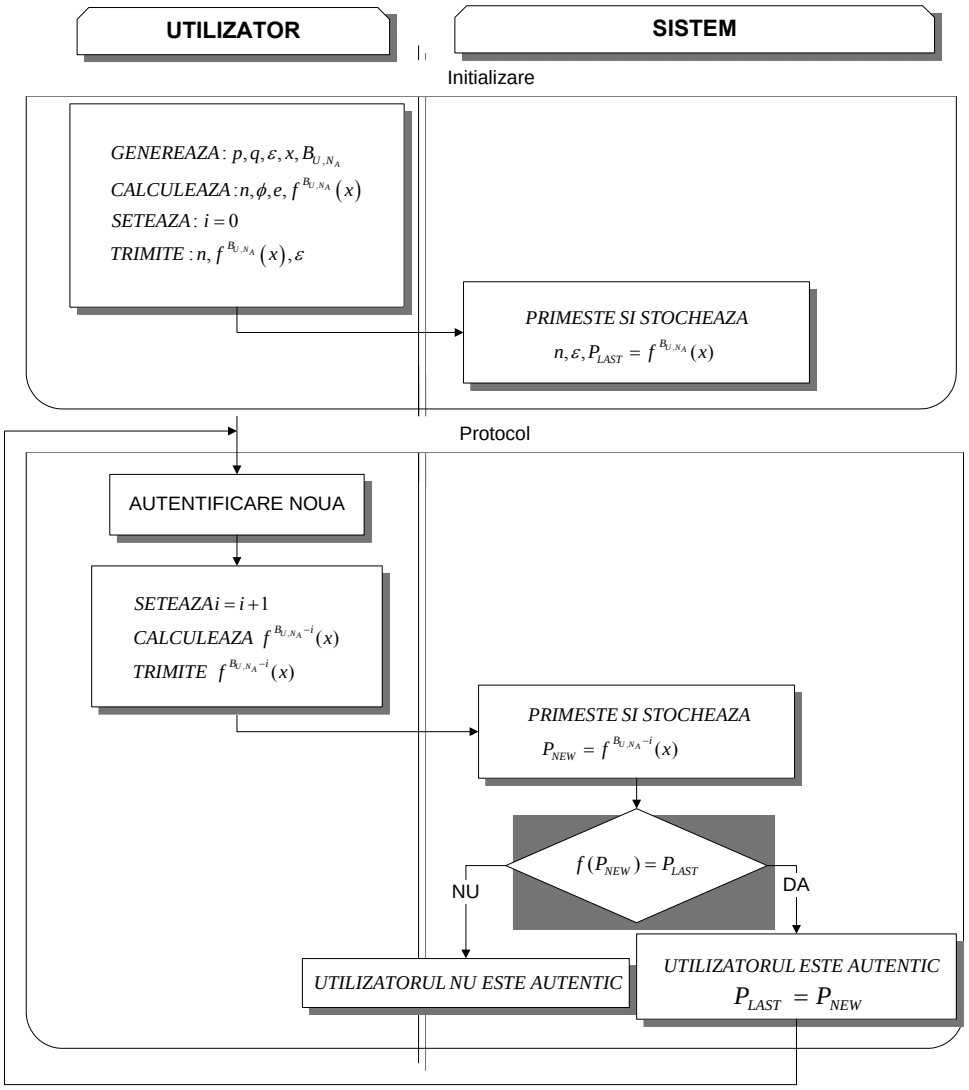


Figura 4.3. Autentificare cu schema Lamport folosind funcţia putere discretă

- **Complexitatea computaţională în cazul utilizării funcţiei putere discretă**

Problema care apare în cazul utilizării funcţiilor definite pe Z_n constă în faptul că valorile sunt mai greu de calculat decât funcţiile hash. Pentru moment dorim să stabilim ordinul de complexitate pentru calculul unei singure parole folosind exponenţierea în Z_n definită de relaţia (4.1). Pentru aspecte generale cu privire la complexitatea computaţională a operaţiilor din Z_n pot fi consultate lucrările [85],

[112]. În mod evident calculul unei parole, care se face folosind (4.2), necesită în primul rând calculul următorului exponent:

$$e = \varepsilon^{\eta - j} \bmod \phi(n) \tag{4.3}$$

După care parola este calculată folosind relaţia următoare:

$$f^{\eta - j}(x) = x^e \bmod n \tag{4.4}$$

Astfel complexitatea de calcul a exponentului este:

$$O_e = \frac{3}{2}\log_2(\eta - i) \tag{4.5}$$

multiplicări modulare în $Z_{\phi(n)}$, iar cea de calcul a parolei:

$$O_{x^e} = \frac{3}{2}\log_2(e) \tag{4.6}$$

multiplicări modulare în Z_n.

Desigur prin adunarea relaţiilor (4.5) şi (4.6) se poate obţine complexitatea pentru calculul unei parole ca fiind:

$$O_{f^{\eta - i}} = \frac{3}{2}\log_2(e) + \frac{3}{2}\log_2(\eta - i) \tag{4.7}$$

Deoarece cantitatea în $\frac{3}{2}\log_2(\eta - i)$ este neglijabilă faţă de $\frac{3}{2}\log_2(e)$ putem aproxima complexitatea de calcul a unei parole ca fiind:

$$O_{f^{\eta - i}} \approx \frac{3}{2}\log_2(\phi(n)) \tag{4.8}$$

Concluzia în baza complexităţii obţinute este că funcţia în Z_n oferă ca avantaj faptul că nu sunt necesare compoziţii succesive pentru obţinerea parolelor dar are ca dezavantaj faptul că pentru o parolă calculul este mai complex decât în cazul unei simple funcţii hash – mai exact intensitatea computaţională fiind cea a unei exponenţieri modulare.

➢ **Ideea unui schimb timp-memorie pentru lanțuri de reziduuri cvadratice**

Prin înlocuirea exponentului în relaţia (4.1) cu $\varepsilon = 2$ obţinem funcţia ridicare la pătrat modulo n:

$$f(x) = x^2 \bmod n \tag{4.9}$$

Această funcţie este one-way deoarece aşa cum s-a precizat şi în secţiunea de fundamentare teoretică calculul reziduurilor cvadratice este posibil doar dacă se cunoaşte factorizarea lui n. În acest caz calculul unei singure parole din lanţ se poate face printr-o singură multiplicare modulară în cazul în care se cunoaşte parola anterioară deoarece este uşor de observat că pentru a i-a parola avem $f^{\eta-i}(x) = f^{\eta-i-1}(x) \cdot f^{\eta-i-1}(x)$. Astfel problema schimbului de timp-memorie (time-memory trade) se pune în modul următor, în funcţie de spaţiul disponibil de stocare sau de preferinţa utilizatorului se poate stoca un anumit număr de parole, fie acesta N_{PP} - numărul de parole precalculate, iar pentru calculul acestor parole este necesar în primul rând calculul parolei $f^{\eta-N_{PP}} = x^{2^{\eta-N_{PP}}} \bmod n$ urmat de N_{PP} multiplicări modulare. Aceasta conduce în final la un efort computaţional total de:

$$O_{f^{\eta-i}} = \frac{3}{2}\log_2\left(\phi(n)\right) + N_{PP} \tag{4.10}$$

multiplicări modulare. În [47] s-a efectuat un studiu complet asupra eficienţei utilizării lanţurilor de reziduuri cvadratice. Studiul de caz din [47] nu este însă axat în jurul unui exemplu suficient de relevant şi funcţia putere discretă este destul de costisitoare, atât din punct de vedere computaţional cât şi ca şi implementare necesitând lucrul într-un grup de întregi în locul unor simple operaţii binare solicitate de funcţiile hash, pentru a fi utilă în autentificarea unei entităţi proces care are în general o soluţie ceva mai simplă. Avantajul real în utilizarea acestei funcţii este pentru construcţia unor protocoale de autentificare a informaţiei, ce vor fi prezentate în secţiunile următoare, iar studiul asupra potenţialei utilizări a acesteia în schema Lamport are rol de fundamentare pentru cele ce urmează.

4.2 Scurtă sinteză asupra metodelor de construcţie a lanţurilor one-way

4.2.1 Taxonomie asupra procedeelor constructive

În contextul creat, înainte de a porni descrierea protocoalelor de autentificare bazate pe lanţuri one-way, devine relevantă o scurtă sinteză asupra mecanismelor care pot sta la baza construcţiei lanţurilor one-way. În literatura de specialitate se întâlneşte cu preponderenţă noţiunea de lanţ hash. Aceasta se

datorează faptului că funcţiile hash sunt cele mai simple primitive criptografice din punct de vedere computaţional şi sunt cele mai frecvent utilizate în practică. Cu toate acestea noţiunea generală este aceea de lanţ one-way aceasta fiind introdusă iniţial de Lamport [79]. În esenţă orice primitivă criptografică este o funcţie one-way şi în consecinţă orice primitivă criptografică poate fi utilizată în construcţia unui lanţ one-way. Până acum am adus în discuţie doar utilizarea funcţiilor hash şi a funcţiei putere discretă, dar putem de exemplu considera funcţia de criptare simetrică $E_k(m)$ a mesajului m cu cheia k şi să definim funcţia $f(x) = E_x(0)$ unde x joacă rol de cheie secretă iar 0 este un mesaj de valoare nulă. Folosind această funcţie putem genera un lanţ one-way, prin compoziţia succesivă a acesteia obţinând de fapt şirul recurent $\sigma_\eta = f^\eta(x) = E_{\sigma_{\eta-1}}(0), \sigma_0 = x$. Este evident că şirul recurent definit astfel descrie un lanţ one-way şi pentru construcţia lui s-a utilizat o funcţie de criptare simetrică şi nu o funcţie hash. În ansamblu, putem concluziona asupra existenţei a două metode de bază în construcţia lanţurilor one-way a căror descriere urmează.

- **Construcţia lanţurilor one-way folosind primitive simetrice**

Utilizarea primitivelor criptografice simetrice reprezintă prima şi cea mai eficientă alternativă în construcţia lanţurilor one-way. Problema care apare este însă dimensiunea lanţului one-way, mai exact, dacă dimensiunea aleasă este prea mică lanţul este uşor de calculat dar se va epuiza rapid iar re-iniţializarea acestuia aduce alte probleme de securitate: este necesar un nou schimb autentificat de cheie. Două soluţii de re-iniţializare se găsesc tratate în [44], [127]. Pe de altă parte dacă lanţul este prea lung necesită prea multă putere de calcul deoarece complexitatea de calcul pentru generarea lanţului, sau a unui element de index dat, este funcţie liniară de dimensiunea lanţului respectiv de poziţia elementului. Rezultate noi aduc optimizări ale complexităţii de calcul pentru cazul generării unui element de index dat [24], [38], [71], [74], [107]. Toate aceste optimizări se bazează pe schimburi între spaţiul de stocare şi timpul de calcul şi au ca principiu stocarea unor valori din lanţ în scopul re-calculării eficiente a valorilor la un index dat.

Dintre primitivele simetrice, funcţiile hash sunt cele mai utilizate în practică, avantajul fiind costul computaţional redus la minim. Aşa cum a fost arătat anterior, se pot utiliza şi funcţii de criptare simetrică în acelaşi scop, sau într-o manieră similară pot fi utilizate coduri MAC. În practică însă aceste alternative nu sunt folosite deoarece nu aduc nici un avantaj în faţa funcţiilor hash şi necesită timp de calcul de câteva ori mai ridicat.

- **Construcţia lanţului folosind primitive asimetrice**

Utilizarea funcţiei putere modulo *n*, care este o primitivă frecvent utilizată în criptografia cu cheie publică, oferă avantajul unor dimensiuni infinite în practică pentru lanţul hash. Desigur noţiunea de lungime infinită poate părea bizară – calculatoarele sunt maşini discrete care pot face un număr finit de paşi, justificarea este însă aceea că dimensiunea lanţului nu mai afectează costul de calcul şi de aceea în principiu orice dimensiune poate fi aleasă pentru lanţ (de exemplu pot fi cu uşurinţă calculate elementele unui lanţ de dimensiune 2^{1024} iar un astfel de lanţ nu se va epuiza niciodată). Faptul că dimensiunea lanţului nu influenţează costul de calcul se datorează teoremei lui Euler care conduce la faptul că atunci când lucrăm

în Z_n exponenţii pot fi reduşi modulo $\phi(n)$. Astfel, timpul de calcul depinde doar logaritmic de dimensiunea lanţului, fiind uşor de observat că într-adevăr $f^{\eta}(x) = x^{\varepsilon^{\eta} \bmod \phi(n)} \bmod n$. De asemenea utilizarea acestei funcţii face posibil şi calculul unor valori la orice index fără a necesita calculul altor valori din lanţ.

Totuşi această funcţie este mult mai costisitoare din punct de vedere al timpului de calcul şi de asemenea dimensiunea cheilor este sporită semnificativ ceea ce face ca în practică utilizarea acestei funcţii să fie limitată ca arie de aplicabilitate. Cel mai eficient din punct de vedere computaţional este cazul exponentului $\varepsilon = 2$ în care se pot calcula elementele într-un time-memory trade şi în acest caz autenticitatea este asigurată la costul de o multiplicare modulară pentru fiecare sesiune. Utilizarea funcţiei putere discretă pentru generarea lanţurilor one-way în schema Lamport este discutată în [48] iar a cazului particular $\varepsilon = 2$ cu time-memory trade în [47], utilizarea acestor funcţii în protocolul DeMA este expusă în [53] iar utilizarea acestui caz particular eficient din punct de vedere computaţional se găseşte în [51].

Soluţii apropiate se găsesc în [11], [12]. În final este uşor de remarcat că în principiu orice semnătura digitală se poate utiliza pentru generarea unui lanţ one-way de dimensiune nelimitată, aceasta se poate realiza prin efectuarea unei semnături peste semnătura şi tot aşa, i.e. un lanţ de semnături.

Pentru generalitate, pe parcursul tezei se va folosi noţiunea de lanţ one-way urmând să reiasă uşor din context, acolo unde este cazul, dacă în particular soluţia descrisă se referă la lanţuri one-way generate de funcţii hash sau generate de primitive asimetrice. În figura 4.4 sunt sistematizate procedeele constructive ale lanţurilor one-way. În figură se remarcă cele două procedee constructive distincte bazate pe funcţii simetrice şi asimetrice, iar în cadrul acestora sunt subliniate cele două soluţii de interes: funcţiile hash utilizate pe scară largă în practică şi funcţia ridicare la pătrat discretă care oferă avantajul unor lanţuri de dimensiune infinită în practică.

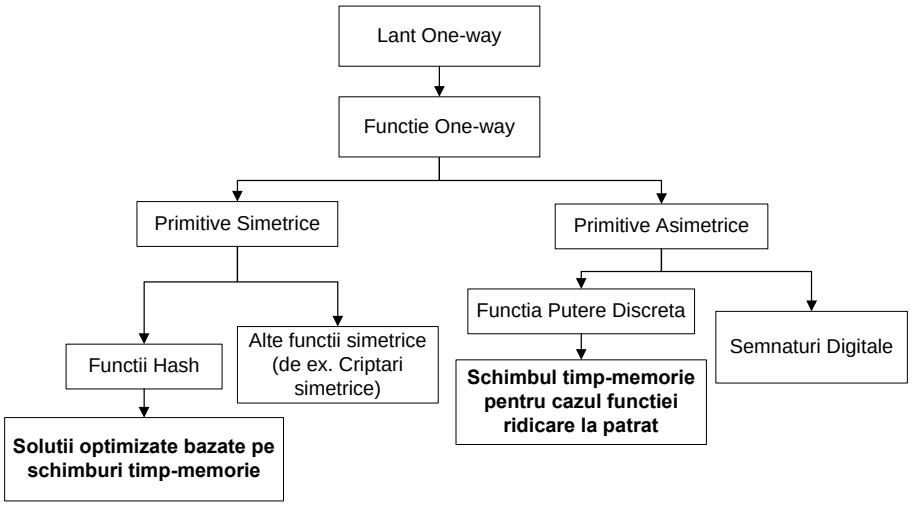


Figura 4.4. Taxonomie a procedeelor constructive pentru lanţuri one-way

4.2.2 Analiză asupra lungimii perioadei lanţurilor de reziduuri cvadratice

Funcţia putere discretă introdusă anterior, i.e. $f(x) = x^e \bmod n$, unde $n = p \cdot q$ este produsul a două numere prime p,q, este frecvent utilizată în criptografia cu cheie publică. Cele mai cunoscute propuneri sunt utilizarea unor cazuri particulare ale acestei funcţii în criptosistemele RSA şi Rabin, unde sunt folosite diverse valori pentru exponentul ε [99], [101]. Pentru a demonstra securitatea soluţiei propuse anterior, de generare a lanţurilor one-way folosind această funcţie, ne interesează în această secţiune stabilirea lungimii perioadei secvenţei generate de compoziţia acestei funcţii, i.e. lanţ one-way de reziduri cuadratice. Aceasta deoarece orice funcţie definită pe un set finit prin compoziţia ei succesivă conduce la un ciclu.

Primul studiu asupra lungimii perioadei secvenţei generate de această funcţie a fost făcut de către Blum et al. care au propus această funcţie în scopul construcţiei unui generator de numere pseudo-aleatoare [15], [16] cunoscut sub numele de generatorul Blum-Blum-Shub. Mult mai recent, aceeaşi funcţie a fost propusă pentru construcţia puzzleurilor criptografice de tip time-lock [103]. Acestea sunt construcţii criptografice ce pot fi utilizate pentru a transmite informaţie "în viitor" şi sunt din aceeaşi familie a puzzle-urilor criptografice utilizate pentru a preveni atacuri de tip DoS. Puzzle-urile de tip "time-lock" au avantajul enorm că pot fi rezolvate doar după o perioadă exactă de timp, fără să lase unui potenţial rezolvator posibilitatea de a paraleliza calculul, aceasta datorită caracterului intrinsec secvenţial al procesului de ridicare la pătrat repetată. Rivest et al. au considerat însă în lucrarea [103] că o analiză amănunţită asupra perioadei nu este necesară pentru scenariile de interes practic. Deci singura analiză asupra perioadei se regăseşte în [16]. Propunerea de puzzle-uri criptografice a lui Rivest et al. utilizează aceleaşi proprietăţi ale funcţiei care au fost anterior exploatate. Mai exact, se exploatează faptul anterior amintit că în grupuri de întregi exponenţii pot fi reduşi modulo ordinul grupului. Din acest motiv, rezultatul compoziţiei succesive al funcţiei poate fi eficient calculat ca $f^{\eta}(x) = x^{2^{\eta}} \bmod n = x^{2^{\eta} \bmod \phi(n)} \bmod n$. Aceeaşi proprietate a fost anterior utilizată pentru a crea un lanţ de dimensiune nelimitată în practică deoarece valoarea lui $f^{\eta}(x) = x^{2^{\eta} \bmod \phi(n)} \bmod n$ poate fi eficient calculată, aşa cum s-a precizat anterior, calculând întâi exponentul $e = 2^{\eta} \bmod \phi(n)$ şi apoi calculând $f^{\eta}(x) = x^e \bmod n$. Astfel, în timp ce complexitatea de calcul în cazul utilizării funcţiilor hash depinde liniar de dimensiunea lanţului, pentru această funcţie timpul de calcul va depinde doar logaritmic de dimensiunea modulului, datorită algoritmului de calcul prin ridicări la pătrat repetate. Mai mult, elementele lanţului one-way pot fi calculate eficient aşa cum a fost explicat anterior la costul unei simple multiplicări modulare.

Vom încerca aşadar să determinăm perioada şirului recurent generat de compoziţia succesivă a funcţiei ridicare la pătrat discretă:

$$k_i = x_0^{2^{\eta - i} \bmod \phi(n)} \bmod n, i = 0,..\eta \qquad (4.11)$$

Aşa cum au intuit Rivest et al., a căuta perfecţiune în teoria numerelor poate fi exagerat în acest context şi numărul x_0 poate fi o valoare aleatoare pentru a

obţine un lanţ de dimensiune nemărginită în practică. Desigur este natural să ne aşteptăm că prin alegerea de valori aleatoare pentru x_0 şi pentru n vom obţine o secvenţă cu o perioadă suficient de mare fără a duce la pierderea securităţii deoarece probabilitatea de a alege un element de ordin mic este neglijabilă. Din aceste motive alegerea de valori aleatoare trebuie să fie sigură în practică, dar, pentru a da un rezultat cât mai complet, vom oferi şi o soluţie de alegere a acestor numere care să nu fie contrară aşteptărilor. În acest sens vom da o soluţie directă care este apropiată de cea utilizată de Blum et al. în dezvoltarea generatorului de numere pseudo-aleatoare.

Pentru a alege aceste numere suntem preocupaţi de două lucruri care influenţează dimensiunea lanţului: primul este ordinul lui x_0 în Z_n, notat cu $ord_n(x_0)$, şi al doilea, ordinul lui 2 în $Z_{\phi(n)}$, notat cu $ord_{\phi(n)}(2)$. Vom nota cu π lungimea perioadei, aceasta reprezentând cel mai mic număr astfel încât $x_0 = x_0^{2^\pi} \bmod n$, i.e.:

$$x_0 = x_0^{2^\pi} \bmod n,\ \neg\exists \pi' < \pi,\ x_0 = x_0^{2^{\pi'}} \bmod n \qquad (4.12)$$

Primul pas pe care îl vom face este obţinerea unei valori suficient de mari pentru $ord_{\phi(n)}(2)$. Ordinul lui 2 în $Z_{\phi(n)}$ poate fi uşor verificat dacă se cunoaşte factorizarea lui $\phi(n)$. În acest scop vom utiliza acelaşi tip de numere prime ca în [16], numite numere prime speciale, pentru care este adevărat că $p = 2 \cdot p' + 1$ şi $p' = 2 \cdot p'' + 1$, aici p, p', p'' sunt toate numere prime. De asemenea, aplicăm aceleaşi cerinţe pentru modul ca în [16] şi cerem ca n să fie un număr special, adică atât p cât şi q să fie numere speciale şi totodată congruent cu 3 mod 4 (aceasta implică faptul că fiecare reziduu cvadratic are o singură rădăcină care este tot un reziduu cvadratic). Dacă p şi q sunt numere prime speciale atunci valoarea lui $\phi(\phi(n))$ este evident $\phi(\phi(n)) = \phi((p-1)\cdot(q-1)) = \phi(4 \cdot p' \cdot q') = 8 \cdot p'' \cdot q''$. Din moment ce $\phi(\phi(n))$ este cunoscut şi factorizarea sa de asemenea, valoarea lui $ord_{\phi(n)}(2)$ poate fi uşor calculată. Vom alege numerele prime p şi q în aceeaşi manieră ca în [16] prin alegerea de numere aleatoare şi selectarea acelor numere care sunt speciale, alegerea se face generând numere aleatoare şi testându-le pentru primalitate prin teste probabilistice. Alegerea se va face astfel încât să garanteze că $ord_{\phi(n)}(2)$ este $2 \cdot p'' \cdot q''$.

Al doilea pas de care trebuie să ne ocupăm este alegerea unui x_0 de ordin cât mai mare în Z_n. În acest scop putem recurge la o soluţie simplă prin alegerea a două numere aleatoare α şi β care sunt generatori în Z_p şi respectiv Z_q. Acum, prin utilizarea Teoremei Chineze a Resturilor putem calcula soluţia următorului sistem:

$$\begin{cases} x_{-1} = \alpha \bmod p \\ x_{-1} = \beta \bmod q \end{cases} \qquad (4.13)$$

Am folosit notaţia x_{-1} pentru a desemna valoare din care vom genera primul reziduu cvadratic din lanţ, i.e. x_0. Ordinul lui x_{-1} în Z_n va fi $ord_n x_{-1} = cmmmc(p-1, q-1)$. Dacă setăm $x_0 = x_{-1}^2 \bmod n$ atunci vom obţine lungimea perioadei π, ca fiind egală cu:

$$\pi = ord_{ord_n(x_0)}(2) = ord_{\phi(n)} 2 \qquad (4.14)$$

Această valoare duce la o lungime a ciclului care este în mod evident suficient de sigură pentru utilizarea în practică.

4.3 Protocoale bazate pe coduri de autentificare şi lanţuri de chei

Obiectivul acestei secţiuni este de a prezenta contribuţiile în zona protocoalelor de autentificare bazate pe lanţuri one-way. Pentru început se creează un preambul care are rolul de a sublinia relevanţa practică a acestor protocoale şi de a creea o imagine de ansamblu asupra soluţiilor existente pentru încadrarea contribuţiilor în peisajul contemporan.

Autentificarea informaţiei este unul dintre cele mai importante obiective de securitate. Cu toate că aparent autentificarea are costuri computaţionale mai scăzute, deoarece codurile de autentificare sunt mai puţin intense din punct de vedere computaţional decât alte primitive criptografice, cum ar fi funcţiile de criptare utilizate pentru asigurarea confidenţialităţii informaţiei sau ca semnăturile digitale utilizate pentru asigurarea non-repudierii, scenariile din lumea reală nu pot fi uşor rezolvate prin aplicarea directă a acestor primitive. Un bun exemplu în acest caz îl constituie un scenariu de transfer în regim broadcast, acest exemplu are relevanţă deoarece în scenarii practice apare frecvent necesitatea de a transmite aceeaşi informaţie către mai mulţi receptori. Problema care apare este faptul că MAC-urile, funcţiile criptografice utilizate în autentificare, necesită chei secrete partajate, şi astfel, un potenţial emiţător trebuie să partajeze o cheie secretă cu fiecare receptor – lucru dezavantajos datorită necesităţii de a dispeceriza un număr ridicat de chei. Mai mult, emiţătorul va trebui să şi calculeze un MAC distinct pentru fiecare receptor chiar dacă informaţia transmisă este aceeaşi. Din fericire, există o soluţie excelentă pentru a evita acest dezavantaj – utilizarea protocoalelor de autentificare bazate pe lanţuri one-way şi sincronizare temporală propuse de Perrig et al. Protocoalele bazate pe acest principiu se dovedesc a fi extrem de versatile oferind proprietăţi de securitate apropiate cu ale protocoalelor care utilizează operaţii costisitoare bazate pe primitive cu cheie publică.

Istoria utilizării lanţurilor one-way a debutat cu propunerea lui Lamport [79] care a introdus utilizarea elementelor dintr-un lanţ one-way în scopul folosirii ca parole one-time pentru autentificarea unui utilizator faţă de un sistem. Mai târziu această propunere a fost utilizată în sistemul S-Key propus de Haller despre care s-a precizat anterior că nu este sigur. Dezavantajul schemei Lamport pentru aplicaţii din lumea reală este evident: datorită autentificării unilaterale, ne-existând autentificare din partea serverului către client, un potenţial adversar poate impersona serverul pentru a sustrage din partea utilizatorului parole încă nefolosite şi a le utiliza ulterior

pentru impersonarea acestuia (acest atac este cunoscut sub numele de atac de tip pre-play).

Deşi au existat diverse propuneri de utilizare a lanţurilor one-way [90], [102], [108], succesul acestora se datorează propunerilor lui Perrig et al. [92], [93], [94], [95] care vor folosi lanţuri one-way pentru asigurarea autenticităţii informaţiei. Acestea au făcut ca protocoale care folosesc lanţuri one-way să-şi găsească aplicabilitate reală în diverse zone de la semnături digitale şi protocoale de routare a traficului până la reţele de senzori [22], [68], [70], [92], [94], [97], [100], [126]. Codurile de autentificarea a mesajelor MAC sunt primitivele criptografice utilizate în acest scop, dar, aşa cum s-a precizat, ele vin cu un dezavantaj: utilizarea cheilor secrete partajate între emiţător şi orice receptor. Folosirea cheilor provenind din elemente ale unor lanţuri one-way este o soluţie eficientă în eliminarea acestui dezavantaj. Pe scurt explicaţia este următoarea: codul MAC rămâne sigur atâta timp cât cheia utilizată pentru calcularea lui este făcută publică doar după ce toţi receptorii au primit şi stocat codul MAC, de asemenea din moment ce fiecare element al unui lanţ one-way este o garanţie pentru elementul imediat următor, următorul element din lanţ poate fi utilizat ca şi cheie pentru următorul MAC şi aşa mai departe.

Cea mai eficientă propunere de protocol în acest sens este protocolul TESLA Timed Efficient Stream Loss-tolerant Authentication propus de Perrig et al. Diverse variante ale acestui protocol sunt propuse, toate fiind bazate pe o sincronizare temporală slabă între emiţător şi receptori. Principiul este de a utiliza un element al unui lanţ one-way ca şi cheie pentru un MAC şi de a face publică această cheie doar într-un pachet următor, condiţia de securitate care trebuie îndeplinită pentru ca autentificarea să fie sigură este faptul că fiecare receptor poate decide bazându-se pe sincronizarea temporală dacă emiţătorul a trimis sau nu pachetul care conţinea cheia MAC-ului. Pe scurt protocolul TESLA oferă autenticitate la costuri scăzute fără să implice secrete partajate între emiţători şi receptori. Pentru acest avantaj protocolul a fost fezabil chiar şi în medii constrânse cum ar fi reţelele de senzori [92].

O alta gamă de protocoale de autentificare, în care elemente din lanţuri one-way sunt utilizate ca şi confirmare autentică, evitându-se astfel utilizarea sincronizării temporale, se găsesc în [10], [50], [51], [53], [56], [57], [60]. Toate aceste propuneri au ca precursor protocolul Guy Fawkes care se bazează pe principii similare [1].

În figura 4.5 este prezentată o clasificare a protocoalelor bazate pe lanţuri one-way. Sunt relevante cele două categorii bazate pe sincronizare temporală respectiv challenge-response, schema Lamport nu este de mare însemnătate practică, valoarea ei fiind de natură teoretică prin principiile introduse.

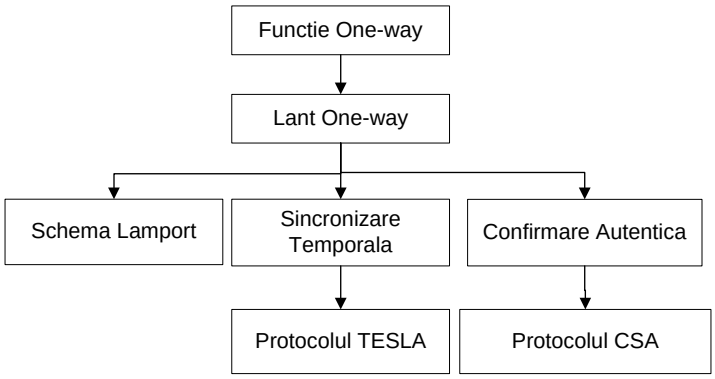


Figura 4.5. Taxonomie a protocoalelor de autentificare bazate pe lanţuri one-way

4.3.1 Protocolul Delayed Message Authentication - Direct Chain Authentication (DeMA-DiCA)

În rândul propunerilor de protocoale de autentificare bazate pe lanţuri one-way, fără sincronizare temporală, protocolul Delayed Message Authentication-Direct Chain Authentication (DeMA-DiCA) este singurul protocol bazat în exclusivitate pe lanţuri one-way. Celelalte propuneri necesită fie sincronizare temporală fie semnături digitale pentru iniţializarea lanţurilor. Protocolul are la bază următoarele două componente distincte ce vor fi detaliate în secţiunile următoare:

1) Protocolul Delayed Message Authentication (DeMA). Este un protocol de autentificare similar cu protocolul CSA descris în capitolul 3. Diferenţa faţă de CSA este faptul că DeMA a fost propus în cazul general al utilizării unor funcţii one-way şi nu în cazul restrâns al funcţiilor hash, precum CSA. Lucrările [51], [53] tratează câteva cazuri de utilizare a funcţiei putere discretă.

2) Protocolul Direct Message Authentication (DiMA). Este un protocol de autentificare ce utilizează două lanţuri one-way pentru producerea unei semnături one-time înlănţuite ce serveşte reiniţializării lanţurilor din DeMA.

Pe scurt, protocolul DeMA-DiCA foloseşte protocolul DeMA pentru schimbul de mesaje şi protocolul DiMA pentru re-iniţializarea lanţurilor one-way.

- **Protocolul DeMA**

Protocolul DeMA este un protocol dedicat schimbului autentic de informaţie între două entităţi care utilizează pentru aceasta câte un lanţ one-way de fiecare parte. Vom nota în continuare cu $\sigma_A(k)$ respectiv $\sigma_B(k)$ elementele lanţurilor one-way utilizate de partea fiecărei entităţi iar relaţiile de calcul corespunzătoare acestora sunt următoarele:

$$\sigma_A(k) = f^{\eta_A - k}(x_A),\ k = \overline{0,\eta} \tag{4.15}$$

$$\sigma_B(k) = f^{\eta_B - k}(x_B),\ k = \overline{0,\eta} \tag{4.16}$$

Aceste elemente vor juca rolul unor chei de sesiune. În relaţiile (4.15) şi (4.16) x_A şi x_B sunt două valori generate aleator şi păstrate secrete de fiecare parte, f este o funcţie one-way respectiv η_A, η_B sunt lungimile lanţurilor alese de fiecare entitate (pentru simetrie pot fi egale), k este numărul sesiunii de comunicare. Într-un pas off-line de iniţializare entităţile îşi fac cunoscute valorile lui $\sigma_A(0)$ respectiv $\sigma_B(0)$ (care reprezintă vârfurile lanţurilor one-way). Iniţializarea se va face în aşa măsură încât să garanteze autenticitatea acestei informaţii şi faptul că aceste lanţuri sunt dorite pentru schimbul de informaţie între cele două entităţi. Ulterior aceste două valori vor fi utilizate pentru testarea autenticităţii informaţiei trimise în sesiunea 1.

Protocolul are un număr variabil de sesiuni de comunicare şi fiecare sesiune constă în două runde, fiecare rundă joacă rol de confirmare pentru runda anterioară şi fiecare sesiune este folosită pentru a transmite un mesaj şi o cheie nouă care la rândul ei face posibilă verificarea autenticităţii mesajului din sesiunea anterioară. Aceasta înseamnă că autenticitatea mesajului din sesiunea k poate fi verificată în sesiunea $k+1$. Pentru sesiunea de comunicare k entitatea A trimite spre B pachetul cu structura:

$$P_{A,k} = \left\{M_{A,k}, MAC_{\sigma_A(k+1)}(M_{A,k}), \sigma_A(k)\right\} \tag{4.17}$$

În acest pachet $M_{A,k}$ reprezintă mesajul, MAC este un cod de autentificare a mesajelor calculat cu cheia $\sigma_A(k+1)$ care va fi făcută publică în sesiunea $k+1$ iar $\sigma_A(k)$ este cheia curentă de sesiune. La primirea mesajului B are de întreprins următoarele operaţii (operaţii similare sunt necesare şi pentru DeMA-QR din secţiunea următoare):

1) Verifică dacă $\sigma_A(k)$ este cheia corectă de sesiune testând că $f(\sigma_A(k)) = \sigma_A(k-1)$ (aici $\sigma_A(k-1)$ este cheia sesiunii anterioare). Dacă cheia este corectă atunci se continuă cu paşii 2 şi 3 altfel se aşteaptă o nouă cheie.

2) $P_{A,k}$ este stocat iar autenticitatea mesajului $M_{A,k-1}$ din sesiunea anterioară $k-1$ poate fi acum verificată folosind noua cheie de sesiune $\sigma_A(k)$ pentru a testa codul de autentificare $MAC(M_{A,k-1}, \sigma_A(k))$.

3) B va confirma primirea unei chei corecte de sesiune prin transmiterea cheii proprii de sesiune $\sigma_B(k)$.

Entitatea A va porni sesiunea următoare de comunicare $k+1$ doar dacă valoarea lui $\sigma_B(k)$ este corectă iar acest lucru este uşor de dovedit verificând că $f(\sigma_B(k)) = \sigma_B(k-1)$. Astfel, sesiunea de comunicare k decurge în următoarele două runde:

Sesiunea k

Runda 1 $A \to B:\ P_{A,k} = \left\{M_{A,k}, MAC_{\sigma_A(k+1)}(M_{A,k}), \sigma_A(k)\right\}$

Runda 2 $B \to A:\ \sigma_B(k)$

În cazul în care este convenabil ca şi B să includă un mesaj propriu în răspuns rundele alternative ale comunicării sunt următoarele:

Sesiune alternativă k

Runda 1 $A \to B:\ P_{A,k} = \left\{M_{A,k}, MAC_{\sigma_A(k+1)}(M_{A,k}), \sigma_A(k)\right\}$

Runda 2 $B \to A:\ P_{B,k} = \left\{M_{B,k}, MAC_{\sigma_B(k+1)}(M_{B,k}), \sigma_B(k)\right\}$

Este important de subliniat că fiecare rundă joacă un rol de confirmare pentru runda anterioară, astfel runda 1 a sesiunii k este confirmarea rundei 2 a sesiunii $k-1$ în timp ce runda 2 a sesiunii k este confirmarea rundei 1 a sesiunii k **-** o astfel de confirmare este trimisă doar dacă cheia de sesiune a rundei confirmate este corectă.

Protocolul poate fi oprit în orice sesiune de către A urmând ca autenticitatea mesajului din acea sesiune să fie dovedită doar când protocolul continuă şi cheia sesiunii următoare este făcută publică. La repornirea protocolului, A va trimite o nouă cheie de sesiune doar dacă pachetul din sesiunea anterioară a fost confirmat, altfel pachetul din runda anterioară va fi retrimis. Pentru exemplificare vom presupune că protocolul a fost oprit în sesiunea k descrisă anterior**.** Dacă A a primit valoarea corectă a lui $\sigma_B(k)$ atunci protocolul poate fi ulterior pornit prin transmiterea pachetului aferent rundei $k+1$ altfel protocolul va fi repornit prin transmiterea pachetului din runda 1 a sesiunii k până când un răspuns valid $\sigma_B(k)$ este primit. Aceste reguli trebuie urmate în mod strict deoarece dacă A trimite pachetul pentru sesiunea $k+1$ fără să fi primit cheia de confirmare $\sigma_B(k)$ din sesiunea k**,** în cazul în care pachetul din sesiunea k nu a fost primit de B acest pachet poate fi acum fraudat de un adversar care are în momentul de faţă acces la $\sigma_A(k)$. Aceeaşi regulă trebuie strict urmată de B care trebuie să trimită confirmarea $\sigma_B(k)$ în runda 2 a sesiunii k dacă şi numai dacă valoarea primită $\sigma_A(k)$ este corectă. Aparent pare a fi convenabil ca oricare dintre entităţi să poată opri şi reporni protocolul prin retrimiterea ultimului pachet, acest lucru nu este însă posibil. În cazul în care oricare dintre entităţi ar putea porni şi opri protocolul,

trimiţând un pachet nou dacă pachetul anterior a fost confirmat respectiv pachetul anterior dacă nu s-a primit o astfel de confirmare, atunci aceasta ar permite unui potenţial adversar să retrimită pachete între cele două entităţi astfel încât protocolul să nu se mai oprească.

- **Limitări şi extensii ale protocolului DeMA**

Limitarea protocolului DeMA apare în momentul în care lanţurile one-way sunt epuizate şi entităţile rămân fără chei de sesiune. În acest caz lanţurile trebuie reiniţializate, practic aceasta înseamnă calcularea unor noi lanţuri şi transmiterea vârfurilor acestora într-o manieră care să garanteze autenticitatea acestei informaţii. Problema care apare este faptul că protocolul DeMA nu se poate utiliza pentru acest scop, motivul este simplu: chiar dacă un adversar nu poate altera autenticitatea informaţiei transmise cu protocolul DeMA acesta poate altera informaţia astfel încât eventualul schimb al noilor elemente din noile lanţuri să cadă testul de autenticitate – moment în care entităţile au rămas deja fără chei şi comunicarea nu mai poate continua. Există trei soluţii pentru această problemă, cea de a 3-a fiind doar o soluţie parţială ce rămâne în continuare vulnerabilă:

1) Utilizarea unei semnături digitale pentru reiniţializarea lanţurilor. O astfel de soluţie este în esenţă abordată în protocolul CSA. De asemenea protocolul DiCA descris în paragraful următor este o soluţie ce se încadrează în această categorie şi are proprietatea de a se baza tot pe lanţuri one-way.

2) Utilizarea lanţurilor de dimensiune nemărginită generate de primitive cu cheie publică. Este valid în acest sens cazul funcţiei putere discretă anterior amintit. În această situaţie lanţul nu este niciodată epuizat şi limitarea dispare. Dezavantajul este că o astfel de soluţie, datorită complexităţii computaţionale a funcţiei putere discretă, nu se pretează pentru toate cazurile întâlnite în practică. Lucrările [51], [53] tratează o astfel de soluţie.

3) Schimbarea structurii pachetului din $\left\{M_{A,k}, MAC_{\sigma_A(k+1)}\left(M_{A,k}\right), \sigma_A(k)\right\}$ în $\left\{E_{\sigma_A(k+1)}\left(M_{A,k}\right), MAC_{\sigma_A(k+1)}\left(M_{A,k}\right), \sigma_A(k)\right\}$; şi în mod similar pentru pachetul din runda 2 din $\left\{M_{B,k}, MAC_{\rho_B(k+1)}\left(M_{B,k}\right), \sigma_B(k)\right\}$ în $\left\{E_{\sigma_B(k+1)}\left(M_{B,k}\right), MAC_{\sigma_B(k+1)}\left(M_{B,k}\right), \sigma_B(k)\right\}$. Unde E reprezintă o primitivă de criptare simetrică şi aceasta înseamnă că mesajul este criptat cu cheia $\sigma_A(k+1)$ care urmează să fie făcută publică doar în sesiunea următoare. Această soluţie poate deveni utilă pentru că un adversar nu poate decide în acest caz dacă informaţia transmisă este sau nu informaţia necesară reiniţializării lanţurilor one-way. Totuşi o astfel de soluţie rămâne ineficientă în faţa unui adversar persistent care continuă să altereze mesaje până când lanţurile celor două entităţi sunt epuizate. De asemenea structura pachetului poate fi simplificată prin renunţarea la MAC în cazul în care criptarea $E_{\sigma_A(k+1)}\left(M_{A,k}\right)$ prezintă elemente de redundanţă care să

previnã adversarul în a introduce în pachet o valoare arbitrară ce ar putea fi decriptată şi interpretată ca fiind corectă.

- **Protocolul DiMA**

Explicaţia pentru neajunsul protocolului DeMA în cazul reiniţializării lanţurilor one-way este faptul că autentificarea este obţinută la o întârziere de o sesiune, lucru care permite primirea unor mesaje care se vor dovedi ca fiind neautentice. Pentru a înlătura acest dezavantaj este necesară prezenţa unui mecanism care să permită autentificarea directă a mesajului. Deoarece dorim în continuare să evităm folosirea secretelor partajate, soluţia la care vom face apel va utiliza două lanţuri one-way de partea fiecărei entităţi; aceasta va conduce de fapt la o schemă de semnătură digitală one-time înlănţuită. La fel ca şi protocolul DeMA, protocolul constă într-un număr variabil de sesiuni şi fiecare sesiune constă în două runde, fiecare rundă fiind confirmarea rundei anterioare. În fiecare rundă două elemente ale lanţului one-way sunt făcute publice şi din aceste două elemente este recuperat mesajul. Continuăm cu descrierea detaliată a protocolului.

Un întreg pozitiv λ este fixat de comun acord între cele 2 entităţi în aşa fel încât să fie uşor pentru ambele entităţi calculul lui $f^{\lambda}(x)$ prin λ compoziţii succesive ale funcţiei one-way (valori diferite pentru λ pot fi utilizate de cele 2 părţi dar aceasta doar ar complica descrierea protocolului). Dacă considerăm mesajul ca fiind un întreg în intervalul $[1,\lambda]$ atunci mesajul va avea $\lfloor \log_2(\lambda-1) \rfloor + 1$ biţi.

Entităţile *A* şi *B* aleg câte doi întregi aleatori x_A, y_A respectiv x_B, y_B şi păstrează secret aceste valori. Cheile de sesiune sunt reprezentate de o pereche de elemente din cele două lanţuri $\theta_A(k), \omega_A(k)$ şi $\theta_B(k), \omega_B(k)$ calculate pe baza valorilor aleatoare alese anterior în conformitate cu relaţiile:

$$\theta_A(k) = f^{\eta - m_{A,k} - (k-1)\cdot\lambda - 1}(x_A) \tag{4.18}$$

$$\omega_A(k) = f^{\eta + m_{A,k} - k\cdot\lambda}(y_A) \tag{4.19}$$

$$\theta_B(k) = f^{\eta - m_{B,k} - (k-1)\cdot\lambda - 1}(x_B) \tag{4.20}$$

$$\omega_B(k) = f^{\eta + m_{B,k} - k\cdot\lambda}(y_B) \tag{4.21}$$

În aceste relaţii k reprezintă numărul sesiunii, η_A şi η_B sunt dimensiunile lanţurilor one-way alese de cele două entităţi iar $m_{A,k}$, $m_{B,k}$ sunt mesajele autentice trimise în fiecare sesiune. Se poate observa din relaţiile anterioare că lanţurile sunt de fapt împărţite în secvenţe de lungime λ şi fiecare astfel de secvenţă corespunde unei anume sesiuni. Astfel, dacă lungimea lanţurilor este η_A respectiv η_B şi

lungimea secvenţelor λ, numărul maxim de sesiuni de comunicare până la epuizarea lanţurilor este $\frac{\eta_A}{\lambda}$ şi respectiv $\frac{\eta_B}{\lambda}$ (în acest sens este desigur practic ca valorile η_A, η_B să fie alese ca multiplii de λ din moment ce numărul de sesiuni este un număr întreg). Structura lanţurilor one-way pentru protocolul DiMA este sugerată în figurile 4.6, 4.7.

În sesiunea 0 entităţile schimbă valorile $\theta_A(0), \omega_A(0)$ şi $\theta_B(0), \omega_B(0)$ în aşa fel încât sursa şi autenticitatea acestei informaţii să fie garantată, această fază de iniţializare va fi întreprinsă off-line conform descrierii protocolului din [50]. În prima rundă a fiecărei sesiuni de comunicare, A va reprezenta mesajul ca o valoare întreagă $m_{A,i} \in [1, \lambda]$, şi va calcula perechea $\theta_A(k), \omega_A(k)$ descrisă prin relaţiile anterioare după care o va trimite lui B. După primirea perechii $\theta_A(k), \omega_A(k)$, B va determina prin compoziţii succesive ale funcţiei f doi întregi α, β având proprietăţile următoare: $0 < \alpha < \lambda$, $0 < \beta < \lambda$, $f^{\alpha}(\theta_A(k)) = \theta_A(k-1)$, $f^{\beta}(\omega_A(k)) = \omega_A(k-1)$. Dacă există aceşti întregi atunci mesajul autentic este $m_{A,k} = \alpha - m_{A,k-1}$, $m_{A,0} = 0$ şi B va confirma primirea acestui mesaj prin trimiterea unei perechi similare $\theta_B(k), \omega_B(k)$ calculate pentru mesajul $m_{B,k}$ (în particular dacă nu există nici o informaţie de transmis către A se va utiliza un mesaj arbitrar din intervalul $[1, \lambda]$). Următoarea sesiune de comunicare $k+1$ este pornită de A doar dacă pachetul primit de la B a fost autentic, A poate verifica autenticitatea acestui pachet într-o manieră similară cu B.

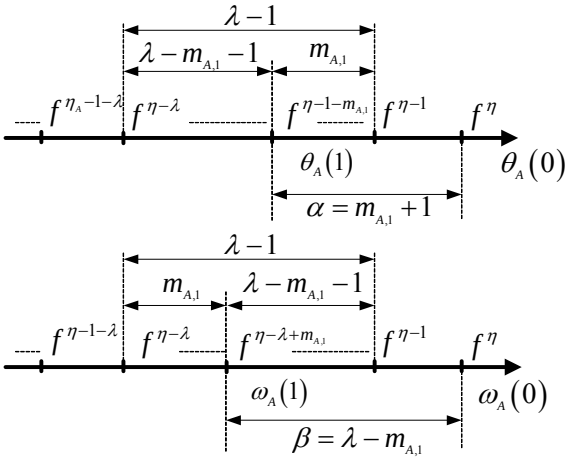


Figura 4.6. Structura lanţurilor one-way în sesiunea 1 a protocolului DiMA

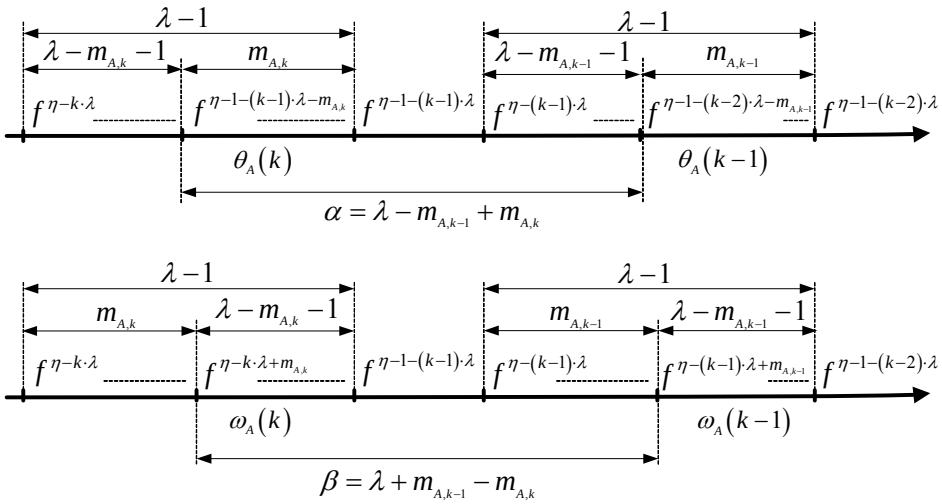


Figura 4.7. Structura lanţurilor one-way în sesiunea *k* a protocolului DiMA

Pentru sesiunea k, comunicarea decurge în modul următor:

Sesiunea *k*

Runda 1**:** $A \to B:\ \theta_A(k),\ \omega_A(k)$

Runda 2**:** $B \to A:\ \theta_B(k),\ \omega_B(k)$

Din nou este important de înţeles că fiecare rundă joacă rolul unei confirmări a rundei anterioare şi o astfel de confirmare este trimisă doar dacă mesajul din runda anterioară a fost autentic – subliniem că autenticitatea mesajului poate fi de această dată verificată în cadrul aceleiaşi runde.

La fel ca şi în cazul protocolului DeMA, protocolul DiMA poate fi pornit şi oprit doar de A în orice moment. Când A decide să repornească comunicaţia va trimite un nou pachet doar dacă ultimul pachet trimis a fost confirmat, în caz contrar el va trimite ultimul pachet neconfirmat până la primirea unei confirmări valide.

- **Rolul parametrului λ în protocolul DiMA**

Parametrul λ poate fi folosit pentru a îmbunătăţi flexibilitatea comunicaţiei. Să presupunem că o entitate doreşte trimiterea unui mesaj M. Prin utilizarea acestui protocol doar mesaje situate în intervalul $[1, \lambda]$ pot fi trimise în fiecare rundă. Aceasta înseamnă că lungimea fiecăruia din cele 2 lanţuri necesare pentru a transmite mesajul M este dată de următoarea relaţie:

$$l_{owc} = \lambda \cdot \left(\left\lfloor \log_\lambda (M-1) \right\rfloor + 1 \right) \qquad (4.22)$$

În timp ce numărul de sesiuni de comunicare este:

$$n_{ses} = \left\lfloor \log_{\lambda}(M-1) \right\rfloor + 1 \tag{4.23}$$

Efortul de calcul asociat fiecărei runde, mai exact numărul de compoziţii succesive ale funcţiei one-way necesar pentru a recupera mesajul din fiecare rundă, va fi:

$$c_{ef} = 2 \cdot \lambda \tag{4.24}$$

Pentru a recupera întreg mesajul M efortul computaţional va fi $n_{ses} \cdot c_{ef}$. Acum se poate observa imediat că prin creşterea valorii lui λ numărul de sesiuni descreşte în timp ce lungimea lanţului şi efortul de a recupera mesajul cresc. Deci un efort computaţional scăzut se obţine pentru valorile $\lambda = 2$ sau $\lambda = 3$, minimul funcţiei $f(x) = x \cdot \log_x a$ fiind în $x = e$ unde $e \approx 2.7$, şi tot pentru aceste valori se obţine şi cel mai mare număr de sesiuni de comunicare.

De exempu pentru $\lambda = 2$ avem lungimea lanţului one-way ca fiind:

$$l_{\min} = 2 \cdot \left(\left\lfloor \log_2(M-1) \right\rfloor + 1 \right) \tag{4.25}$$

Un lanţ puţin mai scurt s-ar obţine pentru $\lambda = 3$ din considerentul de minim anterior amintit. Totodată pentru $\lambda = 2$ se obţine şi cel mai ridicat număr de sesiuni de comunicare:

$$n_{high} = \left\lfloor \log_2(M-1) \right\rfloor + 1 \tag{4.26}$$

dar şi cel mai mic efort computaţional în fiecare rundă:

$$c_{\min} = 4 \tag{4.27}$$

Din moment ce utilizarea unor lanţuri de dimensiuni ridicate duce la creşterea timpului de calcul, valori mici pentru λ sunt preferabile lanţul fiind astfel epuizat mai greu. În medii unde abilităţile de comunicare sunt drastic limitate, valori mari pentru λ pot fi luate în considerare acestea conducând la scăderea numărului de sesiuni, conform relaţiei (4.23), dar şi la creşterea puterii de calcul utilizate.

Pentru exemplificare vom presupune că o entitate trebuie să trimită un mesaj M de 320 biţi. Se observă că dimensiunea mesajului este exact de două ori dimensiunea a două ieşiri ale funcţiei hash SHA1. Prin utilizarea lui $\lambda = 2$ două lanţuri de 642 elemente sunt necesare în 321 sesiuni de comunicare cu un efort de 4 compoziţii ale funcţiei one-way pentru recuperarea mesajului trimis în fiecare sesiune. Prin utilizarea lui $\lambda = 2^{10}$ două lanţuri de 33792 elemente sunt necesare în

33 sesiuni de comunicare cu 2048 compoziţii ale funcţiei one-way pentru recuperarea mesajului trimis în fiecare sesiune.

- **Deficienţa protocolului DiMA**

Protocolul DiMA are o singură deficienţă: lanţul one-way este extrem de rapid epuizat în schimbul de mesaje. Altfel protocolul DiMA este rezistent în faţa unui adversar persistent care putea duce la epuizarea lanţurilor în cazul protocolului DeMA, folosind DiMA lanţurile one-way pot fi iniţializate fără probleme de securitate.

- **Protocolul DeMA-DiCA**

Concluzionăm că protocolul DeMA este mult mai eficient în schimbul de mesaje decât DiMA dar nu poate fi utilizat pentru reiniţializarea lanţurilor one-way. În timp ce DiMA este ineficient la schimbul de informaţie dar nu este vulnerabil la reiniţializarea lanţurilor one-way. Acum, propunerea protocolului DeMA-DiCA devine o consecinţă imediată: desigur că utilizarea unui protocol hibrid care să folosească DeMA pentru schimbul de mesaje şi DiMA pentru reiniţializarea lanţurilor este un bun compromis. De aici şi acronimul DeMA-DiCA Delayed Message Authentication - Direct Chain Authentication. Protocolul funcţionează în baza următoarelor principii:

1) Se utilizează o funcţie one-way pe δ biţi.
2) Două lanţuri one-way sunt generate de fiecare entitate, unul de dimensiune $\eta + 4 \cdot \delta$ altul de dimensiune $4 \cdot \delta$. Fie σ_A lanţul de dimensiune $\eta + 4 \cdot \delta$ de pe partea lui A şi ρ_B lanţul de lungime $\eta + 4 \cdot \delta$ de pe partea lui B. De asemenea, fie $\omega_{\sigma,A}$ lanţul de lungime $4 \cdot \delta$ de pe partea lui A şi $\omega_{\rho,B}$ cel de lungime $4 \cdot \delta$ de pe partea lui B.
3) Primele η elemente din lanţurile σ_A şi ρ_B sunt utilizate pentru a schimba informaţie autentică între A şi B prin utilizarea protocolului DeMA.
4) Ultimele $4 \cdot \delta$ elemente din lanţurile σ_A şi ρ_B alături de cele $4 \cdot \delta$ elemente din lanţurile $\omega_{\sigma,A}$ şi $\omega_{\rho,B}$ vor fi utilizare de către A şi B pentru reiniţializarea lanţurilor folosind protocolul DiMA.

Structura generală a lanţurilor one-way pentru protocolul DeMA-DiCA este sugerată în figura 4.8, distribuţia lor de partea fiecărei entităţi este sugerată în figura 4.9, iar utilizarea şi reiniţializarea lor în figura 4.10.

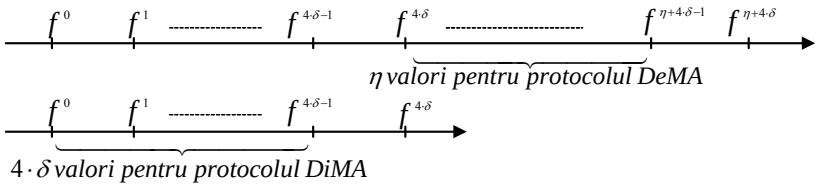


Figura 4.8. Structura lanţurilor one-way în protocolul DeMA-DiCA

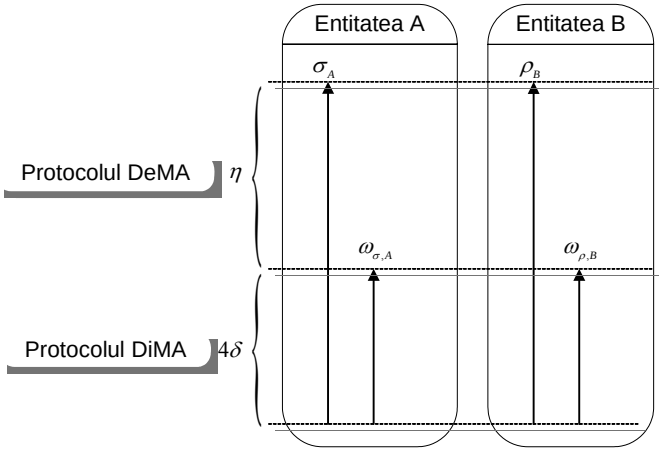


Figura 4.9. Lanţurile one-way de partea celor 2 entităţi în protocolul DeMA-DiCA

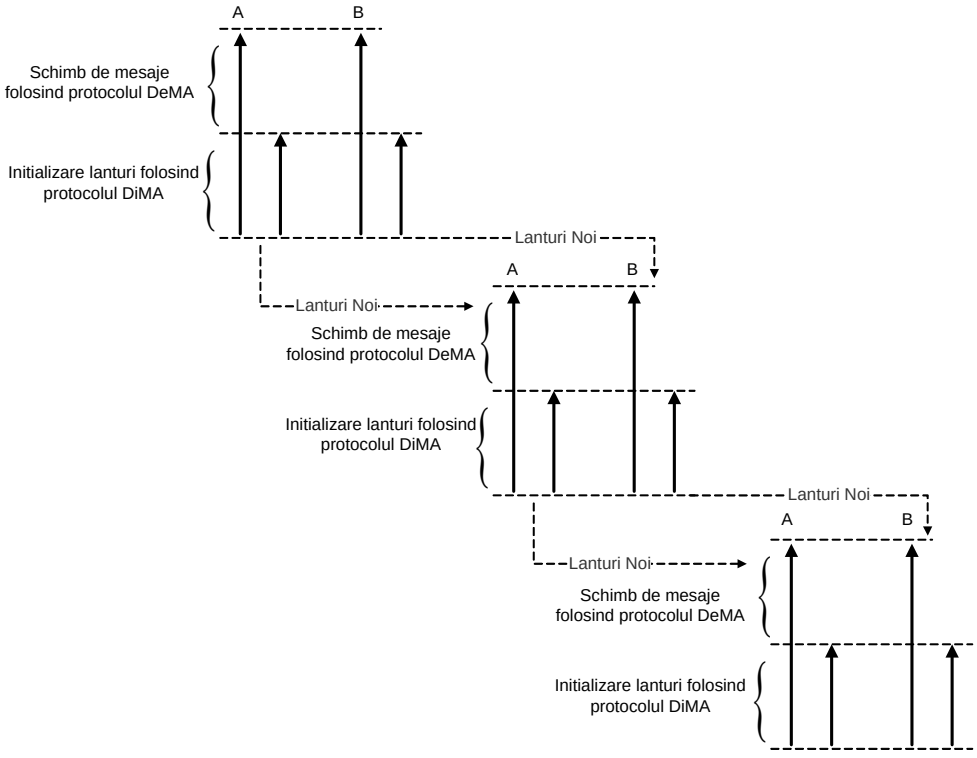


Figura 4.10. Utilizarea lanţurilor one-way în protocolul DeMA-DiCA

4.3.2 Protocolul Delayed Message Authentication cu lanţuri de reziduuri cvadratice (DeMA-QR)

Dacă primitivele simetrice sunt folosite pentru construcţia lanţurilor one-way utilizate în protocolul DeMA, aceste lanţuri pot fi epuizate şi acest lucru este inconvenabil deoarece necesită o reiniţializare care implică un schimb autentificat de cheie. Prin utilizarea funcţiei putere discretă $f(x) = x^{\varepsilon} \bmod n$ această limitare este eliminată din protocolul DeMA deoarece în acest caz lungimea lanţurilor nu influenţează costul de calcul şi astfel pot fi alese lanţuri de dimensiuni nemărginită în practică. În acest caz ambele entităţi participante la schimbul de mesaje vor folosi un modul suficient de mare pentru a nu putea fi factorizat, fie n_A respectiv n_B aceste module, şi două valori aleatoare x_A şi x_B alese de cele două entităţi. Cheile de sesiune sunt definite de relaţiile următoare:

$$\sigma_A(k) = f^{\eta-k}(x_A) = x_A^{2^{\eta-k} \bmod \phi(n_A)} \bmod n_A \tag{4.28}$$

$$\sigma_B(k) = f^{\eta-k}(x_B) = x_B^{2^{\eta-k} \bmod \phi(n_B)} \bmod n_B \tag{4.29}$$

Stadiile protocolului sunt următoarele (este important de reamintit că stadiul de calcul indus în pasul 2 al protocolului poate fi eficient întreprins utilizând soluţia de tip time-memory trade care induce un cost asimptotic de o singură multiplicare modulară):

Generarea Cheilor implică următorii paşi:

1) Valoarea unei constante η care reprezintă numărul maxim de sesiuni de comunicare este fixată de comun acord între cele două entităţi *A* şi *B* (deoarece se utilizează funcţia putere discretă orice valoare poate fi aleasă fără a influenţa costul de calcul).

2) Entitatea *A* alege două numere prime p_A, q_A şi o valoare aleatoare x_A după care calculează $n_A = p_A \cdot q_A$, $\phi(n_A) = (p_A - 1) \cdot (q_A - 1)$ şi $\sigma_A(0)$.

3) Entitatea *B* alege două numere prime p_B, q_B şi o valoare aleatoare x_B după care calculează $n_B = p_B \cdot q_B$, $\phi(n_B) = (p_B - 1) \cdot (q_B - 1)$ şi $\sigma_B(0)$.

4) Entităţile *A* şi *B* se informează reciproc de valorile lui $\sigma_A(0)$, n_A respectiv $\rho_B(0)$, n_B (autenticitatea acestor valori trebuie garantată, în particular ele pot fi schimbate printr-un protocol de schimb autentificat de cheie); toate celelalte valori (numerele prime şi funcţia lui Euler) sunt păstrate secrete de fiecare parte deoarece aflarea lor duce la pierderea totală a securităţii de ambele părţi.

Mesajele din sesiunile $k = 1,...,\eta$:

Runda 1 $A \to B$: $P_{A,k} = \left\{ M_{A,k}, MAC_{\sigma_A(k+1)}\left(M_{A,k}\right), \sigma_A\left(k\right) \right\}$

Runda 2 $B \to A$: $P_{B,k} = \left\{ M_{B,k}, MAC_{\sigma_B(k+1)}\left(M_{B,k}\right), \sigma_B\left(k\right) \right\}$

Acţiunile întreprinse de cei doi participanţi, *A* şi *B*, pe parcursul desfăşurării protocolului pot fi descrise în următorii 6 paşi:

1) Se incrementează numărătorul de sesiune $k = k + 1$.
2) Se generează pachetul pentru sesiunea k (aceasta necesită calcularea unei noi chei de sesiune şi a unui cod MAC).
3) Trimite celeilalte entităţi pachetul pentru sesiunea k .
4) Recepţionează pachetul de la cealaltă entitate.
5) Verifică dacă pachetul primit conţine o cheie autentică de sesiune – aceasta se realizează uşor verificând că $f\left(\sigma\left(k\right)\right) = \sigma\left(k-1\right)$ pentru cheile lui A şi $f\left(\rho\left(k\right)\right) = \rho\left(k-1\right)$ pentru cheile lui B. Dacă cheia de sesiune nu este autentică atunci se revine la pasul 4 altfel se continuă cu pasul 6.
6) Utilizează noua cheie de sesiune pentru a testa autenticitatea mesajului din sesiunea anterioară.

Paşii protocolului pentru sesiunea k sunt de asemenea sugeraţi în figurile 4.11, 4.12. Ambele entităţi A şi B pornesc protocolul cu $k = 0$. Aceeaşi paşi sunt urmaţi de către A şi B dar ordinea este diferită deoarece A porneşte protocolul. În acest sens ordinea în care A urmează paşii este 1, 2, 3, 4, 5, 6 iar pentru B este 4, 5, 6, 1, 2, 3 şi tot aşa pentru terminarea fiecărei sesiuni. Se observă că A porneşte protocolul în pasul 1 în timp ce B porneşte protocolul în pasul 4 iar la epuizarea celor η sesiuni A termină în pasul 6 iar B termina în pasul 3.

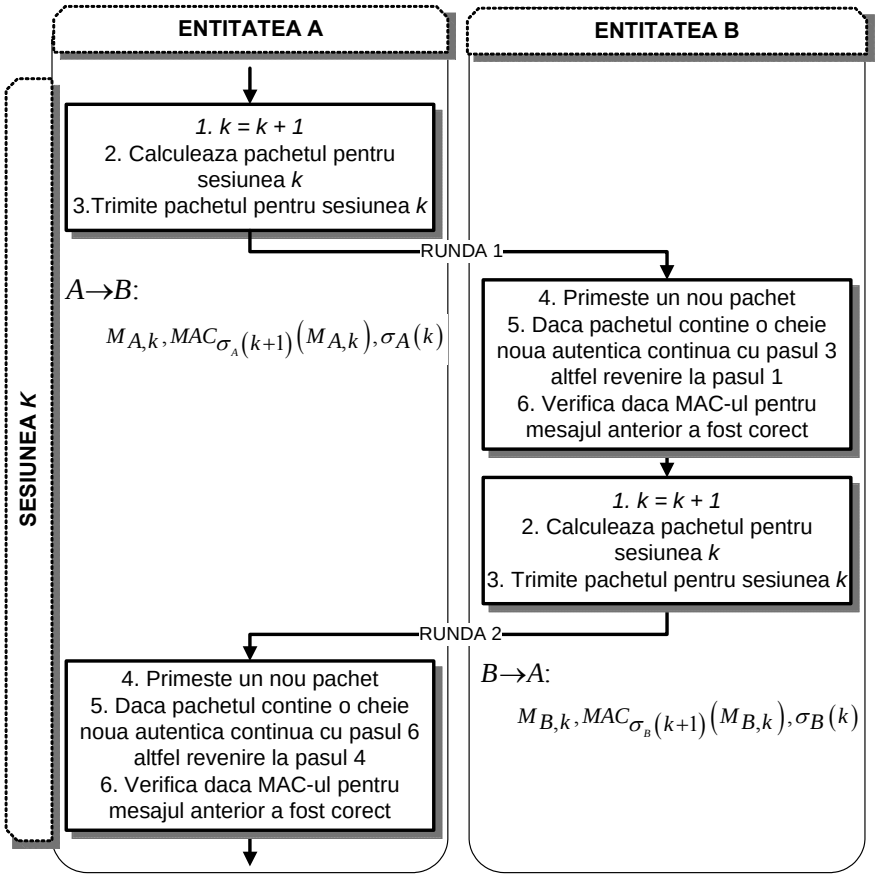


Figura 4.11. Paşii protocolului DeMA în sesiunea *k*

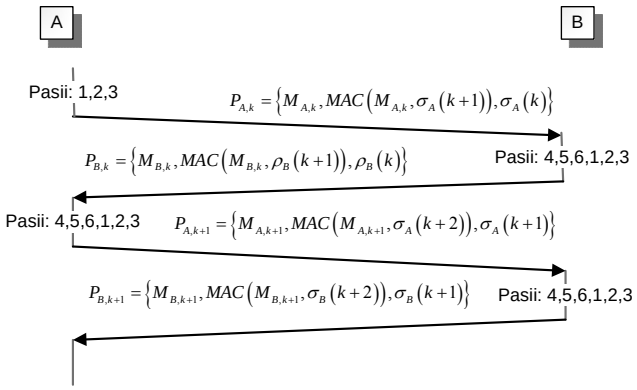


Figura 4.12. Schimbul de mesaje folosind DeMA

- **Performanţa protocolului DeMA-QR**

Deoarece utilizarea primitivelor din criptografia cu cheie publică, cum este funcţia putere discretă, este mai costisitoare din punct de vedere computaţional decât utilizarea primitivelor simetrice este necesar să identificăm cerinţele computaţionale ale protocolului propus.

Timpul de calcul pentru o sesiune a protocolului DeMA depinde de puterea de calcul de pe partea fiecărui participant şi poate fi definit ca:

$$t_{session} = t_{round_A} + t_{round_B} \tag{4.30}$$

Aici prin t_{round_A} şi t_{round_B} sunt notaţi timpii de calcul pentru o rundă de partea lui A respectiv B. Calculele efectuate în fiecare rundă a protocolului DeMA constau în: calculul unei chei de sesiune care necesită o exponenţiere (dacă nu utilizăm soluţia de tip time-memory trade) şi un MAC în pasul 2 plus verificarea noii chei de sesiune care necesită o multiplicare modulară şi un MAC pentru mesajul anterior primit în paşii 5 şi 6. În consecinţă timpul de calcul pentru rundă este dat de următoarea relaţie:

$$t_{round} = t_{exp} + t_{mul} + 2 \cdot t_{MAC} \tag{4.31}$$

Deoarece dintre cele trei componente calculul unei exponenţieri modulare este pe departe cea mai complexă operaţie din punct de vedere computaţional, fără a pierde precizia putem estima timpul de calcul al unei runde ca fiind:

$$t_{round} \approx t_{exp} \tag{4.32}$$

Timpul de calcul poate fi redus în mod substanţial prin utilizarea soluţiei time-memory trade pentru reziduuri cvadratice descrisă în secţiunile anterioare. Astfel, dacă lanţul de η elemente este desfăcut în lanţuri mai mici de λ elemente, timpul de calcul pentru λ runde va fi:

$$t_\lambda = t_{\exp} + (2\lambda - 1) \cdot t_{mul} + 2 \cdot \lambda \cdot t_{MAC} \tag{4.33}$$

De aici timpul mediu de calcul rezultă imediat ca fiind:

$$t'_{round} = \frac{t_\lambda}{\lambda} = \frac{t_{\exp}}{\lambda} + \left(2 - \frac{1}{\lambda}\right) \cdot t_{mul} + 2 \cdot t_{MAC} \tag{4.34}$$

În (4.34) putem ignora timpul de calcul solicitat de MAC, deoarece este de câteva zeci de ori mai mic decât multiplicarea, şi pe măsură ce λ devine mai mare putem estima timpul de calcul ca fiind:

$$t'_{round} \approx 2 \cdot t_{mul} \qquad (4.35)$$

Relaţia (4.35) arată că timpul de calcul pentru o rundă a protocolului DeMA este teoretic redus la timpul de calcul necesar unei multiplicări modulare.

- **Rezultate experimentale**

Paragraful anterior a identificat din punct de vedere teoretic performanţa protocolului propus. În acest paragraf se urmăreşte evaluarea experimentală a performanţei, rezultatele experimentale obţinute fiind în acord cu cele teoretice. În acest sens protocolul a fost implementat în Java folosind NetBeans IDE 5.0. Mediul Java a fost ales datorită suportului oferit pentru lucrul cu întregi de dimensiune mare.

Calculele în grupul de întregi Z_n au fost efectuate folosind metodele oferite de clasa *BigInteger*: *BigInteger multiply(BigInteger val)*, *BigInteger mod(BigInteger m)*, *BigInteger modPow(BigInteger exponent, BigInteger m)*. S-a calculat valoarea lui $x^2 \bmod n$ printr-o multiplicare modulară urmată de o reducere modulo *n*; această metodă a fost preferată ridicării la puterea 2 cu funcţia *modPow* deoarece aceasta s-a dovedit a fi mai lentă.

Utilizatorul *A* al protocolului DeMA joacă rolul de client care se conectează la *B* care reprezintă un server. Arhitectura aplicaţiei este sugerată în figura 4.13, această aplicaţie rulează atât de partea serverului cât şi de partea clientului, diferenţa este că de partea serverului clasa *DeMAClientSC* este înlocuită de clasa *DeMAServerMC* care joacă rol de server. Comunicaţia a fost asigurată prin utilizarea unui obiect *ServerSocket* care acceptă conexiuni de partea serverului şi un obiect *Socket* de partea clientului. Atât de partea clientului cât şi a serverului o instanţă a clasei *DeMACryptoProtocol* este iniţializată şi este utilizată pentru a construi pachetele ce vor fi trimise prin socketurile de comunicaţie. în fiecare instanţă a clasei *DeMACryptoProtocol* există şi o instanţă a clasei *DeMABufferedPKKeyGenerator* respectiv *DeMAGenericPKUser*. Clasa *DeMAGenericPKUser* conţine metoda pentru testarea autenticităţii noii chei de sesiune iar clasa *DeMABufferedPKKeyGenerator* poate fi utilizată pentru generarea noilor chei de sesiune cu soluţia de tip time-memory trade descrisă anterior. Dimensiunea lanţurilor este reprezentată de variabila *bufferSize* definită în clasa *DeMABufferedPKKeyGenerator*. Un obiect *MACGenerator* este de asemenea instanţiat în *DeMACryptoProtocol* pentru generarea şi verificarea MAC-urilor calculate cu funcţia hash SHA1. Arhitectura aplicaţiei este detaliată în figura 4.13.

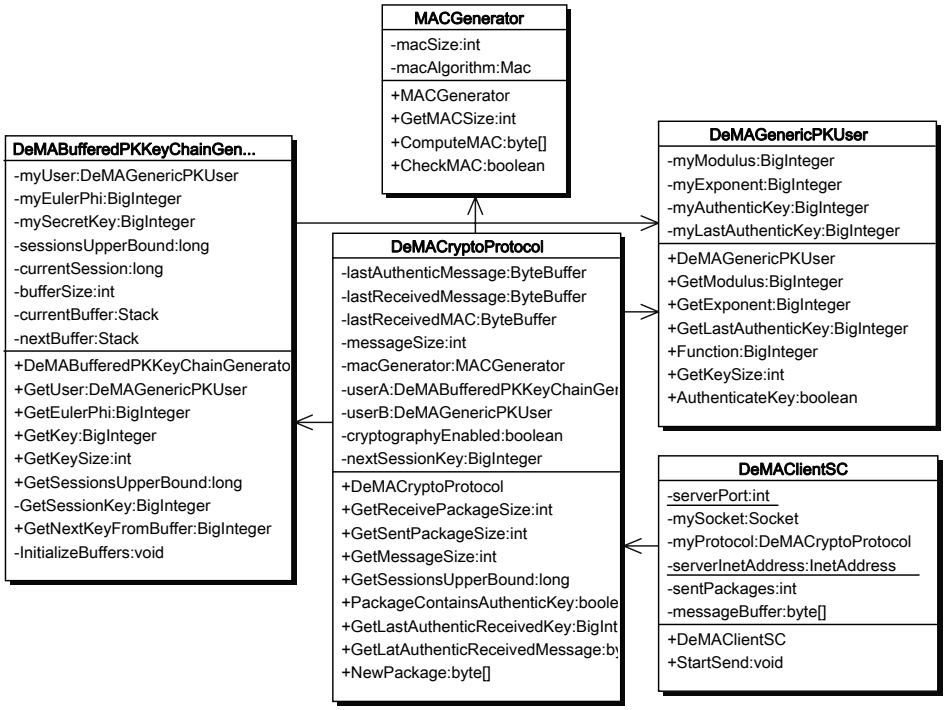


Figura 4.13. Arhitectura aplicaţiei de test a protocolului DeMA-QR

Performanţa protocolului DeMA a fost testată prin rularea implementării Java prezentate pe două computere: un notebook Toshiba Tecra cu procesor Intel Centrino 1.6 Ghz şi un desktop cu procesor AMD Athlon 64 2800+ la 1.8 Ghz. Ambele calculatoare aveau instalat Windows XP şi aveau 512 MB de RAM (aceasta nu este însă foarte relevant pentru timpul de calcul). Pentru început a fost măsurat timpul de calcul pentru primitivele criptografice utilizate din Java calculându-se valoarea medie a unui număr ridicat de rulări ale programului (în jur de 10^6 rulări). În tabelul 4.1 timpii de calcul pentru acestea sunt daţi.

Pentru a testa performanţa protocolului am conectat cele două calculatoare folosind un router TrendNet TW100. Timpii de calcul pentru diferite dimensiuni ale bufferului şi ale lanţului sunt daţi în tabelul 4.2. Pachetele transmise între entităţi constau într-un mesaj de 160 biţi, codul de autentificare MAC de 160 biţi şi o cheie de sesiune de 1024 biţi.

Aşa cum era de aşteptat rezultatele din tabelul 4.2 arată că timpul de calcul este semnificativ îmbunătăţit prin creşterea dimensiunii bufferului. În linia (1) este dat timpul necesar unei sesiuni pentru cazul în care primitivele criptografice sunt activate, i.e. $t_{session}$, în linia (2) este timpul necesar unei sesiuni pentru cazul în care primitivele criptografice sunt dezactivate (acesta este de fapt timpul de comunicare şi este relevant doar ca element de comparaţie). Liniile (3) şi (4) oferă timpii de

calcul off-line, fără comunicare, ce constau în calcularea unei chei de sesiune şi a unui MAC, i.e. t_{round} .

CPU	MAC cu SHA1	Multiplicare Modulară (modul de 1024 biţi)	Exponenţiere Modulară (modul şi exponent de 1024 biţi)
	t_{MAC}	t_{mul}	t_{exp}
Intel Centrino 1.6 Ghz	$10 \times 10^{-6} s$	$74 \times 10^{-6} s$	$50 \times 10^{-3} s$
AMD Athlon 64 2800+ 1.8 Ghz	$8.9 \times 10^{-6} s$	$60 \times 10^{-6} s$	$43 \times 10^{-3} s$

Tabelul 4.1. Timpii de calcul pentru primitivele criptografice utilizate în Java

	Număr de Sesiuni/ Dimensiune Buffer	$10^3 / 1$	$10^3 / 10^2$	$10^5 / 10^3$	$10^5 / 10^4$	$10^6 / 10^4$	$10^6 / 10^5$
(1)	Criptografie Activată $t_{session}$	$40 \times 10^{-3} s$	$1.3 \times 10^{-3} s$	$0.8 \times 10^{-3} s$	$0.7 \times 10^{-3} s$	$0.7 \times 10^{-3} s$	$0.6 \times 10^{-3} s$
(2)	Criptografie Dezactivată	$0.6 \times 10^{-3} s$	$0.6 \times 10^{-3} s$	$0.4 \times 10^{-3} s$	$0.4 \times 10^{-3} s$	$0.4 \times 10^{-3} s$	$0.4 \times 10^{-3} s$
(3)	Comunicare Dezactivată (Centrino 1.6 Ghz) t_{round}	$21 \times 10^{-3} s$	$430 \times 10^{-6} s$	$159 \times 10^{-6} s$	$107 \times 10^{-6} s$	$113 \times 10^{-6} s$	$102 \times 10^{-6} s$
(4)	Comunicare Dezactivată (Athlon 64 2800+ 1.8 Ghz) t_{round}	$18 \times 10^{-3} s$	$422 \times 10^{-6} s$	$139 \times 10^{-6} s$	$94 \times 10^{-6} s$	$96 \times 10^{-6} s$	$87 \times 10^{-6} s$

Tabelul 4.2. Performanţa protocolului DeMA folosind lanţuri de reziduuri cvadratice

Concluzia finală care poate fi trasă din rezultatele experimentale este că utilizarea soluţiei time-memory trade îmbunătăţeşte semnificativ timpii de calcul şi pentru lungimi mari ale bufferului timpii de calcul pentru cazurile în care criptografia este activată sau dezactivată sunt destul de apropiaţi. De exemplu pentru cazul a 10^6 sesiuni la o lungime a bufferului de 10^5 chei performanţa este alterată cu doar

50% de utilizarea criptografiei. Acest rezultat este însă cauzat şi de faptul că dimensiunea cheii criptografice este foarte mare comparativ cu dimensiunea meajului, în cazul în care dimensiunea mesajului ar fi crescută influenţa criptografiei asupra performanţelor ar deveni nesemnificativă. De asemenea cele două linii ale tabelului în care se măsoară performanţa off-line arată că pentru dimensiuni ridicate ale bufferului costul este apropiat de o singură multiplicare modulară – acesta este un rezultat practic care confirmă validitatea relaţiei (4.35).

4.3.3 Protocolul Delayed Message Authentication cu lanţuri de reziduuri cvadratice şi sincronizare temporală (Timed-DeMA-QR)

Protocoalele de autentificare bazate pe lanţuri one-way au înregistrat cel mai mare succes practic în zona protocoalelor de broadcast autentificat deoarece ele fac posibilă utilizarea MAC-urilor în pofida faptului că acestea necesită chei secrete. În acest sens este relevantă construcţia unui protocol bazat pe lanţurile de reziduuri cvadratice introduse anterior care va aduce ca proprietate adiţională faptul că transmisia de broadcast poate dura pe termen nelimitat, lanţul one-way construit astfel fiind nemărginit în practică. Scenariul urmărit presupune existenţa următorilor participanţi: un server de înregistrare RS (Registration Server) şi un număr oarecare de emiţători S (Sender) şi receptori R (Receivers). Fiecare emiţător face publică o informaţie de iniţializare către RS şi ulterior începe să transmită în regim broadcast. Adiţional, dacă diferenţe de ceas între RS şi un anume emiţător S apar şi dinamic, atunci S poate repeta din timp în timp o procedură de sincronizare temporală. Fiecare receptor trebuie să obţină informaţia de iniţializare a emiţătorului de la care doreşte să primească informaţie, să urmeze procedura de sincronizare temporală şi apoi poate să verifice autenticitatea informaţiei primite de la S. Subliniem că între S şi R nu există nici un fel de interacţiune în afară de faptul că S trimite informaţie către R. De asemenea pentru a preveni diferenţe consistente de ceas, receptorii pot urma din timp în timp procedura de resincronizare temporală cu RS. Ca şi în cazul protocolului TESLA este necesară doar o sincronizare temporală slabă, ceea ce înseamnă că receptorii trebuie să aibă garanţia unei margini superioare rezonabile cu privire la timpul de partea lui S. RS nu trebuie să aibă acces la nici o informaţie secretă sau privată, în acest sens el nefiind o entitate cu siguranţă necondiţionată (uncoditionally trusted). Singura cerinţă din partea RS este de a se comporta corect, adică de a fi sigur din punct de vedere funcţional (functionally trusted). Rolul lui RS este deci de a oferi sincronizare temporală şi de a distribui informaţia de iniţializare a emiţătorilor către receptori. Scenariul menţionat poate avea loc pe o perioadă lungă de timp, de exemplu un emiţător S stochează informaţie de partea RS apoi începe să transmită în regim broadcast pentru 5 ani, în toată această perioadă nu este necesară nici o altă interacţiune între emiţători şi serverul de înregistrare, excepţie cazul în care emiţătorul doreşte să îşi sincronizeze ceasul cu serverul de înregistrare.

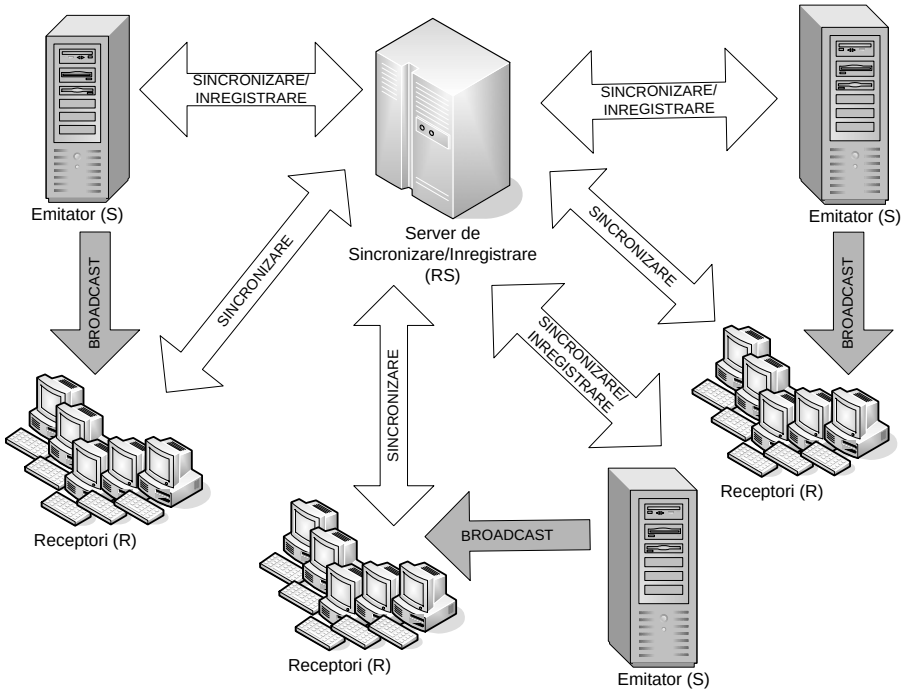


Figura 4.14. Scenariul adresat de protocolul Timed-DeMA-QR

- **Înregistrarea unui emiţător la serverul de înregistrare**

Obiectivul fiecărui emiţător S este de a stabili informaţia de iniţializare la serverul de înregistrare RS. Informaţia constă într-un pachet cu structura $P_{init} = \left(t^{RS}_{broadcast}, S_{id}, n, k_0, \varepsilon_{S,RS}, \delta\right)_{SigS}$ semnat digital de către S. În acest pachet: valoarea $t^{RS}_{broadcast}$ reprezintă valoarea minimă a timpului de partea RS la momentul când S începe broadcastul (în paragraful următor este explicat calculul acestei valori), S_{id} este un identificator al emiţătorului (de exemplu un nume sau o adresa de IP), n este un modul compozit (indicaţii asupra generării acestuia se găsesc în secţiunea cu privire la analiza perioadei lanţului one-way generat de funcţia ridicare a pătrat) şi k_0 este cheia de iniţializare, $\varepsilon_{S,RS}$ este eroarea de sincronizare calculată în conformitate cu indicaţiile de mai jos, δ reprezintă perioada de distribuţie a cheilor iar Sig_S reprezintă faptul că informaţia a fost semnată de către S (ca şi condiţie generală presupunem că fiecare entitate a luat la cunoştinţă cheia publică a fiecărei alte entităţi). Procedura de înregistrare constă în următorii paşi:

1. $S \rightarrow RS: \ Nonce_S$
2. $RS \rightarrow S: \ \left(Nonce_{RS}, Nonce_S, t_{reg}^{RS}\right)_{SigRS}$
3. $S \rightarrow RS: \ \left(Nonce_{RS}, t_{broadcast}^{RS}, S_{id}, n, k_0, \varepsilon_{S,RS}, \delta\right)_{SigS}$

Valoarea $Nonce_S$ este un parametru variant în timp utilizat de S pentru a asigura faptul că răspunsul de la RS nu este o retransmisie a unei informaţii anterioare iar $Nonce_{RS}$ este un nonce utilizat de RS pentru a asigura că informaţia trimisă de S este de asemenea proaspătă, Sig_{RS} denotă faptul că informaţia a fost semnată de către RS. Diferenţa de timp dintre momentul lansării cererii de sincronizare şi momentul primirii unui răspuns de la RS reprezintă eroarea de sincronizare $\varepsilon_{S,RS}$ dintre S şi RS, i.e. $\varepsilon_{S,RS} = t_{reg}^S - t_{start}^S$. Valoarea erorii de sincronizare $\varepsilon_{S,RS}$ trebuie să fie mult mai mică decât perioada de eliberare a cheilor δ, i.e. $\varepsilon_{S,RS} \ll \delta$, aceasta fiind o condiţie naturală pentru eficienţa transmisiei, o explicaţie mai detaliată urmează. Procedura de înregistrare este sugerată de asemenea în figura 4.15.

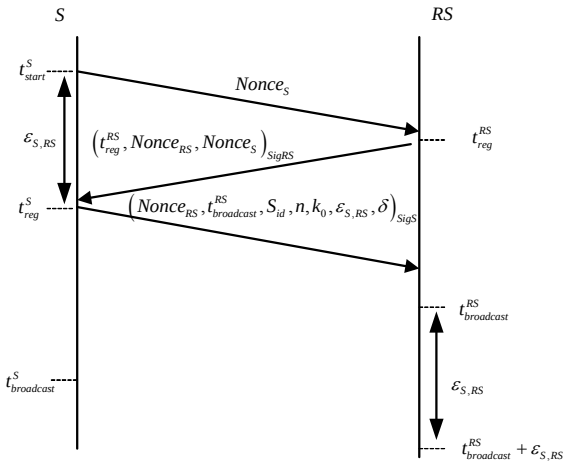


Figura 4.15. Procedura de înregistrare a unui emiţător către serverul de înregistrare

Odată urmată această procedură S poate estima la orice moment de timp t^S valoarea minimă şi maximă a timpului de partea lui RS folosind relaţiile:

$$MinTV^{S,RS}\left(t^S\right) = t^S + t_{reg}^{RS} - t_{reg}^S \tag{4.36}$$

$$MaxTV^{S,RS}\left(t^S\right) = t^S + t_{reg}^{RS} - t_{reg}^S + \varepsilon_{S,RS} \tag{4.37}$$

Presupunând că $t^S_{broadcast}$ este timpul la care emiţătorul S începe broadcastul autentificat, valoarea minimă a timpului de partea lui RS când ceasul lui S indică $t^S_{broadcast}$ poate fi uşor calculată ca $t^{RS}_{broadcast} = MinTV^{S,RS}\left(t^S_{broadcast}\right)$. De asemenea, definim timpul de distribuţie a celei de-a i-a chei ca fiind:

$$DisT^S(i) = t^S_{broadcast} + (i-1)\cdot\delta \tag{4.38}$$

Datorită erorii de sincronizare $\varepsilon_{S,RS}$, prin utilizarea relaţiilor (4.36) şi (4.37) când valoarea timpului de partea lui S este $DisT^S(i)$ valoarea timpului de partea serverului de înregistrare este undeva în intervalul $\left[MinTV^{S,RS}\left(DisT^S(i)\right), MaxTV^{S,RS}\left(DisT^S(i)\right)\right]$; deoarece acesta este intervalul în care cheia i este distribuită vom numi acest interval, interval de distribuţie a cheii i. Este important de remarcat că atâta timp cât condiţia de sincronizare cu eroarea anterior menţionată este păstrată, a i-a cheie nu este eliberată mai devreme de:

$$MDT^{RS}(i) = MinTV^{S,RS}\left(DisT^S(i)\right)$$

$$\Leftrightarrow MDT^{RS}(i) = t^{RS}_{broadcast} + (i-1)\cdot\delta \tag{4.39}$$

Vom numi această valoare Timp Minim de Distribuţie a Cheii pentru a i-a cheie şi o vom nota cu *MDT* (Minimal Disclosure Time). Valoarea lui *MDT* este de interes deoarece în secţiunile următoare se va garanta că pachetul P_i care conţine un MAC calculat cu cheia i+1 nu poate fi falsificat înainte de $MDT^{RS}(i+1)$.

- **Sincronizarea unui receptor cu serverul de sincronizare**

Obiectivul sincronizării receptorului R cu serverul de înregistrare RS este de a obţine valoarea de iniţializare a unui anume emiţător S, i.e. P_{init}, şi de a efectua sincronizarea temporală slabă cu serverul de înregistrare, adică stabilirea unei margini superioare pentru timpul de partea RS. Aceasta va face posibil ca R să poată verifica ulterior autenticitatea informaţiei transmise în regim broadcast de S. Procedura de sincronizare implică următorii paşi:

1. $R \rightarrow RS: S_{id}, Nonce_R$
2. $RS \rightarrow R: \left(Nonce_R, t^{RS}_{sync}, P_{init}\right)_{SigRS}$

$Nonce_R$ este o valoare aleatoare utilizată de R pentru a asigura faptul ca răspunsul de la RS nu este o retransmisie de informaţie expirată şi S_{id} este identificatorul asociat emiţătorului de la care R doreşte să primească informaţie

autentică. Diferenţa de timp dintre paşii 1 şi 2 este eroarea de sincronizare $\varepsilon_{R,RS}$ dintre R şi RS, i.e. $\varepsilon_{R,RS} = t^R_{sync} - t^R_{start}$. Vom impune ca $\varepsilon_{R,RS} + \varepsilon_{S,RS} << \delta$ iar dacă această condiţie nu este respectată atunci sincronizarea trebuie repetată. Această procedură este sugerată de asemenea în figura 4.16.

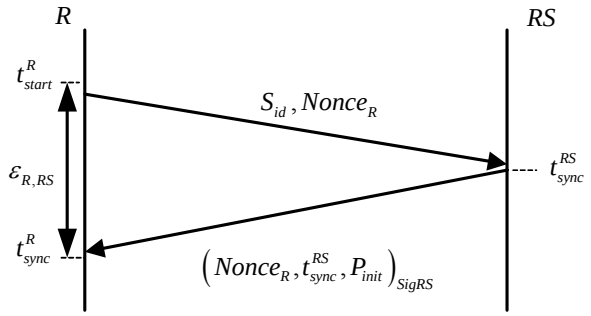


Figura 4.16. Procedura de sincronizare între un receptor şi serverul de înregistrare

După ce procedura de sincronizare este îndeplinită, receptorul R poate estima la orice moment de timp t^R valoarea minimă şi maximă a timpului de partea RS:

$$MinTV^{R,RS}\left(t^R\right) = t^R + t^{RS}_{sync} - t^R_{sync} \tag{4.40}$$

$$MaxTV^{R,RS}\left(t^R\right) = t^R + t^{RS}_{sync} - t^R_{sync} + \varepsilon_{R,RS} \tag{4.41}$$

Folosind aceste relaţii R poate determina valoarea maximă a timpului de partea RS pentru a decide dacă pachetul P_i primit la momentul t^R_i, care conţine un *MAC* calculat cu cheia i+1, este sigur. Aceasta presupune că nu a fost încă distribuită cheia utilizată şi poate fi verificată evaluând condiţia:

$$MaxTV^{R,RS}\left(t^R_i\right) < MDT^{RS}\left(i+1\right) \tag{4.42}$$

Această condiţie verifică faptul că momentul de timp la care a fost primit pachetul *i* este anterior celui de eliberare a pachetului *i+1* care conţine cheia de autentificare. Pentru a preveni eventuale fluctuaţii de ceas, procedura de sincronizare poate fi repetată periodic de către R.

- **Influenţa erorii de sincronizare asupra securităţii**

Datorită erorii de sincronizare $\varepsilon_{S,RS}$ între emiţător şi serverul de iniţializare, cheia k_i este distribuită în cel mai defavorabil caz când valoarea timpului la RS este $MaxTV^{S,RS}\left(DisT^S\left(i\right)\right)$. În acest caz un receptor având eroarea de sincronizare $\varepsilon_{R,RS}$

cu RS ştie că la acest moment valoarea timpului la RS este cel mult $MaxTV^{S,RS}\left(DisT^{S}\left(i\right)\right)+\varepsilon_{R,RS}$. De asemenea trebuie luată în calcul şi întârzierea reţelei pentru un receptor anume R, aceasta fiind durata necesară ca pachetul să circule de la S la R. Desigur, întârzierea reţelei poate varia de la un pachet la altul, dar pentru scopul nostru este suficientă o valoare medie . Fie această întârziere Δ_R. Pentru a verifica condiţia de securitate la ajungerea pachetului este necesară satisfacerea următoarei relaţii:

$$MaxTV^{S,RS}\left(DisT^{S}\left(i\right)\right)+\varepsilon_{R,RS}+\Delta_R < MDT^{RS}\left(i+1\right)$$

$$\Rightarrow \varepsilon_{S,RS}+\varepsilon_{R,RS}+\Delta_R < \delta \qquad (4.43)$$

Deoarece relaţia (4.43) trebuie satisfăcută pentru ca un potenţial receptor să poată primi pachete autentice la o întârziere Δ_R şi erori de sincronizare $\varepsilon_{S,RS}$, $\varepsilon_{R,RS}$ - este necesar ca eroarea de sincronizare să fie mult mai mică (cu câteva ordine de magnitudine) decât perioada de distribuţie a cheilor. Dacă δ este ales de emiţător ca fiind prea mic pentru a satisface (4.43) atunci receptorul va obţine doar pachete la care va trebui sa renunţe deoarece condiţia de securitate nu va fi verificată. Este util de menţionat că o potenţială îmbunătăţire, pentru cazul în care această constrângere nu poate fi satisfăcută, a fost propusă de Perrig et al. în cadrul protocolului TESLA. Aceştia au propus a distribui cheia utilizată pentru calculul codului MAC asociat pachetului P_i doar într-un pachet mai îndepărtat $P_{i+\tau}$ şi nu direct în pachetul P_{i+1}.

- **Resincronizarea între un emiţător şi serverul de înregistrare**

Asigurarea faptului că diferenţele de ceas între emiţător şi serverul de înregistrare sunt neglijabile este o problemă critică a comunicării. Pentru ca securitatea să fie asigurată, emiţătorul trebuie să se asigure că la orice moment t^S valoarea timpului de la RS este între valorile minime şi maxime date în relaţiile (4.36) şi (4.37) – această condiţie fiind necesară pentru distribuţia cheilor în intervalul corect. Dacă emiţătorul suspectează diferenţe de ceas între ceasul său şi al serverului de înregistrare atunci există două potenţiale măsuri de securitate. Prima soluţie este ca emiţătorul să repete procedura de înregistrare şi să înlocuiască valorile vechi de înregistrare cu valori noi de partea RS (aceasta înseamnă repornirea întregului protocol), această soluţie este ineficientă pentru receptori care deja au obţinut valorile de iniţializare. Cea de-a doua soluţie este ca emiţătorul să urmeze o procedură de resincronizare cu RS. La timpul t^S emiţătorul poate estima că timpul de partea serverului de înregistrare este între valorile $MinTV^{S,RS}\left(t^S\right)$ şi $MaxTV^{S,RS}\left(t^S\right)$ prin utilizarea relaţiilor (4.36), (4.37). Pentru obţinerea unei noi sincronizări emiţătorul poate urma procedura de sincronizare descrisă pentru receptori. Emiţătorul joacă rolul unui receptor şi după terminarea procedurii de sincronizare poate estima că valoarea timpului de partea serverului de înregistrare este între $MinTV^{R,RS}\left(t^R\right)$ şi $MaxTV^{R,RS}\left(t^R\right)$ prin utilizarea relaţiilor (4.40) şi (4.41).

Presupunem că $\varepsilon_{R,RS} \le \varepsilon_{S,RS}$, această condiţie fiind necesară pentru ca noua sincronizare să fie mai exactă decât cea anterioară. Acum emiţătorul poate calcula valoarea de ajustare $\xi = MinTV^{R,RS}\left(t^R\right) - MinTV^{S,RS}\left(t^S\right)$ (aici $t^S = t^R$ deoarece emiţătorul şi receptorul sunt în acest caz aceeaşi entitate) şi poate folosi această valoare de ajustare pentru a distribui pachete la timpul $t^S_{broadcast} + (i-1)\cdot\delta + \xi$ în loc de $t^S_{broadcast} + (i-1)\cdot\delta$. Cazul în care $\varepsilon_{R,RS} > \varepsilon_{S,RS}$ trebuie evitat deoarece în anumite situaţii emiţătorul nu poate fi sigur dacă estimarea iniţială este sau nu corectă raportat la noua estimare. De asemenea cel mai bun lucru care poate fi făcut în acest caz este asigurarea faptului că pachetele nu sunt distribuite prea devreme prin aplicarea aceleiaşi metodologii de ajustare. În acest caz însă, în anumite situaţii pachetele vor fi distribuite prea târziu iar unii receptori vor trebui să renunţe la ele deoarece nu verifică condiţia de securitate.

- **Descrierea protocolului**

Cu toate că funcţia putere discretă este mai solicitantă din punct de vedere computaţional decât o funcţie hash şi dimensiunea rezultatului are un număr mai ridicat de biţi, are avantajul că lanţul obţinut poate avea orice dimensiune fără a influenţa costul computaţional, din aceste motive lanţul poate fi eficient utilizat în scenariul nostru de broadcast. Descrierea protocolului devine imediată odată introduse procedurile de sincronizare şi tehnicile de calcul a lanţului de chei.

- Etapă de iniţializare:

Emiţători:

1) Se fixează numărul de sesiuni de comunicare η. În principiu, valoarea lui η poate fi calculată ca $\eta = T / \delta$ unde T reprezintă durata întregii transmisii, δ reprezintă intervalul de distribuţie a cheii. Deoarece protocolul este destinat unei transmisii pe perioade lungi sau incerte, orice valoare pentru η poate fi aleasă, de exemplu, putem alege $\eta = 2^{128}$ care va conduce la un lanţ nelimitat în practică.

2) Se aleg două numere prime speciale p şi q în conformitate cu specificaţiile din secţiunea anterioară pentru a obţine o dimensiune a perioadei suficient de mare.

3) Se generează x_0 aleator.

4) Se calculează $n = p \cdot q$, $\phi(n) = (p-1)\cdot(q-1)$, $k_0 = x_0^{2^\eta \bmod \phi(n)} \bmod n$.

5) Se foloseşte protocolul de înregistrare al emiţătorului pentru a fixa pachetul de iniţializare $P_{init} = \left(t^{RS}_{broadcast}, S_{id}, n, k_0, \varepsilon_{S,RS}, \delta\right)_{SigS}$ pe serverul de înregistrare.

6) Se fixează valoarea de ajustare $\xi = 0$ descrisă în secţiunea cu privire la resincronizarea emiţătorului.

Receptori:

1) Se foloseşte procedura de sincronizare temporală descrisă în secţiunea 2.3 pentru a obţine o margine superioară a timpului de pe partea serverului de înregistrare şi pachetul de iniţializare al unui emiţător.

- Stadiul de comunicare:

Emiţători:

1) Se trimite către toţi clienţii la momentul $t_{broadcast}^{S} + (i-1)\cdot\delta + \xi$ (în regim broadcast) pachetul $P_i = \left\{ i, M_i, MAC_{KD(k_{i+1})}(M_i), k_i \right\}, i = \overline{1,\eta}$. Aici M_i reprezintă mesajul de broadcast şi k_i denotă cheia de sesiune, deci $k_i = x_A^{2^i \bmod \phi(n)} \bmod n$.

Receptori:

1) Dacă pachetul P_i este primit la timp, ceea ce înseamnă că relaţia (4.41), condiţia de securitate, este verificată, atunci se stochează mesajul şi MAC-ul, iar în caz contrar se renunţă la acestea. Se verifică autenticitatea fiecărei chei de sesiune verificând că $k_i = f^{i-l}(k_l)$, aici k_l este ultima cheie autentică primită; în cel mai defavorabil caz se va utiliza la verificare cheia de iniţializare k_0. Dacă este autentică cheia primită atunci aceasta poate fi folosită pentru verificarea autenticităţii pachetelor din sesiuni anterioare.

Ca potenţială îmbunătăţire a protocolului subliniem că emiţătorul poate trimite mai multe pachete autentificate cu cheia k_{i+1} până la momentul la care această cheie expiră, adică $MDT^{RS}(i+1)$. De exemplu, poate trimite pachete cu structura $P_i = \left\{ i, j, M_{i,j}, MAC_{KD(k_{i+1})}(M_{i,j}, j), k_i \right\}, j = \overline{1,r}$, aici r este numărul de mesaje autentificate cu aceeaşi cheie. $KD(k_{i+1})$ reprezintă o funcţie de derivare a cheii cu ajutorul căreia se calculează cheia MAC-ului din cheia de sesiune.

4.4 Construcţia unui KEM bazat pe extensia schemei RSA

4.4.1 Extensia schemei RSA (ExtRSA)

În această secţiune vom introduce o generalizare a schemei RSA care oferă o permutare de tip one-way cu trapă ce poate fi utilizată în cadre de încapsulare a cheii. Se obţine astfel un criptosistem hibrid în care o primitivă asimetrică este utilizată pentru a cripta o cheie cu care se criptează un mesaj folosind o primitivă

simetrică. În esenţă vom folosi o generalizare a funcţiei putere discretă, funcţie ale cărei cazuri particulare sunt utilizate în criptosisteme precum RSA şi Rabin. Pentru ca expunerea să fie cât mai concisă vom folosi următoarele notaţii:

$$\lambda = cmmdc(\varepsilon, p-1) \tag{4.44}$$

$$\mu = cmmdc(\varepsilon, q-1) \tag{4.45}$$

$$\tau_{\min} = \min\left\{\tau \in N \,\middle|\, cmmdc\left(\frac{\phi(n)}{cmmdc(\phi(n), \varepsilon^{\tau_{\min}})}, \varepsilon\right) \neq 1\right\} \tag{4.46}$$

$$\phi'(n) = \frac{\phi(n)}{cmmdc(\phi(n), \varepsilon^{\tau_{\min}})} \tag{4.47}$$

$$\delta' \equiv \varepsilon^{-1} \bmod \phi'(n) \tag{4.48}$$

$$g(x) = x^{\delta'} \bmod n,\ g: Z_n^* \to Z_n^* \tag{4.49}$$

Aşa cum se poate observa, $\tau_{\min}$ este valoarea minimă pentru care relaţia $cmmdc\left(\frac{\phi(n)}{cmmdc(\phi(n), \varepsilon^{\tau_{\min}})}, \varepsilon\right) \neq 1$ este verificată. Valorile λ şi μ sunt cei mai mari divizori comuni dintre exponentul ε şi cele două valori al căror produs este $\phi(n)$. Este important de remarcat că ε este relativ prim la $\phi'(n)$ şi admite ca invers multiplicativ în $Z_{\phi'(n)}$ pe δ'. În cele ce urmează, în teorema 4.3, va fi demonstrat că funcţia $g(x)$ este inversa lui $f(x)$ în anumite subgrupuri din Z_n.

Definiţia 4.1. Definim Z_n^e mulţimea reziduurilor de ordin e modulo n, i.e. $Z_n^e = \left\{x \in Z_n^* \,\middle|\, \exists y \in Z_n^*, x = y^e \bmod n\right\}$.

Teorema 4.2. Toate elementele din $Z_n^{\varepsilon^{\tau_{\min}}}$ au exact o rădăcină de ordin ε care este în $Z_n^{\varepsilon^{\tau_{\min}}}$.

Demonstraţie. Putem demonstra că pentru $\forall a \in Z_n^{\varepsilon^{\tau_{\min}}}$ numărul $b = a^{\delta'} \bmod n$ este rădăcina sa de ordin ε. Acest lucru rezultă dintr-un simplu calcul deoarece din moment ce $a \in Z_n^{\varepsilon^{\tau_{\min}}}$ urmează că există c astfel încât $a \equiv c^{\varepsilon^{\tau_{\min}}} \bmod n$ şi atunci $b^{\varepsilon} \equiv \left(a^{\delta'}\right)^{\varepsilon} \equiv c^{\varepsilon^{\tau_{\min}+1} \cdot \delta'} \bmod n$. Cum $\varepsilon \cdot \delta' \equiv 1 \bmod \phi'(n) \Rightarrow \varepsilon \cdot \delta' = 1 + k \cdot \phi'(n)$ avem $b^{\varepsilon} \equiv c^{\varepsilon^{\tau_{\min}+1} \cdot \delta'} \equiv c^{\left(1+k\phi'(n)\right) \cdot \varepsilon^{\tau_{\min}}} \equiv c^{\varepsilon^{\tau_{\min}}} \equiv a \bmod n$ deci b este rădăcină de ordin ε. Totodată,

faptul că $b \equiv a^{\delta'} \equiv \left(c^{\varepsilon^{r_{\min}}}\right)^{\delta'} \equiv \left(c^{\delta'}\right)^{\varepsilon^{r_{\min}}} \bmod n$ demonstrează că $b \in Z_n^{\varepsilon^{r_{\min}}}$. Acum demonstrăm prin reducere la absurd că b este unic. Presupunem că $d \in Z_n^{\varepsilon^{r_{\min}}}$ şi $d^{\varepsilon} \equiv a \bmod n$ dar $d \neq b \bmod n$. Din moment ce $b, d \in Z_n^{\varepsilon^{r_{\min}+1}}$ trebuie să existe t, u astfel încât $b \equiv t^{\varepsilon^{r_{\min}+1}} \bmod n$, $d \equiv u^{\varepsilon^{r_{\min}+1}} \bmod n$. Deoarece $b^{\varepsilon} \equiv d^{\varepsilon} \bmod n$ avem $t^{\varepsilon^{r_{\min}+1}} \equiv u^{\varepsilon^{r_{\min}+1}} \bmod n$ şi ridicând la puterea δ' obţinem $\left(t^{\varepsilon^{r_{\min}+1}}\right)^{\delta} \equiv \left(u^{\varepsilon^{r_{\min}+1}}\right)^{\delta} \bmod n \Rightarrow t^{\varepsilon^{r_{\min}}} \equiv u^{\varepsilon^{r_{\min}}} \bmod n \Leftrightarrow b \equiv d \bmod n$ ceea ce încheie demonstraţia.

Teorema 4.3. Funcţia $f(x) = x^{\varepsilon} \bmod n$, $f : Z_n^{\varepsilon^{r_{\min}}} \to Z_n^{\varepsilon^{r_{\min}}}$ este o permutare one-way cu trapă şi $g(x) = x^{\delta'} \bmod n$, $g : Z_n^{\varepsilon^{r_{\min}}} \to Z_n^{\varepsilon^{r_{\min}}}$ este inversa sa, i.e. $f(g(x)) = x, \forall x \in Z_n^{\varepsilon^{r_{\min}}}$.

Demonstraţie. În primul rând observăm că deoarece $\forall a \in Z_n^{\varepsilon^{r_{\min}}}$ există un întreg b astfel încât $a = b^{\varepsilon^{r_{\min}}} \bmod n$ iar $f(a) = a^{\varepsilon} \bmod n = \left(b^{\varepsilon}\right)^{\varepsilon^{r_{\min}}} \bmod n$ şi deci $f(a) \in Z_n^{\varepsilon^{r_{\min}}}$. În al doilea rând demonstrăm că g este inversa lui f. Presupunem că f este aplicată asupra unui întreg oarecare $a \in Z_n^{\varepsilon^{r_{\min}}}$. În conformitate cu definiţia lui $Z_n^{e^{th}}$ există un întreg b astfel încât $a = b^{\varepsilon^{r_{\min}}} \bmod n$. Aceasta înseamnă că $g(f(a)) = a^{\varepsilon \cdot \delta'} \bmod n = b^{\varepsilon^{r_{\min}+1} \cdot \delta'}$ şi cum $\varepsilon \cdot \delta' \equiv 1 \bmod \phi'(n) \Rightarrow \varepsilon \cdot \delta' = 1 + k \cdot \phi'(n)$ urmează că $g(f(a)) = a^{\varepsilon \cdot \delta'} \bmod n = b^{\varepsilon^{r_{\min}} \cdot \left(1 + k\phi'(n)\right)} = b^{\varepsilon^{r_{\min}} + k \cdot \varepsilon^{r_{\min}} \cdot \phi'(n)}$. Acum, din moment ce $\varepsilon^{r_{\min}} \cdot \phi'(n) \equiv 0 \bmod \phi(n)$, obţinem $g(f(a)) = b^{\varepsilon^{r_{\min}}} b^{l \cdot \phi(n)} \bmod n = b^{\varepsilon^{r_{\min}}} = a$ - şi aceasta încheie demonstraţia. Ultimul lucru care mai trebuie demonstrat pentru a ne convinge că f este într-adevăr o permutare este că $\forall a \neq b, a \in Z_n^{\varepsilon^{r_{\min}}}, b \in Z_n^{\varepsilon^{r_{\min}}}$ avem $f(a) \neq f(b)$. Aceasta se demonstrează simplu prin reducere la absurd, presupunând că $f(a) = f(b)$ iar acum prin aplicarea lui g urmează că $g(f(a)) \equiv g(f(b)) \bmod n \Rightarrow a \equiv b \bmod n$ ceea ce este o contradicţie cu ipoteza $a \neq b$.

Următorul corolar a fost demonstrat în preambulul teoremei 4.2 şi este şi o consecinţă naturală a teoremei 4.3 ce conduce la calculul rădăcinii de ordin ε.

Corolar 4.4. Dacă $a \in Z_n^{\varepsilon^{r_{\min}}}$ atunci $a^{\delta'}$ este o rădăcină de ordin ε a lui a.

Se poate observa că în cazul în care exponentul este prim relativ la ordinul grupului, i.e. cazul exponenţilor RSA, atunci $Z_n^{\varepsilon^{r_{\min}}} = Z_n^*$. În acest caz criptosistemul RSA este mai eficient decât propunerea de încapsulare ce urmează, dar pentru acest caz potenţiala echivalenţă cu problema factorizării întregilor nu poate fi demonstrată.

➢ **Extensia schemei ExtRSA**

Prin utilizarea directă a preambulului creat se poate obţine o generalizare a criptosistemului RSA care funcţionează pentru orice valoare a exponentului ε (amintim că în cazul RSA exponentul trebuie să fie relativ prim la ordinul grupului). Descrierea criptosistemului cu cheie publică rezultat prin generalizarea funcţiei RSA este următoarea:

ExtRSA.Gen$\left(1^k\right)$: Se generează două numere prime aleatoare p, q şi se calculează $n = pq$ respectiv $\varphi(n) = (p-1)(q-1)$ (presupunem că numerele p şi q au fost generate în aşa fel încât n are k biţi). Generează un întreg ε astfel încât $c.m.m.d.c.\left(\varepsilon, \phi(n)\right) \neq 1$ (se impune această condiţie pentru a face diferenţa faţă de RSA şi a echivala securitatea cu problema factorizării), calculează $\tau_{\min}$ ca fiind valoarea minimă pentru care egalitatea $cmmdc\left(\frac{\phi(n)}{cmmdc\left(\phi(n), \varepsilon^{\tau_{\min}}\right)}, \varepsilon\right) \neq 1$ este satisfăcută. Calculează $\phi'(n) = \frac{\phi(n)}{cmmdc\left(\phi(n), \varepsilon^{\tau_{\min}}\right)}$ şi δ' astfel încât $\delta' \cdot \varepsilon \equiv 1 \bmod \phi'(n)$. Cheia publică este $PK = \left(n, \varepsilon, \tau_{\min}\right)$ iar cea privată este $SK = \left(n, \delta'\right)$.

ExtRSA.Enc$\left(m, PK\right)$: Alege un întreg aleator r şi calculează $c_1 = f^{\tau_{\min}+1}(r) = r^{\varepsilon^{\tau_{\min}+1}} \bmod n$ şi $c_2 = f^{\tau_{\min}}(r) \cdot m \bmod n = r^{\varepsilon^{\tau_{\min}}} \cdot m \bmod n$. Returnează criptotextul $c \leftarrow (c_1, c_2)$.

ExtRSA.Dec$\left(c, SK\right)$: Calculează valoarea lui $m = \left(c_1^{\delta'}\right)^{-1} \cdot c_2 \bmod n$. Returnează mesajul m.

4.4.2 Utilizarea ExtRSA într-un cadru KEM-DEM (ExtRSA-KEM)

Sistemele criptografice cu cheie publică sunt desigur mult mai costisitoare computaţional decât cele cu cheie privată şi acest lucru ar limita serios utilizarea lor în practică. În acest sens criptarea hibridă este o soluţie excelentă pentru a rezolva neajunsul. Cadrul de criptare hibridă denumit KEM-DEM (Key Encapsulation Mechanism – Data Encryption Mechanism), propus de către Shoup [111], este o abordare eficientă care a condus la o soluţie sigură şi eficientă pentru RSA. Acelaşi cadru poate fi aplicat şi în cazul permutării one-way cu trapă obţinută prin generalizarea anterioară. Descrierea completă a mecanismului de încapsulare a cheii, KEM, este următoarea:

$KEM.Gen(1^k)$: Alege două numere prime distincte de k biţi p,q şi un exponent întreg ε. Calculează $n = p \cdot q$, $\phi(n)$, $\tau_{\min}$, $\phi'(n)$, δ. Returnează $pk = (n, \varepsilon, \tau_{\min})$ şi $sk = (n, \delta)$.

$KEM.Key(pk)$: Alege un întreg aleator $\omega \in Z_n$, calculează $\gamma = \omega^{\varepsilon^{\tau_{\min}}} \bmod n$ şi cheia pentru DEM ca $dk = KDF(\gamma)$. Returnează (γ, dk).

$KEM.Enc(\gamma)$: Calculează $\psi = \gamma^{\varepsilon} \bmod n$. Returnează ψ.

$KEM.Dec(\psi)$: Calculează $\gamma = \psi^{\delta} \bmod n$ şi $dk = KDF(\gamma)$. Returnează (dk).

Prin *KDF* desemnăm o funcţie de derivare a cheii. Securitatea acestui cadru KEM/DEM în faţa adversarilor activi presupune că ambele componente KEM şi DEM sunt sigure în faţa adversarilor activi (desigur este o intuiţie bună că dacă ambele componente sunt sigure va fi sigur şi cadrul KEM/DEM obţinut). Dacă partea DEM respectă doar cerinţe scăzute de securitate în faţa adversarilor activi atunci un cadru Tag-KEM/DEM trebuie utilizat. Aceasta este o îmbunătăţire imediată care constă în introducerea unei etichete (Tag) în KEM pentru a asigura non-maleabilitatea componentei DEM. Putem aplica această îmbunătăţire şi în cazul de faţă şi vom obţine următoarul cadru de tip Tag-KEM/DEM:

$TKEM.Enc(\gamma, T)$: Calculează $\psi = \gamma^{\varepsilon} \bmod n$, $\varsigma = H(\gamma, T)$. Returnează (γ, ς).

$TKEM.Dec(\psi, \varsigma, T)$: Calculează $\gamma = \psi^{\delta'} \bmod n$ şi $dk = KDF(\gamma)$. Dacă $H(\gamma, T) \neq \varsigma$ returnează $\perp$ altfel returnează (dk).

În această construcţie eticheta T folosită este chiar criptarea simetrică a mesajului ce se doreşte transmis, i.e. $T = Sym.Enc_{dk}(m)$, iar H este o funcţie hash.

4.5 Demonstraţii formale de securitate pentru ExtRSA-KEM şi DeMA

Toate protocoalele şi sistemele introduse anterior necesită demonstraţii formale de securitate pentru a arăta că prezintă proprietăţi de securitate precum IND şi NM în faţa unor adversari activi. Datorită echivalenţei între IND-CCA2 şi NM-CCA2 în cele ce urmează vom demonstra doar securitatea în faţa adversarului IND-CCA2. Mecanismul de bază al demonstraţiilor de securitate din criptografie este reducţia în timp polinomial între algoritmi a cărei definiţie o introducem în cele ce urmează.

Definiţia 4.5. (Reducţie în timp polinomial) Spunem că problema Π_1 se reduce în timp polinomial la Π_2 şi notăm $\Pi_1 <_P \Pi_2$ dacă există un algoritm A_1 care rezolvă Π_1 ce primeşte ca subrutină un algoritm A_2 care rezolvă Π_2 iar complexitatea lui A_1 este polinomială dacă complexitatea lui A_2 este şi ea polinomială.

Reducţiile în timp polinomial fac posibilă demonstrarea securităţii unui criptosistem prin găsirea unei reducţii între un algoritm de spargere a criptosistemului, fie A_{Adv}, şi un algoritm de rezolvare a unei probleme despre care se ştie că este intractabilă, fie A_{Intr}, i.e. $A_{Intr} <_P A_{Adv}$. În această secţiune în scopul demonstrării securităţii schemelor propuse vor fi făcute reducţii între adversarul IND-CCA2 asupra ExtRSA-KEM respectiv Timed-DeMA-QR şi problema factorizării (IFP). Toate demonstraţiile vor fi făcute în ROM.

4.5.1 Reducţia pentru ExtRSA-KEM

Se doreşte demonstrarea echivalenţei între securitatea ExtRSA-KEM şi problema factorizării. Pentru aceasta întâi se va stabili echivalenţa între calculul rădăcinii de ordin ε şi factorizare, apoi se va face reducţia de la IND-CCA2 la calculul rădăcinii de ordin ε.

- **Reducţia de la calculul rădăcinii de ordin ε la factorizare**

Marele avantaj al generalizării funcţiei RSA este faptul că securitatea criptosistemului (posibilitatea de a decripta mesaje) poate fi demonstrată ca fiind echivalentă cu problema factorizării întregilor atunci când $cmmdc\left(\varepsilon, \phi(n)\right) \neq 1$. Acest lucru nu poate fi demonstrat pentru cazul particular al RSA, mai mult, o lucrare recentă arată că o astfel de echivalenţă s-ar putea să nici nu existe [17]. Pentru cazul în care $\varepsilon = 2$, i.e. criptosistemul Rabin, este binecunoscută această echivalenţă, ceea ce interesează acum este o echivalenţă pentru cazul general când exponentul nu este prim relativ la ordinul grupului. În primul rând amintim un rezultat bine cunoscut din teoria numerelor - Teorema Chineză a Resturilor:

Teorema 4.6. (Teorema Chineză a Resturilor): Fie întregii $n_1, n_2, ..., n_k$ astfel încât $cmmdc\left(n_i, n_j\right) = 1, \forall i \neq j, 0 < i \leq k, 0 < j \leq k$. Atunci următorul sistem de congruenţe are o soluţie unică x în Z_n unde $n = n_1 n_2 ... n_k, a_i \in Z_{n_i}$:

$$\begin{cases} x \equiv a_1 \bmod n_1 \\ x \equiv a_2 \bmod n_2 \\ \quad ..... \\ x \equiv a_k \bmod n_k \end{cases}$$

Mai mult, soluţia acestui sistem poate fi calculată, folosind algoritmul lui Gauss, ca fiind $x = \sum_{i=1}^{k} a_i N_i M_i \bmod n$ unde $N_i = \frac{n}{n_i}$ iar $M_i = N_i^{-1} \bmod n_i$.

Remarca 4.7. Dacă p este prim şi $d \mid p-1$ atunci ecuaţia $x^d \equiv 1 \bmod n$ are exact d soluţii.

Numărul de soluţii al ecuaţiei $x^\varepsilon \equiv 1 \bmod n$ poate fi acum uşor calculat folosind 4.6. şi 4.7.

Teorema 4.8. Ecuaţia $x^\varepsilon \equiv 1 \bmod n$ are exact $\lambda \cdot \mu$ soluţii distincte şi pentru $\lambda + \mu - 2$ dintre ele este adevărat că $cmmdc\left(x_0 - 1, n\right)$ este un factor non-banal al lui n (aici x_0 denotă soluţia ecuaţiei).

Demonstraţie. În conformitate cu remarca 4.7 numărul de soluţii ale ecuaţiei $x^{\varepsilon} \equiv 1 \bmod p$ este λ iar numărul de soluţii ale ecuaţiei $x^{\varepsilon} \equiv 1 \bmod q$ este μ (este de remarcat că de fapt interesează numărul de soluţii al ecuaţiei $x^{\lambda} \equiv 1 \bmod p$ şi $x^{\mu} \equiv 1 \bmod q$ deoarece $x^{\varepsilon/\lambda} \equiv 1 \bmod p$ respectiv $x^{\varepsilon/\mu} \equiv 1 \bmod q$ au o singură soluţie). Acum, datorită izomorfismului descris de Teorema Chineză a Resturilor, fiecare soluţie din Z_p^* poate fi grupată cu fiecare soluţie din Z_q^* şi avem exact $\lambda \cdot \mu$ soluţii distincte. Evident, dacă $x \equiv 1 \bmod p$ şi $x \neq 1 \bmod q$ sau $x \neq 1 \bmod p$ şi $x \equiv 1 \bmod q$ atunci valoarea lui $cmmdc(x_0 - 1, n)$ cu x_0 soluţie a ecuaţiei este un factor non-banal al lui n. În mod evident pentru exact $\mu - 1$ soluţii avem $x \equiv 1 \bmod p$ şi $x \neq 1 \bmod q$ iar pentru $\lambda - 1$ soluţii avem $x \neq 1 \bmod p$ şi $x \equiv 1 \bmod q$. Aceasta conduce la un total de $\lambda + \mu - 2$ soluţii care vor da un factor non-banal al modulului la calculul valorii $cmmdc(x_0 - 1, n)$.

Următoarea teoremă oferă reducţia de securitate între calculul rădăcinii şi problema factorizării întregilor.

Teorema 4.9. Dacă există un algoritm A_{ε} pentru calculul rădăcinilor de ordin ε în $Z_n^{\varepsilon^{\tau_{\min}}}$ atunci există un algoritm A_{fact} care după un apel la A_{ε} returnează un factor al lui n cu probabilitatea $\mathrm{Pr}_{fact} = \left(1 - \frac{1}{\lambda \cdot \mu}\right) \cdot \left(\frac{\lambda + \mu - 2}{\lambda \mu}\right)$.

Demonstraţie. Presupunem că A_{ε} se comportă după cum urmează: la intrarea x returnează $s \in Z_n^{\varepsilon^{\tau_{\min}}}$ astfel încât $x = s^{\varepsilon} \bmod n$ sau $\perp$ dacă un astfel de s nu există. Atunci A_{fact} va lucra după cum urmează: generează un întreg aleator $r \in Z_n$, setează $a_0 = r$, calculează $a_1 = r^{\varepsilon} \bmod n$ x şi îl trimite ca intrare lui A_{ε}. Dacă A_{ε} returnează r (rădăcina deja cunoscută) o nouă valoare $r \in Z_n$ este selectată şi paşi sunt repetaţi. Se observă că probabilitatea ca A_{ε} să returneze r este $1 / \lambda \cdot \mu$ deoarece rădăcinile lui a_1 sunt uniform distribuite în Z_n. Altfel, dacă nu a returnat r, A_{ε} va returna fie o rădăcină distinctă v fie $\perp$. Dacă A_{ε} returnează $\perp$ atunci A_{fact} calculează secvenţa $a_i = \left(a_{i-1}\right)^{\varepsilon} \bmod n, a_0 = r, i > 0$ şi oferă consecutiv fiecare valoare $a_1, a_2, \ldots$ ca intrare lui A_{ε} până când A_{ε} nu mai returnează $\perp$ şi returnează s astfel încât $a_i = s^{\varepsilon} \bmod n$ (evident acest lucru se întâmplă pentru un $i \leq \tau_{\min}$). Deoarece A_{ε} a returnat $\perp$ pentru a_{i-1} şi s pentru a_i avem $s \neq a_{i-1} \bmod n$ şi $s^{\varepsilon} \equiv a_{i-1}^{\varepsilon} \bmod n$ de unde pot fi calculate cele două rădăcini $\left(a_{i-1} \cdot s^{-1}\right)^{\varepsilon} \equiv 1 \bmod n$ şi $\left(a_{i-1}^{-1} \cdot s\right)^{\varepsilon} \equiv 1 \bmod n$ ale lui 1 în Z_n. În conformitate cu teorema 4.3. probabilitatea ca o rădăcină de ordin ε a lui 1 să dea un factor non-banal al lui n este $\frac{\lambda + \mu - 2}{\lambda \cdot \mu}$ şi în consecinţă probabilitatea de factorizare după un apel al lui A_{ε} de către A_{fact} va fi $\mathrm{Pr} = \left(1 - \frac{1}{\lambda \cdot \mu}\right) \cdot \frac{\lambda + \mu - 2}{\lambda \cdot \mu}$. Apeluri repetate la A_{ε}

fac ca această probabilitate să crească. Pe scurt dacă exponentul este mic, probabilitatea de factorizare este mai mare.

Remarca 4.10. În cazul în care nu suntem în posesia unui algoritm A_ε pentru calculul rădăcinii de ordin ε şi tot ce avem este un algoritm A_g care nu returnează $\perp$ sau o rădăcină ci simplu calculează şi returnează $g(a)$ pentru orice intrare a, putem utiliza într-o manieră similară şi pe A_g pentru factorizare. Astfel, fie $a = r^\varepsilon \bmod n$ pentru un întreg aleator $0 \le r < n$ şi $b = g(a)$ valoarea returnată de A_g. Este banal de demonstrat că $b^{\varepsilon^i} \equiv r^{\varepsilon^i} \bmod n$ pentru un întreg $0 \le i \le \tau_{\min}$. Dacă $i = 0$ atunci r era un element din $Z_n^{\varepsilon^{\tau_{\min}}}$ altfel, pentru $i > 0$, obţinem o rădăcină de ordin ε a lui 1 care va duce la factorizarea modulului la fel ca în cazul anterior.

- **Reducţia de la IND-CCA2 asupra ExtRSA-KEM la factorizare**

Securitatea CCA2 a KEM-ului propus se bazează pe faptul că un adversar cu acces adaptiv la oracolul de decriptare O nu poate distinge dacă o anume cheie este sau nu încapsulată în valoarea de challenge $\psi = KEM.Enc(\gamma)$. Un atac asupra acestui obiectiv are patru stagii: în primul o pereche de cheie publică-privată este generată, în pasul doi cheia publică este dată adversarului şi acesta face apeluri la oracolul O, în pasul trei o valoare de challenge este generată care constă în încapsularea unei chei şi o cheie aleasă la întâmplare între cheia încapsulată şi cheia aleasă aleator, în pasul patru adversarul continuă să facă apeluri la oracol şi răspunde la challenge cu un bit b. Acest atac poate fi sintetizat în următorii paşi:

1. $(pk, sk) \leftarrow KEM.Gen(1^k)$
2. $v_1 \leftarrow \mathrm{A}_T^O(pk)$
3. $(\gamma, dk_1) \leftarrow KEM.Key(pk), dk_0 \leftarrow KD, b \leftarrow \{0,1\}, \psi \leftarrow KEM.Enc(\gamma)$
4. $\tilde{b} \leftarrow \mathrm{A}_T^O(v_1, \psi, dk_b)$

Aici v_1 denotă informaţia deţinută de adversar. Avantajul adversarului asupra KEM-ului poate fi acum calculat ca $\xi = \left|\Pr\left[b = \tilde{b}\right] - \frac{1}{2}\right|$. Următoarea teoremă stabileşte nivelul de securitate al mecanismului de încapsulare a cheii:

Teorema 4.11. Dacă există un adversar care poate sparge KEM-ul propus în ROM cu avantaj ξ făcând apeluri la oracolul de decriptare de cel mult q_D ori atunci există un adversar care poate factoriza întregi cu avantaj $\xi' \ge \left(\xi - \frac{q_D}{n}\right) \cdot \left(1 - \frac{\phi(n)}{\lambda \cdot \mu \cdot n}\right) \cdot \left(\frac{\lambda + \mu - 2}{\lambda \cdot \mu}\right)$.

Demonstraţie. Intuiţia pe care ne bazăm este următoarea: vom demonstra că un adversar care poate sparge KEM-ul propus poate factoriza întregi cu avantajul $Adv \ge \xi - \frac{q_D}{n}$. Din cauză că fiecare astfel de rădăcină conduce la o probabilitate de

factorizare egală cu $\left(1-\frac{1}{\lambda\cdot\mu}\right)\cdot\left(\frac{\lambda+\mu-2}{\lambda\mu}\right)$ rezultatul este direct. Dorim deci să demonstrăm că $Adv \geq \xi - \frac{q_D}{n}$ în ROM. Presupunând că funcţiile hash se comportă ca şi funcţii aleatoare este uşor să simulăm oracolul de decriptare. O listă *KList* este păstrată, la fiecare intrare r valoarea $y = r^{\varepsilon} \bmod n$ este calculată şi o valoare aleatoare k este generată – aceste valori sunt păstrate în lista *KList*. Dacă criptotextul y este trimis către oracolul de decriptare, lista *KList* este parcursă şi se verifică dacă valoarea lui y a fost trimisă anterior. În caz afirmativ se returnează valoarea corespunzătoare a lui k iar în caz contrar se generează o nouă valoare şi este păstrată în listă perechea corespunzătoare. Mai târziu, dacă o valoare r este trimisă către oracol, astfel încât există un $y = r^{\varepsilon} \bmod n$ valoarea lui k generată anterior este returnată. Presupunem că în acest mediu un adversar câştigă tot timpul cu avantaj ξ'. Fie *Awins* evenimentul prin care adversarul ghiceşte cu succes bitul ascuns b şi *Bad* evenimentul în care adversarul trimite la oracolul de decriptare valoarea criptotextului challenge sau valoare lui γ către oracolul de derivare a cheii. Acum avem:

$$\Pr[Awins] = \Pr\left[Awins \cap \overline{Bad}\right] + \Pr[Awins \cap Bad] \qquad (4.50)$$

Evident oracolul primeşte ca intrare valoarea de challenge doar înainte de a fi dată către adversar şi acest lucru se poate întâmpla cu probabilitate cel mult egală cu $\frac{q_D}{n}$. Fie *AskKDF* evenimentul în care γ este trimis către oracolul de derivare a cheii, urmează că:

$$\frac{1}{2} + \xi \leq \Pr\left[Awins \cap \overline{Bad}\right] + \frac{q_D}{n} + \Pr[AskKDF] \qquad (4.51)$$

Dacă evenimentul *Bad* nu are loc atunci tot ceea ce ştie adversarul este independent de challenge şi în acest caz avem:

$$\Pr\left[Awins \cap \overline{Bad}\right] = \frac{1}{2} \qquad (4.52)$$

Aceasta conduce în final la:

$$\Pr[AskKDF] \geq \xi - \frac{q_D}{n} \qquad (4.53)$$

Dar evenimentul *AskKDF* implică faptul că rădăcina de ordin ε a fost dezvăluită şi deci adversarul poate fi utilizat pentru a calcula rădăcini de ordin ε cu avantaj $Adv \geq \xi - \frac{q_D}{n}$. Deoarece fiecare rădăcină conduce la un factor cu

probabilitate $\left(1-\frac{1}{\lambda\cdot\mu}\right)\cdot\left(\frac{\lambda+\mu-2}{\lambda\mu}\right)$ avem $\xi' \geq \left(\xi-\frac{q_D}{n}\right)\cdot\left(1-\frac{1}{\lambda\cdot\mu}\right)\cdot\left(\frac{\lambda+\mu-2}{\lambda\cdot\mu}\right)$ şi aceasta demonstrează teorema.

4.5.2 Reducţia pentru Timed-DeMA-QR

Se poate de asemenea demonstra în ROM, că spargerea protocolului Timed-DeMA-QR este echivalentă cu rezolvarea problemei factorizării întregilor. Deoarece această problemă nu este rezolvabilă în momentul de faţă, lucru general cunoscut şi acceptat, se poate spune despre securitatea protocolului că atinge un nivel suficient de înalt.

Informal, este evident că pentru a sparge protocolul, un adversar trebuie să calculeze un pachet $P_i=\left\{i, M_{att}, MAC_{KD(k_{att})}\left(M_{att}\right), k_i\right\}$ unde k_{att} este cheia construită de adversar şi $f\left(k_{att}\right)=k_i$. Vom presupune că acest pachet ajunge la timp şi se dovedeşte autentic atunci când pachetul P_{i+1} ce conţine cheia k_{att} este primit. Dacă $k_i=f\left(k_{att}\right)$ atunci k_{att} este rădăcina pătrată a lui k_i şi din moment ce modulul este produsul a două numere prime vor exista exact patru rădăcini pătrate a lui k_i. Astfel, când serverul eliberează cheia k_{i+1}, avem $k_{i+1}\neq\pm k_{att} \bmod n$ cu probabilitate ½ şi de aceea cu probabilitate ½ adversarul poate factoriza modulul (este cunoscut faptul că problema calculului rădăcinilor pătrate este echivalentă cu problema factorizării). Aceasta înseamnă că dacă un adversar poate frauda protocolul de autentificare prin calculul unei chei k_{att} atunci poate rezolva şi problema factorizării întregilor, iar din moment ce rezolvarea acestei probleme este ne-fezabilă, securitatea protocolului propus este demonstrată.

Pentru completitudinea rezultatului vom face acum o scurtă descriere a unei demonstraţii formale de securitate pentru protocol. Vom presupune că nu există diferenţe de ceas între emiţător şi receptor, eroarea de sincronizare fiind în limita tolerabilă (deci cheile sunt transmise în intervalul corect). De asemenea vom presupune că valorile de iniţializare schimbate cu serverul de sincronizare sunt de asemenea corecte.

În demonstraţia care urmează vom folosi notaţia $\left\{M, MAC_{KD(k_{-1})}\left(M\right), k\right\}$ unde $k=f\left(k_{-1}\right)$. Evident orice pachet trimis în protocolul de broadcast descris corespunde acestei notaţii, de fapt singura diferenţă faţă de descrierea protocolului este renunţarea la indicele i. De asemenea, demonstraţia de securitate va fi făcută pentru un atac de tip criptotext adaptiv al unui adversar împotriva proprietăţii IND a unui MAC având o cheie aleatoare încapsulată cu funcţia f şi o valoare aleatoare care substituie MAC-ul, i.e. IND-CCA2. Prima lucrare care a utilizat proprietatea IND pentru a demonstra securitatea unui astfel de protocol a fost lucrarea în care s-a demonstrat securitatea protocolului TESLA. Aici însă vom folosi o demonstraţie diferită, care este apropiată de demonstraţiile de securitate pentru cadre KEM, precum cea din secţiunea anterioară. O astfel de demonstraţie oferă de fapt un argument solid de securitate într-un mediu mult mai puţin constrâns. Deoarece atât codul MAC cât şi funcţia de derivare a cheii se comportă ca funcţii pseudo-aleatoare, vom presupune că $MAC_{KD(k_{-1})}\left(M\right)$ se comportă tot ca funcţie pseudo-aleatoare şi

vom utiliza un oracol aleator pentru a simula această funcţie. Acum vom presupune existenţa unui adversar IND-CCA2 împotriva construcţiei $\left\{M, MAC_{KD(k_{-1})}(M), k\right\}$ şi următoarea teoremă stabileşte echivalenţa între fraudarea acestei construcţii şi problema factorizării întregilor:

Teorema 4.12. Dacă există un adversar A care poate distinge între $\left\{M, MAC_{KD(k_{-1})}(M), k\right\}$ şi $\{M, r, k\}$ cu avantajul ε făcând q_D apeluri la un oracol O pentru coduri MAC care primind M, k returnează $MAC_{KD(k_{-1})}(M)$, atunci adversarul poate fi utilizat pentru a factoriza întregi cu avantaj $\varepsilon' > \frac{1}{2}\left(\varepsilon - \frac{q_D}{n}\right)$.

Demonstraţie. Oracolul O pentru coduri MAC este simulat după cum urmează. O listă *MACList* iniţial vidă este construită. Dacă adversarul trimite la O valorile M, k_{-1} pentru a obţine $MAC_{KD(k_{-1})}(M)$ atunci $k = k_{-1}^{\,2} \bmod n$ este calculat şi o nouă valoare r este generată iar valorile M, k, r, k_{-1} sunt păstrate în *MACList*. Dacă adversarul trimite la O valorile M, k pentru a obţine $MAC_{KD(k_{-1})}(M)$ atunci în *MACList* sunt căutate valorile corespunzătoare lui M, k şi dacă sunt găsite vor fi returnate, altfel o nouă valoare r este generată şi tripletul M, k, r este păstrat în *MACList*. Dacă lui O i se trimite vreodată M, k_{-1} astfel încât $k = k_{-1}^{\,2} \bmod n$ şi M, k sunt în *MACList* atunci valoarea corespunzătoare r din *MACList* este returnată. Subliniem că oracolul O lucrează într-o manieră similară cu oracolul de decriptare pentru RSA-KEM.

Presupunând că adversarul returnează $\tilde{b}$ avantajul acestuia poate fi definit ca $\varepsilon = \left|\Pr\left[b = \tilde{b}\right] - \frac{1}{2}\right|$. Fie *Awins* evenimentul în care A ghiceşte corect bitul ascuns, fie *AskChall* evenimentul în care A trimite oracolului valorile M, k în faza de căutare şi *AskMAC* evenimentul când A trimite M, k_{-1} cu $k = k_{-1}^{\,2} \bmod n$. Fie $\overline{Bad}$ evenimentul asociat absenţei lui *AskChall* şi *AskMAC*; deoarece în această situaţie tot ceea ce ştie adversarul este independent de valoarea de challenge avem $\Pr\left[Awins \cap \overline{Bad}\right] = \frac{1}{2}$. Deoarece $\Pr\left[Awins\right] = \frac{1}{2} + \varepsilon$, şi $\Pr\left[Awins\right] = \Pr\left[Awins \cap \overline{Bad}\right] + \Pr\left[Awins \cap Bad\right]$, urmează că $\frac{1}{2} + \varepsilon \le \frac{1}{2} + \frac{q_D}{n} + \Pr\left[AskMAC\right]$. Cum evenimentul *AskMAC* implică faptul că rădăcina pătrată a lui k este făcută publică şi cunoaşterea rădăcinilor pătrate este echivalentă cu factorizarea avem $\varepsilon' \ge \frac{1}{2} \cdot \Pr\left[AskMAC\right]$. Acum, prin înlocuire în relaţia anterioară, obţinem următorul avantaj la factorizare $\varepsilon' \ge \frac{1}{2}\left(\varepsilon - \frac{q_D}{n}\right)$. Deci putem spune că $IFP <_P IND - CCA2(Timed - DeMA - QR)$ şi cum

$IND-CCA2(Timed-DeMA-QR) <_P IFP$ înseamnă că problemele sunt echivalente, deci $IND-CCA2(Timed-DeMA-QR) \Leftrightarrow IFP$.

4.6 Puzzleuri criptografice înlănţuite pentru prevenirea atacurilor DoS

4.6.1 Scurtă descriere a procedeelor de construcţie pentru puzzleuri criptografice

Puzzleurile criptografice sunt construcţii bazate pe funcţii criptografice simetrice, asimetrice sau fără cheie, a căror rezolvare constă în recuperarea unei valori care a jucat rolul de cheie la construcţia lor. Acestea au aplicaţii în diverse zone ale securităţii informaţiei cea mai relevantă aplicaţie fiind protecţia în faţa atacurilor de tip Denial of Services (DoS). Diverse lucrări au propus utilizarea acestora pentru a preveni atacuri DoS asupra sistemelor sau chiar asupra protocoalelor de autentificare [3], [75], [78]. Ideea care stă la baza utilizării puzzleurilor criptografice pentru a combate aceste atacuri într-un scenariu de tip client-server este următoarea: dacă nu există suspiciuni că un atac ar avea loc de partea serverului atunci resursele sunt alocate către clienţi în mod normal, altfel dacă apar suspiciuni că un anume atac ar putea avea loc (dacă numărul de resurse solicitate creşte drastic) atunci severul trimite puzzleuri către fiecare client care solicită resurse. În acest fel un potenţial adversar care a solicitat mai multe resurse are de rezolvat mai multe puzzleuri consumând în acest fel din resursele proprii. Există trei procedee constructive distincte pentru puzzleuri ce vor fi descrise în cele ce urmează în scopul formării unui referenţial.

- **Puzzleuri bazate pe inversarea funcţiilor one-way**

Puzzleurile bazate pe inversare funcţiilor one-way, în particular a funcţiilor hash, oferă cea mai directă soluţie constructivă. Folosirea acestor funcţii este o bună idee în primul rând pentru că necesită putere de calcul nesemnificativă la construcţia puzzleului. Ideea care stă la baza acestor construcţii este simplă: pentru a rezolva puzzleul trebuie găsită valoarea intrării unei funcţii hash cunoscându-se ieşirea şi o parte a intrării. Mai exact construcţia poate fi făcută în baza următoarei relaţii: $P = f(\sigma), \sigma = \rho \,||\, rand_l$. Având valoarea lui P şi ρ potenţialul rezolvator trebuie să găsească valoarea aleatoare de l biţi notată cu $rand_l$ care a fost concatenată cu ρ pentru a obţine rezultatul aferent. Aceasta echivalează cu căutarea unui spaţiu de dimensiune 2^l şi poate fi efectuată într-un timp mediu de 2^{l-1} funcţii hash. Valoarea l se mai numeşte, din motive evidente, nivel de dificultate al puzzleului.

- **Puzzleuri bazate pe inversarea logaritmilor discreţi**

Ideea care stă la baza acestei construcţii este inversarea logaritmului discret problemă cunoscută ca fiind intractabilă. Logaritmii discreţi au proprietăţi care fac ca aceste puzzleuri să fie utile în diverse scenarii [8]. Un exemplu relevant de astfel de puzzle este în lucrarea [121] unde noţiunea de bastion este introdusă. Denumirea

de bastion o poartă o entitate de încredere care protejează un număr de servere prin trimiterea de puzzleuri către clienţii care solicită resurse de la aceste servere. Pentru construcţie se foloseşte un mecanism bazat pe schema Diffie-Helmann care pe scurt poare fi descris după cum urmează. Fără a pierde din generalitatea schemei, vom presupune că se lucrează în Z_p cu p prim şi vom menţiona că mecanismul funcţionează în orice alt grup unde problema logaritmului discret este intractabilă. Serverul publică valoarea $a^x \bmod p$ iar bastionul publică $a^y \bmod p$ unde a este generator în Z_p şi x, y numere aleatoare. Se observă că deoarece avem $\left(a^x\right)^y \equiv \left(a^y\right)^x \equiv a^{x \cdot y} \bmod p$ atât serverul cât şi bastionul cunosc valoarea lui $a^{xy} \bmod p$. Clientul primeşte de la bastion valoarea lui $a^y \bmod p$ şi doar o informaţie parţială despre y (de exemplu ultimii l biţi ai lui y). A rezolva puzzleul înseamnă a găsi valoarea $a^{xy} \bmod p$. Desigur, pentru a găsi această valoare, clientul trebuie să găsească întâi valoarea lui y şi aceasta va presupune inversarea logaritmului discret folosind informaţia parţială despre y. Marele avantaj al unei astfel de construcţii este că bastionul poate proteja mai multe servere fără a le implica în construcţia efectivă a puzzleului şi deci fără a pierde timpul serverelor în procesul de construcţie.

- **Puzzleuri cu blocare temporală Time-Lock (bazate pe reducerea exponenţilor în grupuri de întregi)**

Utilizarea funcţiei ridicare la pătrat discretă, i.e. $f\left(x\right) = x^2 \bmod n$, folosită pentru autentificare în secţiunile anterioare, a fost propusă şi pentru puzzleuri criptografice în [103]. Această funcţie poate fi folosită pentru a construi puzzleuri cu blocare temporală care pot fi asemănate cu nişte capsule folosite pentru a trimite informaţie în viitor (practic informaţia criptată poate fi decriptată doar după un anumit timp). Construcţia se bazează pe faptul deja amintit că exponenţii pot fi reduşi modulo $\phi\left(n\right)$ atunci când lucrăm în Z_n. Dacă lucrăm cu funcţia $f\left(x\right) = x^2 \bmod n$ atunci compoziţia succesivă a funcţiei de η ori poate fi calculată, aşa cum s-a spus deja, ca $f^\eta\left(x\right) = x^{2^\eta \bmod \phi(n)} \bmod n$. Operaţie care poate fi eficient făcută de entitatea care cunoaşte factorizarea lui n; altfel această operaţie fiind un procedeu secvenţial care implică η compoziţii succesive. Pentru valori mari ale lui η următoarea relaţie defineşte un puzzle de tip time-lock: $P_K = K + a^{2^\eta} \bmod n$. Rezolvarea acestui puzzle presupune găsirea valorii lui K. Având valoarea lui a şi η o entitate care nu este în posesia factorizării lui n poate calcula $a^{2^\eta} \bmod n$ doar cu η ridicări succesive la pătrat. Ulterior cheia K recuperată din P_K poate fi folosită pentru a decripta o informaţie criptată cu aceasta.

Procedeul constructiv pentru puzzleuri introdus în prezenta teză, poate fi văzut şi ca o combinaţie între puzzleurile neparalelizabile cu blocare temporală şi cele bazate pe simpla inversare a funcţiilor one-way. Scopul acestei construcţii este atingerea unei proprietăţi care nu apare în puzzleurile anterior amintite şi anume

faptul că un client poate rezolva doar o parte din puzzle şi să trimită serverului o soluţie parţială urmând ca resursele să îi fie alocate în consecinţă.

4.6.2 Construcţia puzzleurilor înlănţuite

Ideea de a utiliza puzzleuri criptografice pentru a consuma resurse de partea celor care le solicită este corectă, doar că de partea unui client onest care solicită resurse lucrurile pot fi diferite. De exemplu putem presupune cazul unui client care a câştigat acces la mai multe servere care sunt sub atac. Acest client se poate trezi brusc cu un număr ridicat de puzzleuri primite din partea serverelor lucru care i-ar consuma o cantitate mare de resurse proprii. În acest context ar fi mai bine ca un client să poată continua utilizarea acestor resurse cu anumite priorităţi, fiind plauzibil ca anumite resurse să fie mai relevante pentru client decât altele. Puzzleurile înlănţuite care vor fi introduse în această secţiune oferă o soluţie bună pentru această problemă, deoarece puzzleurile înlănţuite oferă rezolvări mult mai flexibile în care un client poate alege să rezolve doar o cantitate anume de puzzleuri dintr-un lanţ şi să obţină resurse proporţional cu cantitatea rezolvată. Un astfel de scenariu este sugerat în figura 4.17.

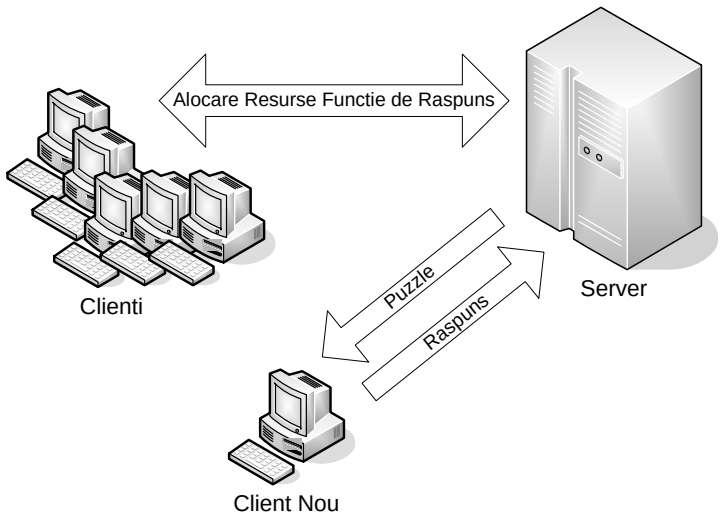


Figura 4.17. Scenariu de utilizare a unui puzzle înlănţuit

Un puzzle înlănţuit constă într-un număr de puzzleuri care pot fi rezolvate doar într-o ordine prestabilită. Vom defini mulţimea puzzleurilor ca $\Pi = \{P_0, P_1, \ldots, P_{n-1}\}$ şi relaţia de ordine între acestea ca $\Omega : \Pi \times \Pi \to \{0,1\}$. Pentru orice $i \neq j$ avem $\Omega(P_i, P_j) = 1$ dacă rezolvarea P_i necesită soluţia lui P_j şi $\Omega(P_i, P_j) = 0$ în caz contrar. În consecinţă un puzzle înlănţuit este format dintr-o mulţime de puzzleuri Π şi o relaţie de ordine Ω. Evident această mulţime şi relaţia de ordine definesc un graf orientat aciclic şi în fiecare puzzle înlănţuit trebuie să

existe cel puţin un puzzle a cărui rezolvare să nu depindă de rezolvarea altor puzzleuri.

În cele ce urmează ne vor interesa doar puzzleuri construite pe baza funcţiilor hash deoarece acestea sunt cele mai puţin intense funcţii din punct de vedere computaţional dar conceptele introduse sunt generale şi pot fi extinse pentru puzzleuri construite cu orice alt tip de funcţii. Definim în acest sens două tipuri de puzzleuri înlănţuite:

1) Puzzleuri înlănţuite liniar. Puzzleurile înlănţuite liniar sunt acele puzzleuri unde oricare puzzle P_i depinde de exact un singur alt puzzle P_j. Desigur, pentru ca lanţul de puzzleuri să poată fi rezolvabil, există un singur puzzle a cărui rezolvare nu depinde de rezolvarea nici unui alt puzzle şi de la a cărui rezolvare porneşte rezolvarea întregului lanţ de puzzleuri.
2) Puzzleuri înlănţuite aleator. Puzzleurile înlănţuite aleatoriu sunt o generalizarea a puzzleurilor înlănţuite liniar în care puzzleurile sunt înlănţuite aleatoriu, un puzzle putând să depindă de unul sau mai multe puzzleuri. Remarcăm din nou, că pentru ca lanţul de puzzleuri să poată fi rezolvabil, există un singur puzzle a cărui rezolvare nu depinde de rezolvarea nici unui alt puzzle.

Pentru a construi puzzleuri liniar înlănţuite vom folosi o funcţie hash f iar primul puzzle îl vom defini ca:

$$P_0 = f^2(\sigma_0), \sigma_0 = \rho_0 \,||\, rand_l \tag{4.54}$$

Simbolul || denotă concatenare iar prin $rand_l$ notăm o secvenţă de l biţi aleatori, din aceste motiv prin $rand_l$ ne referim tot timpul la o secvenţă nouă de l biţi aleatori. Următoarea relaţie va fi utilizată pentru construcţia celorlalte puzzleuri din lanţ:

$$P_i = f^2(\sigma_i), \sigma_i = (\rho_i \,||\, rand_l) \oplus f(\sigma_{i-1}), 0 < i < n \tag{4.55}$$

Este necesar a calcula valoarea lui $f(\sigma_{i-1})$ şi a efectua un XOR cu $\rho_i \,||\, rand_l$, deoarece dacă efectuăm XOR direct între σ_{i-1} şi $\rho_i \,||\, rand_l$, va urma că $\sigma_i = (\rho_i \,||\, rand_l) \oplus (\rho_{i-1} \,||\, rand_l) = (\rho_i \oplus \rho_{i-1}) \,||\, rand_l$ ceea ce va face soluţia lui P_i independentă de soluţia lui P_{i-1}. Rezolvarea puzzleului P_i, $0 \le i < n$ înseamnă găsirea unei valori σ_i având ρ_i astfel încât $P_i = f^2(\sigma_i)$ lucru care echivalează cu căutarea prin forţă brută, i.e. căutarea exhaustivă, a unui spaţiu de dimensiune 2^l ce va lua în medie 2^{l-1} verificări de cheie; deoarece fiecare cheie necesită evaluarea a două funcţii hash, efortul de calcul va fi de 2^l evaluări de funcţie hash. Deoarece inversarea unei funcţii hash nu este fezabilă, calculul lui σ_i va necesita desigur valoarea lui σ_{i-1}. Figura 4.18 ilustrează o astfel de construcţie.

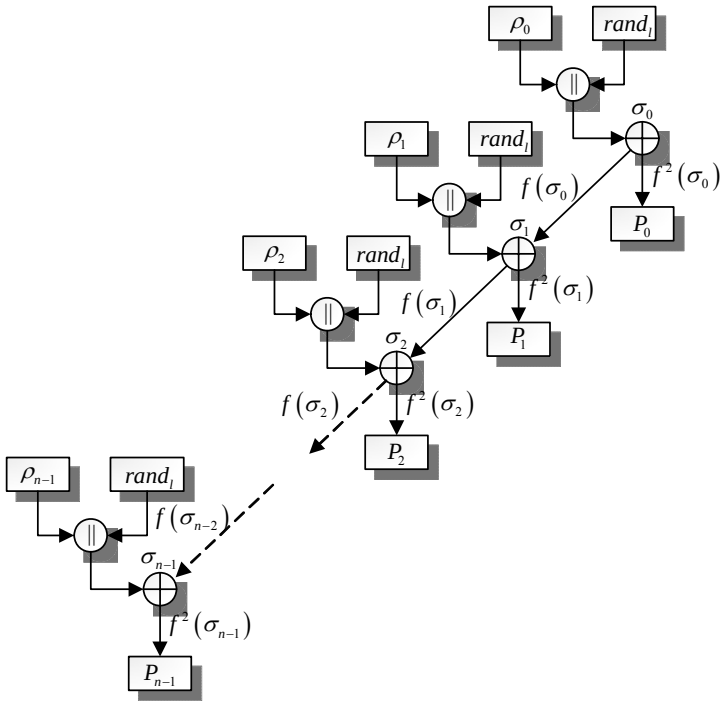


Figura 4.18. Exemplu de puzzle liniar înlănţuit

Desigur pentru a rezolva puzzleul înlănţuit clientul va trebui să rezolve lanţul în ordinea în care a fost construit. În acest caz, al puzzleurilor liniar înlănţuite relaţia de ordine Ω este definită ca:

$$\Omega(P_i, P_j) = \begin{cases} 1, & i = j+1 \\ 0, & i \neq j+1 \end{cases} \qquad i, j \in \{0,1,\ldots,n-1\} \tag{4.56}$$

Construcţia puzzleurilor înlănţuite aleator se face într-o manieră similară cu a celor înlănţuite liniar. Relaţia (4.54) va fi din nou folosită pentru definirea lui P_0. În ceea ce priveşte puzzleul P_i, acesta poate fi făcut să depindă aleator de oricare P_j pentru $j = 0,\ldots,i-1$ prin calcularea lui P_i cu următoarea relaţie:

$$P_i = f^2(\sigma_i),$$

$$\sigma_i = (\rho_i \,||\, rand_l) \oplus \left(\Omega(P_i, P_{i-1}) \cdot f(\sigma_{i-1}) \oplus \ldots \oplus \Omega(P_i, P_0) \cdot f(\sigma_0)\right),$$

$$0 < i < n \tag{4.57}$$

Aici prin Ω desemnăm relaţia de ordine definită după cum urmează:

$$\Omega\left(P_i, P_j\right) = \begin{cases} 0, & i \geq j \\ r, & i < j \end{cases} \qquad i, j \in \{0,1,\ldots,n-1\} \tag{4.58}$$

Prin r notăm o valoare aleatoare din mulţimea $\{0,1\}$. Aceasta va face ca σ_i să depindă aleatoriu de oricare dintre cheile σ_j pentru $j = 0,\ldots,i-1$. Dacă relaţia de ordine Ω este cunoscută atunci rezolvarea puzzleului înlănţuit aleator este de aceeaşi complexitate cu rezolvarea puzzleului înlănţuit liniar. Altfel, dacă ordinea în care puzzleurile au fost înlănţuite este necunoscută clientul poate rezolva puzzleul dar numărul de combinaţii posibile creşte exponenţial.

Trebuie remarcat că şi în cazul puzzleurilor liniar înlănţuite, dacă ordinea în care puzzleurile au fost construite nu este cunoscută de client, deziderat care poate fi uşor atins prin transmisia puzzleurilor într-o ordine aleatoare, atunci clientul poate în continuare rezolva puzzleul dar sunt $n \cdot (n-1)/2$ combinaţii posibile. Nu vedem însă motive serioase pentru a ascunde ordinea în care au fost construite puzzleurile deoarece creşterea complexităţii unui puzzle poate fi uşor făcută prin creşterea nivelului de dificultate sau a numărului de puzzleuri. Concluzionăm deci că singura construcţie utilă şi cu o proprietate relevantă în practică (rezolvarea parţială) sunt puzzleurile liniar înlănţuite.

5 Aplicaţii în sisteme de control

5.1 Sinteza unui protocol cu aplicaţie în sisteme de control

În trecut, sistemele de control industriale erau izolate de reţelele publice şi securitatea lor se baza pe obscuritatea protocoalelor respectiv pe perimetre închise inaccesibile publicului larg. În prezent lucrurile s-au schimbat şi sistemele informatice industriale trebuie să comunice prin reţele publice, precum Internet-ul, devenind astfel expuse în faţa unor potenţiali adversari. Astfel, introducerea unor măsuri de securitate în acest sens a devenit strict necesară. În consecinţă utilizarea criptografiei în acestă zonă a devenit de interes. Totuşi implementarea criptografiei întâmpină o dificultate majoră: criptografia necesită resurse computaţionale care uneori nu sunt disponibile. În acest context considerăm că utilizarea protocoalelor de autentificare bazate pe lanţuri one-way este de mare interes deoarece acestea au necesităţi computaţionale scăzute şi oferă mari avantaje de securitate. În mare, implementarea unui astfel de protocol nu trebuie să ridice probleme, deoarece astfel de protocoale pot fi construite pe funcţii simple one-way care au necesităţi computaţionale scăzute. Este de remarcat că astfel de protocoale au fost utilizate chiar şi în medii constrânse precum reţelele de senzori.

Prin protocolul sintetizat în acest capitol se obţin îmbunătăţiri substanţiale ale protocolului bazat pe lanţuri one-way DeMA, anterior propus, în scopul atingerii următoarelor obiective, necesare unui scenariu de control la distanţă în prezenţa unor potenţiali adversari. Dintre caracteristicile protocolului DeMA şi îmbunătăţirile aduse prin acest capitol enumerăm:

1) Protocolul se bazează doar pe funcţii one-way care sunt simplu de calculat. În acest sens se pot utiliza funcţii hash din considerente de performanţă computaţională sau funcţia ridicare la pătrat discretă pentru lanţuri de dimensiune nelimitată.

2) Oferă autenticitate în transmisia informaţiei, ceea ce înseamnă că informaţia nu a fost alterată şi provine de la o sursă a cărei identitate poate fi garantată.

3) Nu depinde de secrete partajate, fiecare entitate nu stochează decât secretele proprii, deci protocolul propus nu se bazează pe chei secrete partajate. Subliniem însă că protocolul nu este nici perfect asimetric deoarece pierderea unei chei private de partea unei entităţi duce la pierderea securităţii pentru întregul protocol.

4) Nu necesită sincronizare temporală între entităţi şi în consecinţă ratele de transfer sunt flexibile.

5) Oferă o garanţie pentru secvenţierea pachetelor (time-line) ceea ce înseamnă că ordinea în care ajung pachetele nu poate fi schimbată de un potenţial adversar.

6) Întârzierile de autentificare sunt semnificativ scurtate. Prin natura protocolului întârzierile de autentificare sunt inevitabile. Prin utilizarea directă a propunerilor din capitolul anterior întârzierea de autentificare ar fi de 4 runde (problema va fi explicată în secţiunea 5.1.3), în timp ce prin propunerea din această secţiune întârzierea devine de doar 2 runde.

7) Reducerea necesităţilor computaţionale prin renunţarea la operaţiunile criptografice neesenţiale pentru asigurarea securităţii. Se propune şi o variantă simplificată de protocol care are costuri computaţionale minime.

8) Se oferă o variantă de protocol pentru obţinerea de informaţii suplimentare despre autenticitatea comenzii şi a răspunsului.

5.1.1 Scenariul urmărit

Importanţa securităţii în sistemele informatice industriale este unanim recunoscută, o monografie asupra acestui subiect se găseşte în [30], dar există multe alte lucrări care referă probleme similare [20], [30], [33], [41], [73], [77], [91], [104], [113], [114], [116], [118], [125]. În principiu sistemele de control la distanţă au multe caracteristici care le fac diferite de sistemele tradiţionale de procesare a informaţiei şi mai mult implementările diferă de la caz la caz fără să existe o abordare unitară universal acceptată. În ultimii ani se desfăşoară un efort constant de a standardiza caracteristicile sistemelor de control industriale, un exemplu bun este efortul pentru standardizarea Ethernetului industrial [34]. Pe scurt, un sistem de conducere este alcătuit dintr-un dispozitiv de conducere şi un proces condus, rolul dispozitivului de conducere fiind de a regla comportamentul procesului condus. Reglarea se face prin intermediul unei legi de reglare, care este o funcţie ce primeşte ca intrare ieşiri din procesul condus şi referinţa şi are ca ieşire comanda. În figura 5.1 este prezentată o astfel de structură.

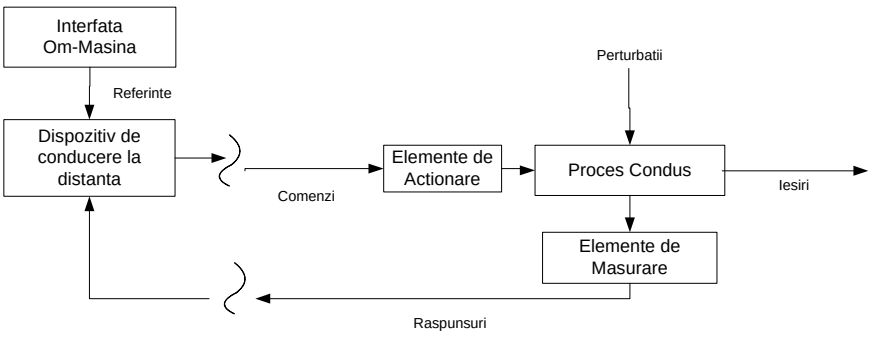


Figura 5.1. Sistem de conducere

În mod uzual, deoarece este impropriu ca procesul condus să fie lăsat să evolueze singur în cazul în care comunicaţia este pierdută cu dispozitivul de conducere la distanţă, un dispozitiv de conducere local este prezent pentru a prelua funcţia de conducere. În figura 5.2 se prezintă un sistem de conducere tolerant la erori cu o buclă internă necesară asigurării unui regim de siguranţă.

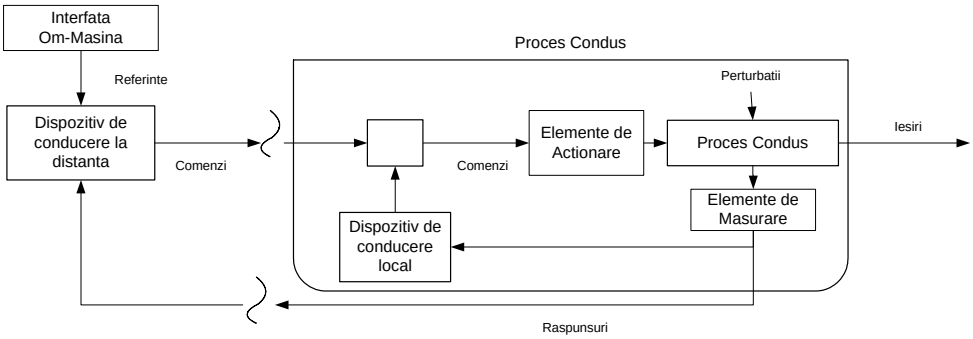


Figura 5.2. Sistem de conducere tolerant la pierderea comunicării

Un astfel de sistem nu va schimba însă scenariul urmărit deoarece din punct de vedere al securităţii noul ansamblu format din bucla internă este tot un proces condus urmând ca securitatea să fie asigurată între dispozitivul de conducere şi noul proces condus. În consecinţă prin protocolul introdus suntem interesaţi în asigurarea securităţii între cei doi participanţi: sistemul de conducere şi procesul condus. Informaţia schimbată va fi denumită comandă când circulă de la dispozitiv la proces şi răspuns când provine de la proces şi este destinată dispozitivului.

5.1.2 Probleme urmărite şi rezolvarea lor

O structură de sistem de conducere se numeşte robustă dacă poate opera în toate condiţiile de operare posibile. În mare, problema controlului robust este de a găsi o lege de conducere care să păstreze indicatorii de calitate ai răspunsului sistemului în intervalele tolerate în ciuda eventualelor incertitudini. Aceste incertitudini pot include: perturbaţii, zgomotul de măsurare, erori de modelare datorate parametrilor variabili în timp sau neliniarităţilor. Evident, când dispozitivele de conducere operează prin reţele publice, condiţiile de operare vor include şi prezenţa unor potenţiali adversari, care pot întreprinde acţiuni ca: intercepţia şi modificarea unor mesaje. Tratarea problemelor legate de intruşi este în primul rând problema criptologiei şi doar în al doilea rând a controlului automat. Din aceste motive nu vom altera noţiunea de control robust şi o vom extinde cu cea de control robust securizat. Numim un sistem de conducere robust ca fiind securizat dacă el îşi păstrează robusteţea chiar şi în prezenta adversarilor. Desigur, problemele de securitate tratate, apar în transmisia informaţiei între dispozitivul de conducere şi procesul condus (atât pe linia directă cât şi pe cea de reacţie). Nu suntem preocupaţi de probleme cum ar fi întârzierile sau pierderea comunicării, deoarece nu există contramăsuri criptografice împotriva acestor probleme, în cele din urmă acestea fiind problema controlului robust şi nu a criptografiei. În aceste condiţii

suntem preocupaţi de asigurarea securităţii controlului în prezenta potenţialilor adversari care pot altera informaţia vehiculată între dispozitivul de conducere şi procesul condus. Remarcăm faptul că un scenariu de conducere rămâne sigur în două cazuri:

1) Cazul în care un adversar nu poate altera autenticitatea comenzilor şi reacţiilor.

2) Cazul în care procesul condus poate decide dacă comanda este autentică şi a fost într-adevăr calculată în baza stării curente a procesului.

Validitatea ambelor cazuri poate fi uşor explicată. Pentru ambele cazuri autenticitatea comenzilor este în mod evident necesară. Totuşi aceasta nu este suficientă deoarece un potenţial adversar poate altera autenticitatea răspunsurilor din proces cauzând elaborarea de către dispozitivul de conducere a unor comenzi incorecte – din aceste motive în primul caz se impune şi autentificarea mărimii de reacţie a procesului condus denumită în continuare răspuns. Pentru cel de-al doilea caz, chiar dacă autenticitatea răspunsului nu poate fi verificată de către dispozitivul de conducere, atâta timp cât procesul condus poate verifica autenticitatea comenzii şi faptul că ea a fost calculată în baza răspunsului curent, scenariul de control rămâne robust – deoarece o comandă frauduloasă nu va fi acceptată. Subliniem că ambele situaţii a) şi b) oferă condiţiile necesare şi suficiente de securitate, deoarece dacă un adversar alterează autenticitatea informaţiei tot ce va reuşi să facă este să introducă întârzieri în sistem pentru că pachetele neautentificate nu vor putea fi folosite.

Soluţia propusă constă în asigurarea autenticităţii informaţiei schimbate între dispozitivul de conducere şi procesul condus prin utilizarea unui protocol de autentificare bazat pe lanţuri one-way. Cu toate că în literatura de specialitate există suficiente propuneri de astfel de protocoale, nici una din aceste propuneri nu este suficient de bună pentru scenariul de conducere urmărit. Acest lucru se datorează în special faptului că marea parte a propunerilor de autentificare bazate pe lanţuri one-way adresează scenarii de broadcast, în care transmisia este unidirecţională fără a presupune un răspuns asociat fiecărui pachet de informaţie transmis. Noţiunea de control este indisolubil legată de conceptul de feed-back şi datorită acestui fapt o comunicare unidirecţională este în mod evident ineficientă. De asemenea, în general un scenariu de conducere presupune o comunicare de tip unul-la-unul cu un singur dispozitiv de conducere şi un singur proces, nemaifiind necesară transmisia către mai multe procese conduse lucru care simplifică tratarea problemei şi face posibile câteva modificări pentru a reduce resursele computaţionale necesare şi cele de comunicare. Această ipoteză nu exclude cazul în care un regulator conduce mai multe procese, având pentru fiecare o ieşire de comandă calculată după un algoritm diferit, dar exclude varianta utilizării protocoalelor pentru broadcast.

În acest context, în paragraful următor vor fi introduse câteva variante ale protocolului DeMA bazat pe lanţuri one-way şi fără sincronizare temporală. În principiu, utilizarea unui astfel de protocol de autentificare are următoarele merite:

1) Securitatea acestui tip de protocoale este bine stabilită prin demonstraţii formale şi nu se cunosc deficienţe de securitate (metode prin care ar putea fi sparte).

2) Aceste protocoale pot fi construite pe baza celor mai simple funcţii one-way, cum sunt de exemplu funcţiile hash.

3) Protocoalele bazate pe lanţuri one-way oferă cea mai bună alternativă pentru operaţiile costisitoare ale primitivelor cu cheie publică fără să necesite secrete partajate.

5.1.3 O abordare directă de autentificare folosind protocolul DeMA

Ca şi abordare directă, protocolul DeMA poate fi utilizat aşa cum a fost descris în capitolul anterior. În cele ce urmează, dispozitivul de conducere va fi notat cu A iar procesul condus cu B, comanda din sesiunea k cu $c_{A,k}$ iar răspunsul cu $r_{B,k}$. Amintim că datorită întârzierii de autentificare, autenticitatea acestor valori din sesiunea k poate fi verificată doar în sesiunea următoare $k+1$. Pentru claritate, vom reda paşii protocolului pentru patru sesiuni consecutive între dispozitivul de conducere şi procesul condus:

Sesiunea k

$$A \to B:\ c_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}\right), \sigma_A(k)$$

$$B \to A: r_{B,k}, MAC_{KD(\sigma_B(k+1))}\left(r_{B,k}\right), \sigma_B(k)$$

Sesiunea $k+1$

$$A \to B:\ c_{A,k+1}, MAC_{KD(\sigma_A(k+2))}\left(c_{A,k+1}\right), \sigma_A(k+1)$$

$$B \to A: r_{B,k+1}, MAC_{KD(\sigma_B(k+2))}\left(r_{B,k+1}\right), \sigma_B(k+1)$$

Sesiunea $k+2$

$$A \to B:\ c_{A,k+2}, MAC_{KD(\sigma_A(k+3))}\left(c_{A,k+2}\right), \sigma_A(k+2)$$

$$B \to A: r_{B,k+2}, MAC_{KD(\sigma_B(k+3))}\left(r_{B,k+2}\right), \sigma_B(k+2)$$

Sesiunea $k+3$

$$A \to B:\ c_{A,k+3}, MAC_{KD(\sigma_A(k+4))}\left(c_{A,k+3}\right), \sigma_A(k+3)$$

$$B \to A: r_{B,k+3}, MAC_{KD(\sigma_B(k+4))}\left(r_{B,k+3}\right), \sigma_B(k+3)$$

Folosind această abordare este evident că emiterea comenzii care corespunde răspunsului $r_{B,k}$ de către dispozitivul de conducere se poate face doar după ce autenticitatea răspunsului poate fi verificată. Dar, pentru aceasta trebuie aşteptat până în runda 2 a sesiunii $k+1$, pentru a testa autenticitatea valorii $r_{B,k}$, apoi comanda poate fi trimisă în runda 1 a sesiunii $k+2$ şi autenticitatea acestei

comenzi poate fi testată de procesul condus în runda 1 a sesiunii $k+3$. În mod clar acest lucru va cauza o întârziere de 4 runde. Subliniem că întârzierile de autentificare pot fi de asemenea rezolvate din design-ul dispozitivului de conducere, şi aceasta rămâne ca subiect pentru lucrări ulterioare. În cele din urmă pentru procese lente întârzierea poate să nici nu fie relevantă.

5.1.4 Optimizări ale protocolului DeMA

Datorită naturii scenariului propus câteva modificări devin necesare faţă de abordarea iniţială. Există două probleme distincte care dorim să le rezolvăm prin abordările următoare: reducerea întârzierile de autentificare şi reducerea costurilor de calcul.

Deoarece costurile de calcul implicate de protocol provin din utilizarea codurilor MAC şi a funcţiilor hash pentru generarea lanţului one-way iar aceste funcţii nu sunt deloc intense din punct de vedere computaţional, fiind cele mai simple primitive criptografice, vom fi în primul rând interesaţi de problema reducerii întârzierilor de autentificare. Din fericire, întârzierea de 4 runde care se obţine prin aplicarea directă a protocolului DeMA, aşa cum s-a observat în secţiunea anterioară, poate fi înjumătăţită.

Această reducere a întârzierii de autentificare poate fi obţinută prin legarea răspunsului de comandă. Rezultatul se bazează pe observaţia că la primirea unui răspuns din partea procesului condus în sesiunea k dispozitivul de conducere poate deja calcula comanda care va fi trimisă către procesul condus. Chiar dacă în acest moment răspunsul nu este încă dovedit ca fiind autentic iar comanda poate fi calculată pe un răspuns eronat, procesul condus va detecta orice comandă necorespunzătoare la verificarea autenticităţii comenzii deoarece aceasta este acum legată de răspuns. Astfel orice comandă care nu a fost construită pe răspunsul corect va fi neglijată de către proces.

Pentru a lega comanda de răspuns trebuie doar să calculăm codul MAC pe concatenarea celor două valori, astfel, în loc de a calcula $MAC_{KD(\sigma_A(k+1))}(c_{A,k})$ vom calcula $MAC_{KD(\sigma_A(k+1))}(c_{A,k}, r_{B,k-1})$ de partea dispozitivului de conducere – acest lucru leagă comanda de răspuns din punct de vedere criptografic. Comanda astfel calculată poate fi transmisă direct în sesiunea k şi la primirea comenzii $k+1$ procesul condus poate decide dacă aceasta corespunde răspunsului din sesiunea k. Aceasta transformă sesiunile protocolului propus în următoarele:

Sesiunea $k, 1 \le k \le \eta$ **a protocolului DeMA în Varianta cu Comandă Directă** (Direct Command Variant - DCV)

$$A \to B:\ c_{A,k}, MAC_{KD(\sigma_A(k+1))}(c_{A,k}, r_{B,k-1}), \sigma_A(k)$$

$$B \to A: r_{B,k}, MAC_{KD(\sigma_B(k+1))}(r_{B,k}), \sigma_B(k)$$

În această variantă prin legarea răspunsului de comandă a fost redusă întârzierea de autentificare la doar două runde. Astfel autentificarea comenzii trimisă în sesiunea k şi calculată pentru răspunsul din sesiunea $k-1$ se obţine în runda 1

a sesiunii $k+1$ când soseşte cheia de autentificare a codului $MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}, r_{B,k-1}\right)$.

Varianta DCV poate fi simplificată din punct de vedere computaţional de partea procesului condus prin eliminarea MAC-ului calculat pe răspuns. Cu toate că într-adevăr dispozitivul de conducere nu poate testa autenticitatea răspunsului, aceasta nu va afecta scenariul de control robust deoarece comanda va fi acceptată de procesul condus doar dacă este legată de răspunsul corect. Aceasta conduce la următoarele variante de protocol:

Sesiunea $k, 1 \le k \le \eta$ **a protocolului DeMA în Varianta Simplificată** (Simplified variant - SV)

$$A \to B:\ c_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}, r_{B,k-1}\right), \sigma_A(k)$$

$$B \to A: r_{B,k}, \sigma_B(k)$$

În varianta simplificată dispozitivul de conducere nu poate verifica dacă răspunsul primit este corect deoarece nu există un cod de autentificare calculat asupra acestuia. Aceasta obstrucţionează observabilitatea procesului condus deoarece răspunsul procesului nu poate fi verificat pentru autenticitate şi deci nu există nici o garanţie criptografică asupra stării procesului, dar procesul condus poate verifica dacă valorile comenzilor au fost într-adevăr calculate pe răspunsul corect şi în acest fel robusteţea controlului este păstrată.

Poate fi de asemenea important ca atât dispozitivul de conducere cât şi procesul condus să verifice dacă răspunsul şi comanda primite sunt corecte (de exemplu interfeţele operator – proces condus au de obicei afişaje cu privire la aceste valori). În cazul în care aceste necesităţi trebuie atinse dispunem de următoarea variantă în care poate fi verificată care parte a mesajului este eronată (comanda sau răspunsul). Pentru aceasta se poate calcula un MAC separat pentru ambele valori, descrierea protocolului în acest caz este următoarea:

Sesiunea $k, 1 \le k \le \eta$ **a protocolului DeMA în Varianta Completă** (Complete variant - CV)

$$A \to B:\ c_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(c_{A,k}\right), MAC_{KD(\sigma_A(k+1))}\left(r_{B,k-1}\right), \sigma_A(k)$$

$$B \to A: r_{B,k}, MAC_{KD(\sigma_B(k+1))}\left(c_{A,k}\right), MAC_{KD(\sigma_B(k+1))}\left(r_{B,k}\right), \sigma_B(k)$$

În această variantă de protocol atât dispozitivul de conducere cât şi procesul condus pot verifica individual autenticitatea valorilor primite la o întârziere de doar două runde. Acelaşi lucru este obţinut prin aplicarea directă a protocolului, detaliată în secţiunea anterioară, dar la un delay de 4 runde cu avantajul unor costuri computaţionale ceva mai scăzute datorită absenţei calculului unui cod MAC separat pe comandă respectiv răspuns.

5.1.5 Câteva aspecte de securitate

Deoarece câteva propuneri de autentificare bazate pe lanţuri one-way au întâmpinat diverse probleme de securitate, care urmează să fie discutate în acest paragraf, sunt necesare câteva explicaţii cu privire la faptul că aceste atacuri nu reprezintă o vulnerabilitate pentru protocolul propus.

Prima vulnerabilitate este cea a sistemului de autentificare cu lanţuri one-way S-Key care poate fi spart printr-un atac de tip pre-play aşa cum a fost amintit şi în capitolul 4. Atacul constă în stocarea elementelor lanţului one-way de către un adversar pentru o potenţială impersonare ulterioară a utilizatorului. Un astfel de atac nu poate fi aplicat asupra protocolului propus deoarece doar câte o cheie este eliberată la un moment dat, şi o nouă cheie va fi eliberată doar la momentul unei confirmări din partea partenerului de comunicare. Este aşadar imposibil pentru un adversar să extragă chei din lanţ pentru impersonarea ulterioară a unui participant.

Un atac de tip man-in-the-middle asupra unei variante a CSA a fost descris de Perrig et al., atacul este eliminat prin propunerea din [10]. Un astfel de atac nu este posibil asupra propunerii de protocol DeMA deoarece scenariul adresat are doar un emiţător şi un receptor a căror lanţuri se iniţializează într-un stadiu off-line. De asemenea atacul nu este posibil nici pentru iniţializarea on-line cu semnături digitale propusă pentru protocolul DeMA în [51]. Metoda de iniţializare garantează celor două parţi faptul că lanţurile nu au mai fost folosite anterior şi permite informarea participanţilor asupra valorilor de iniţializare:

$$A \longrightarrow B:\ \left\{A, B, N_A, Sig_A\left(A, B, N_A\right)\right\};$$

$$B \longrightarrow A:\left\{B, A, N_B,\ Sig_B\left(A, B, N_A, N_B\right)\right\};$$

$$A \longrightarrow B:\ \left\{A, \sigma_A(0),\ Sig_A\left(A, B,\ N_A, N_B, \sigma_A(0)\right)\right\};$$

$$B \longrightarrow A:\left\{B, \sigma_B(0),\ Sig_B\left(A, B,\ N_A,\ N_B, \sigma_B(0)\right)\right\}$$

Aici Sig_A, Sig_B denotă o semnătură digitală calculată de dispozitivul de conducere A respectiv de procesul condus B în timp ce N_A, N_B sunt doi parametrii varianţi în timp pentru a garanta unicitatea comunicării. În cele din urmă şi protocolul Direct Chain Authentication (DICA) din [50] poate fi utilizat în acelaşi scop sau orice alt protocol de schimb autentic de cheie.

O dovadă informală a securităţii protocolului DeMA poate fi uşor trasată. Securitatea protocolului se bazează pe faptul că având $M_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(M_{A,k}\right), \sigma_A(k)$ nu se poate calcula $MAC_{KD(\sigma_A(k+1))}\left(M'\right)$ pentru orice alt $M' \neq M_{A,k}$. Deoarece funcţia f este one-way, din $\sigma_A(k) = f^{\eta-k}\left(x_A\right)$ nu se poate calcula $\sigma_A(k+1) = f^{\eta-k-1}\left(x_A\right) = f^{-1}\left(\sigma_A(k)\right)$ şi neavând cheia MAC-ului este evident că MAC-ul nu poate fi falsificat. Din aceste motive, singura garanţie necesară este că nu a fost distribuită cheia $\sigma_A(k+1)$ la momentul când pachetul $M_{A,k}, MAC_{KD(\sigma_A(k+1))}\left(M_{A,k}\right), \sigma_A(k)$ este trimis. Acest lucru este garantat deoarece A va

distribui valoarea $\sigma_A(k+1)$ doar când confirmarea $\sigma_B(k)$ este primită de la *B* şi evident un potenţial adversar nu poate falsifica $\sigma_B(k)$ deoarece funcţia *f* este one-way aşa cum s-a precizat. O demonstraţie formală de securitate poate fi uşor trasată pe baza modelului ROM într-o manieră similară cu cele din secţiunea 4.5 dar considerăm că o astfel de demonstraţie nu este necesară fiind un simplu exerciţiu. De asemenea, demonstraţii formale de securitate pentru protocoale similare se găsesc în [10], [95].

5.2 Utilizarea protocolului de autentificare DeMA în conducerea robotului X-80

Aşa cum este arătat de lucrări recente, utilizarea criptografiei în domeniul sistemelor de control la distanţă este o provocare majoră în prezent când aceste sisteme au început să comunice prin reţele publice unde informaţia este expusă în faţa unor potenţiali adversari [30]. Dificultatea în utilizarea tehnicilor criptografice îmbracă două aspecte. Primul aspect este datorat necesităţilor asupra puterii de calcul şi comunicare a echipamentelor, deoarece aceste resurse sunt în general constrânse în sistemele industriale. Cel de-al doilea aspect este datorat implicării criptografiei în dinamica procesului aceasta putând cauza întârzieri şi incertitudini legate de primirea informaţiilor necesare controlului.

Din aceste considerente, prima problemă care trebuie rezolvată este cea a puterii de calcul şi comunicare necesare, protocoalele criptografice trebuind adaptate la un consum minim de resurse. Pentru aceasta au fost propuse diverse protocoale, de exemplu cel din [125] care poate fi utilizat pentru a asigura securitatea pe liniile de comunicare între echipamente SCADA. În ceea ce priveşte cel de-al doilea aspect, problema care trebuie rezolvată este faptul că în comunicarea prin reţele publice pot apărea întârzieri sau chiar incertitudini cu privire la timpul de sosire al comenzilor şi răspunsurilor. Pentru rezolvarea acestei probleme au fost propuse diverse tehnici de reglare care funcţionează inclusiv în prezenţa unor incertitudini în comunicare [72].

Interesul nostru este cu privire la primul aspect, mai exact dezvoltarea unor protocoale criptografice eficiente care necesită resurse de calcul scăzute şi au proprietăţi solide de securitate. Vom evita utilizarea unei soluţii standardizate, precum SSL deoarece nu suntem interesaţi de criptarea unui canal de comunicare pentru a asigura confidenţialitatea informaţiei, în schimb suntem interesaţi de autenticitatea acesteia. Este recunoscut că în sisteme de control la distanţă autenticitatea este mult mai importantă decât confidenţialitatea deoarece informaţia nu poate fi utilizată atâta timp cât nu există garanţii asupra sursei şi prospeţimii sale. În acest scop vom utiliza protocolul DeMA bazat pe lanţuri one-way descris anterior care diferă semnificativ de paradigma SSL. Meritul acestei abordări este în primul rând ca experiment deoarce putem trage concluzii cu privire la eficienţa protocoalelor bazate pe lanţuri one-way. În al doilea rând protocoale bazate pe lanţuri one-way nu necesită funcţii de criptare asimetrică aşa cum necesită paradigma SSL şi pot fi utilizate acolo unde criptarea asimetrică trebuie evitată şi sunt disponibile doar funcţii one-way fără trapă.

5.2.1 Aplicaţia dezvoltată

Un robot X-80 este conectat la un calculator local în regim wireless prin protocolul 802.11. Câteva detalii tehnice despre acest robot sunt descrise în continuare, pagina producătorului este disponibilă la [28].

Robotul este amplasat pe două roţi de 18 cm diametru, fiecare dintre ele fiind conectată la un motor de curent continuu de 12V care poate fi controlat independent. Comenzile suportate de robot permit controlul celor două motoare de curent continuu în trei variante: în bucla deschisă bazat pe PWM şi în buclă închisă bazat pe poziţie respectiv pe viteză. Menţionăm că în aplicaţie a fost folosit controlul prin poziţie pentru care a fost utilizată comanda *PositionTimeCtrAll* din pachetul software al robotului, comandă care primeşte ca argument pentru definirea poziţiei numărul de impulsuri la care trebuie să ajungă codificatorul de rotaţie de pe fiecare roată. Regulatoarele pentru cele două roţi sunt de tipul PID (Proportional Integrator Derivator), valorile pentru parametrii PID, i.e. k_P, k_I, k_D, putând fi setate prin funcţiile puse la dispoziţie de către producător. Valorile utilizate în aplicaţia de test sunt cele folosite şi de către producător în aplicaţia demonstrativă.

Robotul este echipat cu următorii senzori: senzori ultrasonici (sonare), senzori infraroşu, senzori de detecţie a prezenţei umane, senzori de temperatură. În aplicaţia noastră am utilizat cele trei sonare dar aplicaţia poate fi uşor extinsă pentru a prelua informaţie de la oricare dintre senzorii robotului. De asemenea robotul este dotat cu camera video care poate achiziţiona imagini la rezoluţia de 352x288 pixeli; producătorul indicând o rată de achiziţie de cel mult 4 fps (în aplicaţia noastră am achiziţionat imagini la o rată de 1 fps). Camera este acţionată de un servo-motor care permite mişcarea pe orizontală şi verticală a acesteia. Alte dispozitive mai sunt ataşate la robot, precum un microfon şi un difuzor, pentru detalii suplimentare poate fi consultată documentaţia robotului [28].

Aşa cum este sugerat în figura 5.3 robotul este conectat la un calculator, care joacă rolul unui dispozitiv de conducere local, prin intermediul unui router wireless. Acest calculator local joacă de asemenea rolul de server şi acceptă conexiuni de la calculatoare externe. Comunicarea între robot şi aplicaţia de pe calculatorul local este făcută cu ajutorul unei aplicaţii gateway furnizate de producător cu kitul de instalare al robotului, comenzile către acesta fiind transmise din Visual Studio prin intermediul unui control ActiveX. Producătorul indică o rată de transfer de 12 Hz cu care comenzile pot fi trimise către robot. Pentru configurare, robotul dispune de o interfaţă web care poate fi uşor accesată. Deoarece conexiunea wireless dintre robot şi calculatorul local suportă securitate WEP nu suntem interesaţi de asigurarea securităţii pe acest traseu şi ceea ce interesează va fi asigurarea securităţii între serverul local şi clientul aflat la distanţă. Pentru acest scop am construit o aplicaţie client care a fost rulată pe un laptop de pe care ne-am conectat la server şi au fost trimise comenzi către robot, astfel clientul jucând rolul unui dispozitiv de conducere la distanţă. În scenariul astfel creat utilizarea criptografiei a devenit necesară deoarece pachetele trimise între client şi server au circulat prin reţele nesigure şi puteau fi uşor interceptate de adversari.

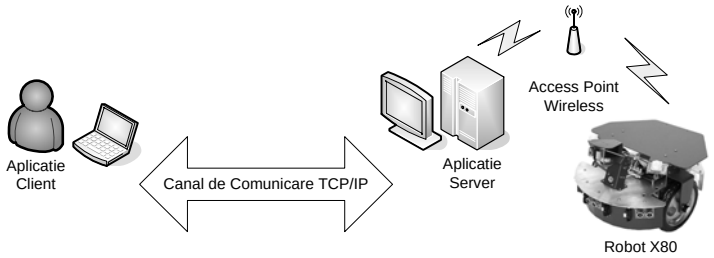


Figura 5.3. Cadrul de desfăşurare a scenariului de control a robotului X-80

În figura 5.4 este prezentat cadrul aplicaţiei sub aspectul unui sistem de control. Obiectivul aplicaţiei este controlul mişcării robotului între puncte ţintă, controlul făcându-se pe bază de poziţie folosind valorile din codificatoarele de rotaţie ale roţilor. De asemenea către clientul aflat la distanţă sunt trimise imaginile capturate de la camera video ataşată robotului. Pentru testele ale căror rezultate sunt menţionate în secţiunea următoare controlul a fost făcut manual de către utilizator prin transmiterea către robot a mişcărilor de bază: deplasări stânga, dreapta, sus, jos. Orice alt algoritm de reglare poate fi însă uşor implementat în aplicaţie. Subiectul studiului din această secţiune este însă perfomanţa în comunicare şi nu algoritmul de control care este utilizat. Subliniem că toată informaţia este garantată ca fiind autentică deoarece este vehiculată prin intermediul protocolului DeMA descris anterior.

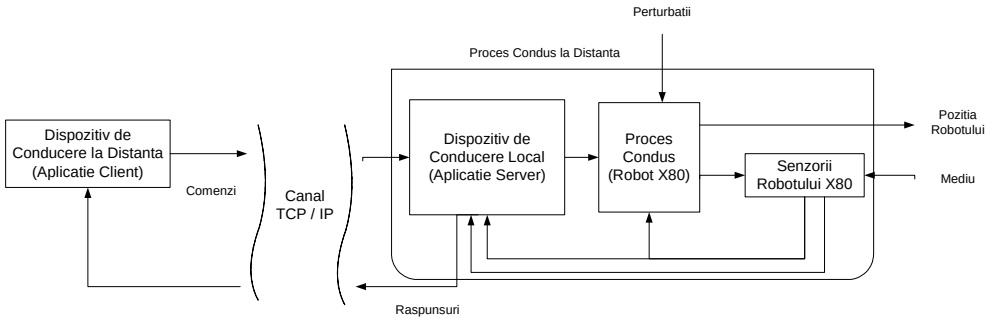


Figura 5.4. Privire asupra aplicaţiei de control a robotului X-80 ca şi sistem de conducere la distanţă

În ceea ce priveşte implementarea propriu-zisă a aplicaţiei, aceasta a fost realizată în C#. Biblioteca de funcţii oferită de producător a fost construită pentru versiunile anterioare cadrului .NET respectiv pentru VC++ sau Visual Basic 6. Am evitat utilizarea mediului VB 6.0 deoarece este depăşit ca şi limbaj de programare neoferind nici măcar suportul minim necesar pentru partea de criptografie, comunicare pe socketuri sau orientare pe obiecte. În acelaşi timp utilizarea VC++ ar fi fost greoaie şi ar fi necesitat prea mult timp pentru implementare. În acest

context, am considerat oportună folosirea limbajului C# şi a controlului ActiveX oferit de producător care poate fi cu uşurinţă utilizat [28]. Trebuie însă menţionat că, datorită divergenţelor între platforma .NET şi codul sursă unmanaged, controlul ActiveX a dus uneori la blocări. Un pachet software diferit de cel oferit de producător care poate fi utilizat pentru a comunica cu robotul este disponibil la [115]. Acest pachet pare să dea rezultate mai bune în comunicare şi plănuim utilizarea acestuia în lucrări viitoare pentru diverse îmbunătăţiri.

În ceea ce priveşte funcţiile criptografice induse de protocolul DeMA, s-au utilizat toate funcţiile hash şi codurile de autentificare MAC disponibile în platforma .NET: RIPEMD, MD5, SHA1, SHA256, SHA384, SHA512. Aceasta a presupus utilizarea următoarelor clase: *MD5CryptoServiceProvider*, *RIPEMD160Managed*, *SHA1Managed*, *SHA256Managed*, *SHA384Managed*, *SHA512Managed, HMACMD5, HMACRIPEMD160, HMACSHA1, HMACSHA256, HMACSHA384, HMACSHA512.* Remarcăm că implementarea funcţiei SHA1 din clasa *SHA1CryptoServiceProvider* oferă timpi de calcul mai mari decât implementarea aceleiaşi funcţii din clasa *SHA1Managed*.

Protocolul de comunicare a fost implementat pe baza socketurilor TCP/IP disponibile în platforma .NET. Rezultatele experimentale obţinute sunt prezentate în secţiunea următoare.

5.2.2 Rezultate experimentale obţinute

Obţinerea unor rezultate experimentale este strict necesară pentru evaluarea performanţei computaţionale şi de comunicare a protocoalelor propuse. În primul rând sunt necesare rezultate cu privire la eficienţa primitivelor criptografice folosite. Pentru aceasta tabelele 5.1 şi 5.2 arată timpii calcul, exprimaţi în secunde, pentru funcţii hash şi coduri MAC. Timpii au fost estimaţi prin calcularea fiecărei funcţii de 10^6 ori şi evaluarea timpului mediu de calcul (la fiecare iteraţie intrarea funcţiei a fost ieşirea din iteraţia anterioară).

Pentru obţinerea rezultatelor experimentale cu privire la protocol aceeaşi funcţie hash care a fost utilizată pentru calculul cheilor de sesiune, i.e. elementele din lanţul one-way, a fost utilizată şi pentru calculul codului H-MAC. Aplicaţia este însă flexibilă şi permite utilizarea de funcţii distincte pentru lanţul one-way şi pentru codul MAC.

În lucrarea [33] câteva noţiuni pentru evaluarea performanţelor sistemelor industriale în comunicarea prin Internet sunt introduse şi explicate. Aceste noţiuni, adoptate de NIST (National Institute of Standards and Technology) şi ODVA (Open DeviceNet Vendor Association), provin din [18], [19]. În baza acestei terminologii, vom măsura performanţa protocolului DeMA prin evaluarea timpului unei runde dus-întors Round Trip Time (RTT), care este timpul necesar pentru calculul unei comenzi de către dispozitivul de conducere, trimiterea comenzii către procesul condus şi primirea răspunsului autentic. În aplicaţia dezvoltată acest lucru presupune execuţia celor nouă paşi sugeraţi în figura 5.5. Utilizarea valorii RTT ca şi criteriu de performanţă pentru protocolul propus este necesară deoarece alte metrici ca şi latenţa răspunsului sau latenţa comenzii nu sunt suficient de bune pentru măsurarea performanţei pentru că prin natura lui protocolul necesită un drum dus-întors al pachetului de la dispozitivul de conducere la procesul condus pentru ca autentificarea să poată fi făcută. În tabelul 5.3 numărul mediu de pachete pe secundă este dat şi de asemenea întârzierea care apare în buclă închisă. De exemplu în cel mai rău caz latenţa este de 0.02 secunde, acest caz fiind pentru

funcţia SHA512. De asemenea remarcăm faptul că numărul minim şi maxim de pachete trimis într-o secundă variază destul de mult, şi din aceste motive am prezentat doar valoarea medie. Rezultatele au fost măsurate în LAN dar aplicaţia rulează în orice reţea care suportă comunicare prin TCP/IP.

Rezultatele din tabelul 5.3, arată că este dimensiunea funcţiei hash cea care influenţează performanţa în comunicare. Este uşor de observat că în conformitate cu tabelul 5.1 timpul de calcul pentru funcţia SHA-256 şi codul MAC aferent este mai mic decât pentru MD5. În acelaşi timp, în tabelul 5.3 cea mai bună performanţă la comunicare a fost obţinută cu funcţia MD5 datorită dimensiunii scăzute a ieşirii acesteia. Din aceste motive o dimensiune scăzută a ieşirii pentru funcţia hash este preferabilă. Totuşi funcţia MD5 este cunoscută ca având câteva vulnerabilităţi, şi este puţin probabil ca în viitor să ofere un nivel suficient de securitate. Observăm că şi în cazul utilizării funcţiei SHA-512 care are cea mai mare dimensiune a ieşirii am obţinut o valoare medie de 50 pachete pe secundă care este mult peste viteza la care poate funcţiona robotul (de exemplu robotul poate transmite maxim 4 fps în timp ce aplicaţia poate transmite 50 fps). Aceasta arată în cele din urmă că utilizarea criptografiei este fezabilă în aplicaţia propusă. Trebuie de asemenea menţionat că rezultatele experimentale nu au un caracter absolut şi sunt relevante pentru mediul în care aplicaţia a fost dezvoltată, sistemul de operare Windows şi platforma .NET, în alte medii putând fi obţinute rezultate diferite. Indiferent însă de mediul de lucru, raporturile de eficienţă între funcţiile utilizate vor rămâne neschimbate, deci sub raport comparativ între diferite funcţii rezultatele ar trebui să rămână valide indiferent de mediul de implementare.

Funcţie Hash	CPU Intel T2300@1.66Ghz	CPU Intel E6750@2.66Ghz
MD5	$9.37\times10^{-6}s$	$5.15\times10^{-6}s$
RIPEMD160	$2.81\times10^{-6}s$	$1.56\times10^{-6}s$
SHA1	$2.03\times10^{-6}s$	$1.40\times10^{-6}s$
SHA-256	$3.28\times10^{-6}s$	$1.87\times10^{-6}s$
SHA-384	$9.53\times10^{-6}s$	$4.21\times10^{-6}s$
SHA-512	$9.68\times10^{-6}s$	$4.37\times10^{-6}s$

Tabelul 5.1 Timpul de calcul pentru funcţiile hash din .NET

H-MAC	Intel T2300@1.66Ghz	Intel E6750@2.66Ghz
MD5	$21.25\times10^{-6}s$	$11.56\times10^{-6}s$
RIPEMD160	$9.68\times10^{-6}s$	$5.15\times10^{-6}s$
SHA1	$22.18\times10^{-6}s$	$11.87\times10^{-6}s$
SHA-256	$10.78\times10^{-6}s$	$5.78\times10^{-6}s$
SHA-384	$35.78\times10^{-6}s$	$15.93\times10^{-6}s$
SHA-512	$35.93\times10^{-6}s$	$16.09\times10^{-6}s$

Tabelul 5.2. Timpul de calcul pentru codurile MAC din .NET

Funcţia Hash pentru chei şi MAC	Dimensiunea Ieşirii (în biţi)	Pachete/Secundă (Valoare Medie)	Timpul unui round-trip
MD5	*128*	*64*	*0.016 s*
RIPEMD160	*160*	*56*	*0.017 s*
SHA1	*160*	*61*	*0.016 s*
SHA-256	*256*	*56*	*0.017 s*
SHA-384	*384*	*52*	*0.019 s*
SHA-512	*512*	*50*	*0.020 s*

Tabelul 5.3. Statistici la comunicare cu protocolul DeMA pentru diverse funcţii criptografice

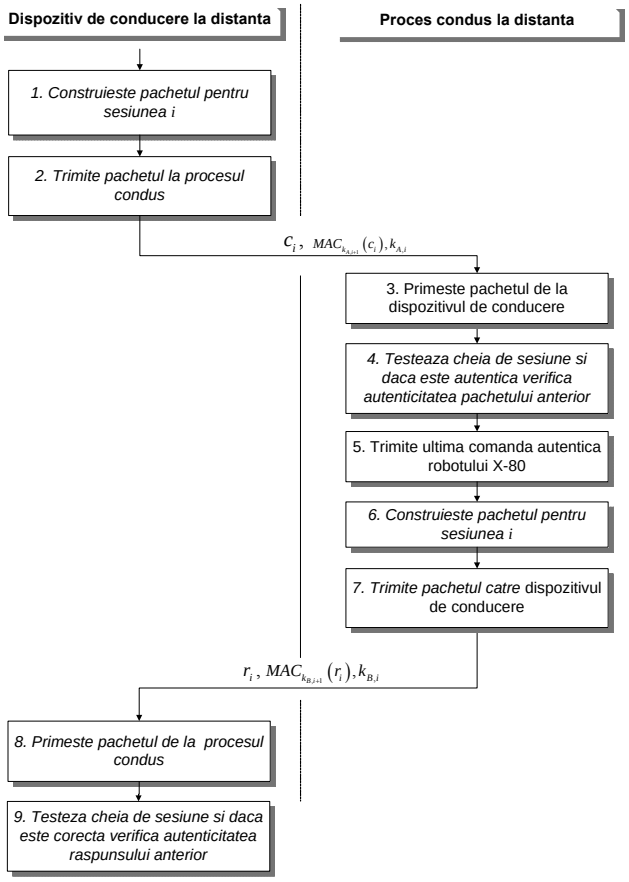


Figura 5.5. Diagrama paşilor implicaţi în comunicarea cu robotul X-80 la sesiunea i

6 Concluzii

6.1 Contribuţii aduse

Considerăm necesar un scurt sumar al contribuţiilor aduse, deoarece contribuţiile unei teze au în primul rând valoare de concluzie în ceea ce priveşte relevanţa ei. Teza are la bază peste 20 de lucrări publicate ca prim sau unic autor, peste jumătate din ele fiind indexate în diverse baze de date DBLP, ISI-Proceedings, IEEE-Xplore, INSPEC, SCOPUS etc. De asemenea au fost câştigate două granturi naţionale de tip TD care au acoperit o parte din aceste cercetări. Pentru claritatea expunerii sintetizăm următoarele contribuţii în zone de cercetare distincte aparţinând criptografiei teoretice şi aplicate în egală măsură:

1) **Introducerea unui procedeu nou pentru construcţia lanţurilor one-way folosind funcţia putere discretă**. Sunt discutate diverse cazuri de utilizare, avantajul major fiind faptul că dimensiunea lanţului poate fi nelimitată datorită posibilităţii de a reduce exponenţii modulo ordinul grupului. Secţiunea 4.2 tratează acest rezultat care se găseşte publicat în lucrările [47], [48], [52].

2) **Construcţia unui protocol de autentificare bazat în exclusivitate pe lanţuri one-way fără sincronizare temporală (DeMA-DiCA)**. Protocolul este în esenţă similar protocolului CSA dar diferă prin faptul că foloseşte lanţuri one-way inclusiv la iniţializare şi astfel nu mai necesită o semnătură digitală aşa cum s-a propus pentru CSA. Protocolul DeMA-DiCA este descris în secţiunea 4.3.1 şi publicat în lucrarea [50].

3) **Construcţia unui protocol de autentificare bazat pe lanţuri one-way nemărginite în practică (DeMA-QR)**. Lanţurile one-way construite cu funcţii hash au dezavantajul că lanţul poate fi epuizat şi necesită reiniţializare. Pentru a evita acest neajuns se pot utiliza lanţurile amintite anterior construite cu funcţia putere discretă. Protocolul DeMA-QR descris în secţiunea 4.3.2 foloseşte astfel de lanţuri pentru schimbul autentic de informaţie, rezultatul este publicat în lucrările [51], [53].

4) **Construcţia unui protocol de autentificare în regim broadcast pe termen nelimitat (Timed-DeMA-QR)**. Cazul particular, cel mai avantajos din punct de vedere computaţional, al utilizării funcţiei putere discretă în construcţia lanţurilor one-way, este cel al utilizării funcţiei ridicare la pătrat discretă care face posibilă autentificarea în regim broadcast pe termen nelimitat la costul unei singure multiplicări modulare. Chiar dacă o multiplicare modulară este de câteva zeci de ori mai intensă ca o funcţie hash, aceasta reprezintă unica alternativă cunoscută pentru un astfel de scenariu. Secţiunea

4.3.3 tratează un protocol pentru broadcast pe termen nelimitat publicat în lucrările [57], [60].

5) **Construcţia unui criptosistem cu cheie publică bazat pe extensia funcţiei RSA**. Funcţia RSA presupune utilizarea unui exponent public care este relativ prim la ordinul grupului. Extensia propusă permite utilizarea oricărui exponent indiferent de relaţia în care se află cu ordinul grupului. Dezavantajul propunerii este acela că timpul de calcul necesar este mai ridicat ca în cazul funcţiei RSA dar avantajul este securitatea demonstrată ca fiind echivalentă cu problema factorizării întregilor, lucru care încă nu a putut fi realizat pentru RSA. Pentru sporirea eficienţei schemei propuse aceasta a fost utilizată într-un cadru KEM/DEM. Secţiunea 4.4 tratează acest rezultat ce a fost publicat în lucrările [59], [55].

6) **Construcţia puzzleurilor criptografice**. A fost propus un nou mecanism de construcţie a puzzleurilor criptografice care permite crearea unor puzzleuri înlănţuite ce au ca avantaj diminuarea puterii de paralelizare a rezolvării şi posibilitatea de a da doar soluţii parţiale a puzzleului făcând astfel rezolvarea acestora şi alocarea de resurse de partea unui server aflat sub atac mai flexibilă. În secţiunea 4.6 este succint prezentat rezultatul iar publicarea s-a făcut în lucrarea [54].

7) **Aplicarea tehnicilor criptografice în sisteme informatice şi de control la distanţă**. Au fost realizate de-a lungul celor 4 ani de stagiu doctoral diverse implementări ale soluţiilor propuse, rezultate cu privire la acestea fiind publicate de autor în diverse lucrări [49], [58], [61], [63], [65]. În teză, în capitolul 5 este prezentat rezultatul obţinut pentru implementarea unui protocol de autentificare pentru controlul robotului X-80, rezultat care arată în primul rând care sunt aşteptările cu privire la performanţa unui astfel de protocol. Rezultatul a fost publicat în lucrarea [62].

8) **Analiza criptografică a protocoalelor de autentificare**. Un protocol criptografic de autentificare a entităţilor considerat sigur de unii autori a fost spart şi diverse vulnerabilităţi ale acestuia arătate. De asemenea a fost desfăşurat şi un studiu asupra unui protocol de autentificare (NTLM) folosit de câteva produse Microsoft şi au fost arătate câteva vulnerabilităţi ale acestuia. Rezultatele nu au fost incluse în prezenta lucrare din considerente de spaţiu dar au fost publicate în lucrările [46], [64].

Aceste contribuţii trebuie văzute ca soluţii alternative pentru cele deja existente în domeniu sau ca şi drumuri nou deschise. Este foarte greu, şi prea rar se întâmplă în realitate, ca soluţii propuse în cadrul unui doctorat să ajungă soluţii de mare actualitate în practică. Soluţiile astfel create pot necesita aşadar dezvoltări ulterioare.

6.2 Concluzii cu privire la criptografie

Concluziile care pot fi trase cu privire la criptografie sunt foarte variate şi acoperă o zonă destul de largă.

Poate cel mai important lucru de observat este că obiective reale de securitate şi formalisme noi asupra funcţiilor criptografice şi adversarilor activi au fost aduse în ultimul deceniu. Acest lucru a dus la regândirea totală a primitivelor criptografice utilizate în practică. Lucru care face ca în zilele noastre sisteme binecunoscute, precum RSA-ul de exemplu, să nu fie utilizate în practică în forma în care au fost propuse iniţial. Obiectivele şi modelele adversarilor par a fi destul de complete, astfel este posibil ca ele să rămână valabile fără prea multe modificări pe termen destul de lung. Rămâne deschisă problema sistemelor în ansamblu în care criptografia este doar o simplă componentă, sisteme care devin din ce în ce mai complexe şi o dată cu creşterea complexităţii lor va creşte şi numărul de atacuri posibile. În acest sens metodele de verificare automată a sistemelor vor avea multe de spus.

Chiar dacă obiectivele de securitate şi modele adversarilor sunt suficient de solide, metodele de a demonstra că un sistem este cu adevărat sigur sunt încă incomplete. Demonstraţiile automate de securitate vor juca un rol important în viitor. Totuşi acestea nu pot fi folosite suficient de bine la momentul de faţă pentru a demonstra securitatea funcţiilor criptografice. Singurul mod de a realiza acest lucru este Modelul Oracolului Aleatoare (ROM), model care a fost dovedit ca fiind inconsistent în diverse situaţii, dar care rămâne singura alternativă. În viitorul cât mai scurt este posibil să apară alte tehnici şi modele de a demonstra securitatea criptosistemelor. Rămân de asemenea deschise potenţiale probleme în soluţiile criptografice bazate pe probleme de teoria numerelor încă nedemonstrate. Este relevant de remarcat că există soluţii utilizate pe scară largă precum RSA-ul despre a cărui securitate nu s-a putut demonstra ca fiind echivalentă cu problema factorizării în ciuda a 30 de ani de evoluţie şi a faptului că factorizarea modulului rămâne singura variantă de a sparge complet securitatea acestui criptosistem.

Dacă în trecut construcţiile criptografice erau gândite doar pentru a atinge anumite obiective în prezent drumul criptografiei este către soluţii eficiente şi optimale. Sunt utilizate în practică soluţii mult mai eficiente decât în trecut, chiar de mii de ori în unele cazuri dacă facem de exemplu o comparaţie între primul protocol care avea proprietatea IND propus de Goldwasser şi Micali şi RSA-OAEP care se utilizează în prezent şi are aceeaşi proprietate. De asemenea soluţiile moderne ating şi obiective de securitate din ce în ce mai avansate. Relativ recent a fost adusă şi prima schemă de semnătură digitală demonstrată ca fiind optimală în ipoteza respectivă de lucru, cu alte cuvine nu se poate face mai mult decât atât. Pe termen scurt şi mediu este posibil să mai apară şi alte soluţii optimale, precum şi să sporească eficienţa tehnicilor curente, acest lucru fiind strict necesar deoarece odată cu creşterea puterii de calcul creşte şi puterea de a ataca primitivele criptografice. În acest context soluţiile bazate pe construcţii hibride între tehnicile simetrice şi asimetrice vor fi din ce în ce mai frecvente în practică, fiind în cele din urmă împletirea armonioasă a acestora cea care poate conduce la soluţii eficiente. Tot în acest context, al eficienţei, soluţii precum protocoalele bazate pe lanţuri one-way şi sincronizare temporală, sau în general protocoale care sunt construite pe primitive simetrice şi totuşi au proprietăţi de securitate asimetrice vor fi din ce în ce mai răspândite în practică. Poate din acelaşi considerent, dar şi din cel amintit anterior, al lipsei de demonstraţii pentru securitatea unor sisteme bazate pe probleme de teoria numerelor, vom asista în următorii ani la intrarea în practică a unor alte zone

până acum marginalizate din criptografie, precum semnăturile digitale one-time care oferă caracteristici asimetrice în ciuda utilizării unor funcţii simetrice.

În mare, cu privire la criptografie se poate concluziona că oferă soluţii reale, eficiente şi sigure pentru problemele cu care se confruntă societatea informaţională. Utilizarea criptografiei în practică va fi din ce în ce mai intensă deoarece soluţiile oferite de acest domeniu oferă cel mai ridicat nivel de securitate.

6.3 Concluzii cu privire la utilizarea criptografiei în sisteme de control

Cu privire la utilizarea criptografiei în sisteme de control la distanţă cea mai relevantă concluzie, care poate fi regăsită în marea partea a lucrărilor dar care este demonstrată şi de prezentul material este că utilizarea criptografiei este fezabilă şi nu alterează cu mult performanţele computaţionale sau de comunicare. Nu este însă clar dacă sunt reale problemele legate de constrângerile cu privire la puterea de calcul şi comunicare prezentă în sistemele industriale. Aceasta în primul rând pentru că resursele în discuţie variază mult prea tare de la un scenariu la altul. Este semnificativ în acest caz compararea unui exemplu de tele-medicină, care implică control automat pentru braţul de robot folosit pentru operaţii pe subiecţi umani la distanţă şi ratele de transfer sunt la ordinul zecilor de megabiţi pe secundă, cu un alt scenariu frecvent în mediile industriale în care comunicarea se face pe portul serial şi rata de transfer este doar la ordinul zecilor de kilobiţi pe secundă. Evident există o discrepanţă majoră între aceste scenarii. Tot din aceste considerente utilizarea unor soluţii standardizate este încă discutabilă. Pentru aducerea în practică a unor soluţii standardizate poate că sunt necesari încă 5-10 ani pentru maturizarea acestui domeniu, a utilizării criptografiei în sisteme industriale.

Importanţa criptografiei în securitatea sistemelor industriale şi de control trebuie din ce în ce mai mult conştientizată, utilizarea fiind încă neglijată în unele zone şi rolul ei subestimat. În acest sens, în viitorul apropiat este cert că numărul de atacuri reuşite va creşte şi aceasta va face ca necesitatea securităţii criptografice să fie luată şi mai în serios, sperăm că aceasta nu se va întâmpla doar după ce vom asista la un atac major în următorii ani sau următoarele decenii. De fapt utilizarea criptografiei în sisteme informatice a fost condiţionată de existenţa unor atacuri serioase asupra sistemelor în cauză. În mare parte datorăm larga răspândire a securităţii în domeniul calculatoarelor, sistemelor de operare etc. faptului că a existat o spaimă constantă de partea agenţiilor de securitate cu privire la expunerea acestei informaţii. Este sigur că va fi doar o chestiune de timp până când vom porni autoturismele folosind chei criptografice, şi acest orizont de timp probabil se referă la câţiva ani, deoarece inclusiv în prezent există astfel de soluţii în practică. Deoarece multe carduri bancare folosesc funcţii criptografice precum funcţii hash sau DES, putem spune că umblăm cu DES-ul în buzunar, este însă doar o problemă de timp până vom umbla cu curbe eliptice în buzunar şi vom asculta informaţie decriptată când vorbim la telefonul mobil.

Domeniul utilizării criptografiei în sisteme informatice şi în special în sisteme de control, este un domeniu aflat încă în anii tinereţii sale, lucru care îl face dinamic şi atractiv, dornic de noi soluţii, dar totodată vulnerabil şi nesigur.

Bibliografie

[1] R. Anderson, F. Bergadano, B. Crispo, J.H. Lee, C. Manifavas, R. Needham, A New Family of Authentication Protocols, ACM Operating Systems Review, ACM, 1998.

[2] R. J. Anderson, H. Chan, A. Perrig, Key infection: Smart trust for smart dust. In 12th IEEE International Conference on Network Protocols, pp. 206-215, IEEE, 2004.

[3] T. Aura, P. Nikander, J. Leiwo, DOS-Resistant Authentication with Client Puzzles, LNCS 2845, Springer, 2001.

[4] J. Bayne, An Overview of Threath and Risk Assesment, SANS Institute, 2002.

[5] M. Bellare, P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, ACM Conference on Computer and Communications Security, pages 62–73, 1993.

[6] M. Bellare, P. Rogaway, Optimal asymetric encryption – How to encrypt with RSA, Advances in Cryptology – EuroCrypt 94, LNCS 950, Springer, 1995.

[7] M. Bellare, R. Canetti, H. Krawczyk, Keying Hash Functions for Message Authentication, Advances in Cryptology – CRYPTO 96, LNCS 1109, Springer, 1996.

[8] M. Bellare, S. Goldwasser, Verifiable Partial Key Escrow, ACM Conference on Computer and Communications Security 1997, pp. 78-91, ACM, 1997.

[9] M. Bellare, A. Desai, D. Pointcheval, P. Rogaway, Relations among notions of security for public-key encryption schemes. Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology, LNCS, 1462:26 – 45, 1998.

[10] F. Bergadano, D. Cavagnino, B. Crispo, Individual Authentication in Multiparty Communications. Computer & Security, vol. 21 n. 8, pp.719-735, Elsevier Science, 2002.

[11] K. Bicakci, N. Baykal, Infinite Length Hash Chains and Their Applications, 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises WETICE, IEEE, 2002.

[12] K. Bicakci, N. Baykal, Improving the Security and Flexibility of One-time Passwords by Signature Chains, Turkish Journal of Electrical Engineering & Computer Sciences, 11, p.223-236, 2003.

[13] D. Bleichenbacher, U. Maurer, Directed Acyclic Graphs, One-way Functions and Digital Signatures, Advances in Cryptology CRYPTO94, 75-82, LNCS 839, Springer-Verlag, 1994.

[14] D. Bleichenbacher, U. Maurer, On the Efficiency of One-time Digital Signatures, Advances in Cryptography ASIACRYPT 96, pp. 145—58, LNCS 1163, Springer-Verlag, 1996.

[15] L. Blum, M. Blum, M. Shub, Comparison of Two Pseudo-Random Number Generators, Advances in Cryptology Proceedings of Crypto 82, pp. 61-78, 1982.

[16] L. Blum, M. Blum, M. Shub, A Simple Unpredictable Pseudo-Random Number Generator, SIAM Journal on Computing, Volume 15 , Issue 2 , pp. 364 – 383, 1986.

[17] D. Boneh, R. Venkatesan, Breaking rsa may not be equivalent to factoring, Proceedings of Eurocrypt 98, LectureNotes in Computer Science, vol. 1233, pp. 59–71, Springer-Verlag, 1998.

[18] S. Bradner, Benchmarking Terminology for Network Interconnection Devices, RFC 1242, 1991.

[19] S. Bradner, J. McQuaid, Benchmarking Methodology for Network Interconnection Devices, RFC 2544, 1999.

[20] E. Byres, J. Lowe, The Myths and Facts behind Cyber Security Risks for Industrial Control Systems. VDE Congress'04, 2004.

[21] R. Canetti, O. Goldreich, S. Halevi, The random oracle methodology, revisited, Journal of the ACM (JACM), Volume 51 , Issue 4, 2004.

[22] S. Cheung, An efficient message authentication scheme for link state routing, ACSAC, p. 90--98, 1997.

[23] H-Y. Chien, J-K. Jan, Robust and Simple Authentication Protocol. Oxford Journal, The Computer Journal, Vol. 46, No. 2, 2003.

[24] D. Coppersmith, M. Jakobsson, Almost Optimal Hash Sequence Traversal, Sixth International Conference on Financial Cryptography 2002, LNCS 2357, Springer, 2003.

[25] R. Cramer, V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosenciphertext attack, SIAM Journal on Computing, vol. 33, Issue1, pp. 167 – 226, 2004.

[26] W. Diffie, M.E. Helmann, New directions in cryptography, IEEE Transactions on Information Theory, 1976.

[27] D. Dolev, C. Dwork, M. Naor, Non-malleable cryptography, Proceedings of the 23rd Symposium on Theory of Computing, ACM STOC, pages 542–552, 1991.

[28] Dr. Robot Inc., Developer and manufacturer of mobile robotics technol-ogy, http://www.drrobot.com/.

[29] W. Du, R. Wang, P. Ning, An Efficient Scheme for Authenticationg Public Keys in Sensor Networks. Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc, ACM/IEEE, 2005.

[30] D. Dzung, M. Naedele, T.P.Hoff, M. Crevatin, Security for Industrial Communication Systems, vol. 93, no. 6, IEEE Trans., 2005.

[31] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, 1985.

[32] S. Even, O. Goldreich, S. Micali, On-line/offline Digital Signatures. In Journal of Cryptology pp 35-67, Springer, 1995.

[33] J. Falco, J. Gilsinn, K. Stouffer, IT Security for Industrial Control Systems: Requirements Specification and Performance Testing, NDIA Homeland Security Symposium & Exhibition, 2004.

[34] M. Felser, T. Sauter, Standardization of Industrial Ethernet – the Next Battlefield?, 5th IEEE International Workshop on Factory Communication Systems IEEE, 2004.

[35] FIPS 46, Data Encryption Standard (DES), National Institute of Standards and Technology (NIST)., U.S. Department of Commerce, 1976, re-înnoit în 1988, 1993, 1999 ca FIPS 46-1, 46-2, 46-3.

[36] FIPS 180-1, National Institute of Standards and Technology (NIST). Announcing the Secure Hash Standard, U.S. Department of Commerce, FIPS, 1995.

[37] FIPS 197, Announcing the Advanced Encryption Standard. http://csrc.nist.gov/publications/fips/fips197/ fips-197.pdf, FIPS, 2001.

[38] M. Fischlin, Fast Verification of Hash Chains, Topics in Cryptology – CT-RSA 2004, Springer, 2004.

[39] E. Fujisaki, T. Okamoto, How to enhance the security of public-key encryption at minimum cost, Lecture Notes in Computer Science, 1560:53–68, 1999.

[40] E. Fujisaki, T. Okamoto, D. Pointcheval, J. Stern, RSA– OAEP is secure under the RSA assumption. Advances in Cryptology — Proceedings of CRYPTO '2001 (19 – 23 august 2001, Santa Barbara, California, USA), Lecture Notes in Computer Science, 2139, 2001.

[41] Gao, Critical Infrastructure Protection, Challenges and efforts to secure control systems, United States General Accounting Office, 2004.

[42] S. Goldwasser, S. Micali, R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, Society for Industrial and Applied Mathematics, Journal of computing, 17(2), pp. 281-308, 1988.

[43] S. Goldwasser, S. Micali, Probabilistic encryption. Journal of Computer and System Sciences, 28:270-299, 1984.

[44] V. Goyal, How To Re-initialize a Hash Chain.URL: http://eprint.iacr.org/2004/097.pdf, 2004.

[45] Groza B., Introducere in criptografia cu cheie publică, 2007, 136 pagini, Editura Politehnica, ISBN 978-973-625-654-9.

[46] B. Groza, D. Petrica, Cryptanalysis of an Authentication Protocol, Proceedings of 7th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC'05, pp. 147-153, IEEE Comp. Soc., 2005.

[47] B. Groza, D. Petrica, T.L. Dragomir, A time-memory trade to generate one-time passwords using quadratic residues over Zn, Studies in Informatics and Control vol. 14 no. 3, 2005.

[48] B. Groza, D. Petrică, One time passwords for uncertain number of authentications, CSCS-15, 15th International Conference on control systems and computer science, Politehnica University of Bucharest, 2005.

[49] B. Groza, D. Petrica, T.L. Dragomir, Security based on cryptographic techniques for remote control systems, SINTES, XII International Symposium on System Theory, Oct. 20-22, Craiova, Proceedings, Vol 4 Computer Engineering, ISBN 973-742-148-5, 973-742-154-X, pp. 729-734, 2005.

[50] B. Groza, Using one-way chains to provide message authentication without shared secrets, 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPeru'06, Lyon, France, pp. 82-87, IEEE Comp. Soc, 2006.

[51] B. Groza, D. Petrica, T.L. Dragomir, Using the Discrete Squaring Function in the Delayed Message Authentication Protocol, Proceedings of International Conference on Internet Surveillance and Protection, ICISP'06, Cap-Esterel, France, IEEE Comp. Soc., 2006.

[52] B. Groza, Construction techniques for one-way chains and their use in authentication, Control Engineering and Applied Informatics Journal, vol. 8, no. 1, ISSN 1454-8658, pp. 42-51, 2006.

[53] B. Groza, The Delayed Message Authentication Protocol with Chains Constructed on the Discrete Power Function, 7th International Conference on Technical Informatics CONTI'2006, ISBN 973-625-319-X, pp. 33-36, 2006.

[54] B. Groza, D. Petrica, On chained cryptographic puzzles, 3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, SACI'2006, ISBN 963-7154-46-9, pp. 182-191, 2006.

[55] B. Groza, An extension of the RSA trapdoor in a KEM/DEM Framework, Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC'07, IEEE Comp. Soc., 2007.

[56] B. Groza, T.L. Dragomir, On the use of one-way chain based authentication in secure control systems,Second International Conference on Availability, Reliability and Security (ARES'07), International Workshop on Advances in Information Security (WAIS'07), Vienna, Austria, pp. 1214-1221, IEEE Comp. Soc., 2007.

[57] B. Groza, Broadcast authentication protocol with time synchronization and quadratic residues chains, Second International Conference on Availability,

Reliability and Security (ARES'07), International Symposium on Frontiers in Availability, Reliability and Security (FARES'07), Vienna, Austria, pp. 550-557, IEEE Comp. Soc., 2007.

[58] B. Groza, D. Petrica, S. Barbu, M. Bilanin, Implementation of an Authentication Protocol for Sending Audio-Video Information in Java, SACI 2007, IEEE Comp. Int. Soc., 2007.

[59] B. Groza, On the use of the discrete power function for building public-key cryptosystems, Applied Informatics & Communications, WSEAS Press, Greece, pp. 7-11 (Best Paper Award), 2007.

[60] B. Groza, Broadcast authentication with practically unbounded one-way chains, JOURNAL OF SOFTWARE (JSW), Volume 3, Issue 2 (acceptata spre publicare in 2007 aparuta in Februarie 2008), ISSN : 1796-217X, Academy Publishers, Finlanda, 2008.

[61] B. Groza, P.S. Murvay, I. Silea, T. Ionica, Cryptographic authentication on a 8051 based development board, The Third International Conference on Internet Monitoring and Protection, ICIMP 2008, 2008.

[62] B. Groza, T.L. Dragomir, Using a Cryptographic Authentication Protocol for the Secure Control of a Robot over TCP/IP, IEEE-TTTC International Conference on Automation, Quality & Testing, Robotics, AQTR 2008 (THETA 16), 2008.

[63] B. Groza, D. Pop, I. Silea, Java Implementation of an Authentication Protocol with Application on Mobile Phones, IEEE-TTTC International Conference on Automation, Quality & Testing, Robotics, AQTR 2008 (THETA 16), 2008.

[64] B. Groza, A. Alexandroni, I. Silea, An Overview of the NTLM Authentication and its Weaknesses in SharePoint Solutions, THE 8th INTERNATIONAL CONFERENCE ON TECHNICAL INFORMATICS, CONTI 2008, 2008.

[65] B. Groza, Putanu E.A., T.-L. Dragomir, D. Petrica, Development of a Client-Server Platform for Simulation of Remote Control Systems from Matlab, National Conference on Electrical Drives, CNAE 2008, 2008.

[66] N. Haller, C. Metz, P. Nesser, M. Straw, The S/KEY One-Time Password System, Internet RFC 1760, 1995.

[67] N. Haller, C. Metz, P. Nesser, M.Straw, A One-Time Password System. Internet RFC 2289, 1998.

[68] R. Hauser, T. Przygienda, G. Tsudik, Reducing the cost of security in link-state routing, in Symposium of Network and Distributed Systems Security, 1997.

[69] J. Herranz, D. Hofheinz, E. Kiltz, The Kurosawa-Desmedt Key Encapsulation is not Chosen-Ciphertext Secure, http://eprint.iacr.org/2006/207.pdf, 2006.

[70] Y.-C. Hu, D. B. Johnson, A. Perrig, SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks, in The 4th IEEE Workshop on Mobile Computing Systems and Applications, IEEE, 2002.

[71] Y.-C. Hu, M. Jakobsson, A. Perrig, Efficient Constructions for One-way Hash Chains, Proceedings of Applied Cryptography and Network Security (ACNS 2005), LNCS 2947, Springer, 2005.

[72] O. C. Imer, S. Yuksel, T. Basar, Optimal control of lti systems over unreliable communication links, Automatica, (42), 2006.

[73] Internet Security Systems, Security Best Practices for SCADA Networks and Process Management Systems, http://documents.iss.net/marketsolutions/SCADABrochure.pdf, 2004

[74] M. Jakobsson, Fractal hash sequence representation and traversal, IEEE International Symposium on Information Theory, IEEE, 2002.

[75] A. Juels, J. Brainard, Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. Proceedings of Networks and Distributed Security Systems 99, pp. 151-165, 1999.

[76] K. Kurosawa, Y. Desmedt, A New Paradigm of Hybrid Encryption Scheme, Advances in Cryptology - CRYPTO 2004, LNCS vol. 3152, pp. 426-442, Springer-Verlag, 2004.

[77] S. Kuvshinkova, SQL SLAMMER worm lessons learned for consideration by the electricity sector 9/5/2003, www.myitforum.com/articles, 2003.

[78] K. Lakshminarayanan, D. Adkins, A. Perrig, I. Stoica, Taming IP Packet Flooding Attacks, 2nd ACM Workshop on Hot Topics Networks, Cambridge, MA, ACM, 2003.

[79] L. Lamport, Password Authentication with Insecure Communication. Communication of the ACM, 24, 770-772, ACM, 1981.

[80] A. K. Lenstra, Further progress in hashing cryptanalysis, http://cm.bell-labs.com/who/akl/hash.pdf, 2005.

[81] D. Liu, P. Ning., Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks, 10th Network and Distributed System Security Symp., pp. 263-276, 2003.

[82] B. Lu, U.W. Pooch, A Lightweight Authentication Protocol for Mobile Ad Hoc Networks, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), pp. 546 - 551, 2005.

[83] W. Mao, Modern Cryptography: Theory and Practice, Prentice Hall PTR, 740 pagini, ISBN 0130669431, 2003.

[84] McAfee, Mitigating the Top 10 Network Security Risks in SCADA and Process Control Systems, A McAfee IntruShield solution guide, disponibil la http://www.mcafee.com/us/local_content/white_papers/wp_cor_scada_001_0407.pdf, 2007.

[85] A.J. Menezes, P.C. Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 816 pagini, ISBN 0849385237, 1996.

[86] R. C. Merkle, A digital signature based on a conventional encryption function. In CRYPTO '87, pages 369-378, LNCS 293, Springer, 1988.

[87] V. S. Miller, Use of elliptic curves in cryptography, (1986) Advances in Cryptology, Proceedings of Crypto'85, LNCS 218, 1985.

[88] C. Mitchell, L. Chen, Comments on the S/KEY User Authentication Scheme, Operating Systems Review, 1996.

[89] C. J. Mitchell, Remote user authentication using public information, Cryptography and Coding, 9th IMA International Conference on Cryptography and Coding, Cirencester, pp.360-369, LNCS 2898, Springer, 2003.

[90] K. Q. Nguyen, Y. Mu, V. Varadharajan, Digital coins based on hash chain Proceedings of the 20th National Information Systems Security Conference, Baltimore, USA, pp. 72-79, 1997.

[91] NRC, Information Notice 2003-14, Potential of Plant Computer Network to Worm Infection, NRC, 2003.

[92] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J.D.Tygar, SPINS: Security Protocols for Sensor Network, Proceedings of Seventh Annual International Conference on Mobile Computing and Networks MOBICOM, ACM/IEEE, 2001.

[93] A.Perrig, R. Canetti, J. D. Tygar, D. Song, Efficient Authentication and Signing of Multicast Streams Over Lossy Channels, IEEE Symposium on Security and Privacy, IEEE, 2000.

[94] A.Perrig, The BiBa one-time signature and broadcast authentication protocol, Proc. of ACM Conference on Computer and Communications Security, pp.28-37, ACM/IEEE, 2001.

[95] A. Perrig, R. Canetti, J. D. Tygar, D. Song, The TESLA Broadcast Authentication Protocol, In RSA CryptoBytes, 5:2, Summer/Fall, pp. 2-13, 2002.

[96] A. Perrig, R. Canetti, D. Song, D. Tygar, Efficient and Secure Source Authentication for Multicast, Proceedings of Network and Distributed System Security Symposium, 2001.

[97] A. Perrig, Y.-C. Hu, A Survey of Secure Wireless Ad Hoc Routing, IEEE Security and Privacy Volume 2, Issue 3, IEEE, 2004.

[98] A. Perrig, J. Stankovic, D. Wagner, Security in wireless sensor networks. Communication of the ACM, 47(6):53-57, ACM, 2004.

[99] M. Rabin, Digitalized signatures and public key functions as intractable as factorization, MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[100] L. Reyzin, N. Reyzin, Better than Biba: Short One-Time Signatures with Fast Signing and Verifying, ACISP 2002, LNCS 2384, Springer, 2002.

[101] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, 1978.

[102] R. Rivest, A. Shamir, Payword and Micromint: Two simple micropayment schemes. CryptoBytes, volume 2, no. 1, RSA Laboratories, 1996.

[103] L. Rivest, A. Shamir, D.A. Wagner, Time-lock puzzles and timed-release Crypto, available at http://theory.lcs.mit.edu/~rivest/publications.html, 2000.

[104] W. F. Rush, J. A. Kinast, Here's what you need to know to protect SCADA systems from cyber-attack, Pipeline & Gas Journal, 2003.

[105] B. Schneier, Applied Cryptography, John Wiley & Sons, 784 pagini, ISBN 0471117099, 1996.

[106] B. Schneier, Cryptanalysis of MD5 and SHA: Time for a New Standard, http://schneier.com/essay-074.html, 2004.

[107] Y. Sella, On the Computation-Storage Trade-offs of Hash Chain Traversal, preprint, 2003.

[108] Y. Sella, Double Hash Chains, preprint, 2003.

[109] E. Shi, A. Perrig, Designing secure sensor networks. IEEE Wireless Communications, 11(6), IEEE, 2004.

[110] V. Shoup, OAEP reconsidered. Lecture Notes in Computer Science, 2139, 2001.

[111] V. Shoup, A proposal for an ISO standard for public key encryption. Input for Committee, 2001.

[112] V. Shoup, Computational Introduction to Number Theory and Algebra, disponibilă la www.shoup.net/ntb, 2004.

[113] T. Simcock, Power Facility under Fire, Cyber Warrior, Assigment Version: 1.0, Assigment Option 2, SANS Institute, 2004.

[114] T. Stephanou, Assesing and exploiting the internal security of an organization, SANS Institute, 2001.

[115] T. Tayor, X80 WiRobot, http://sky.fit.qut.edu.au/%7Etaylort2/X80/.

[116] D. Thomas, Are our critical systems safe from cyber attack?, www.vnunet.com/computing/analysis, 2005.

[117] T. Tsuji, A. Shimizu, A one-time password authentication method, Master Thesis, Kochi University of Technology, 2003.

[118] U.S. Department of Energy, 21 Steps to Improve Cyber Security of SCADA Networks, 2002.

[119] X. Wang, H. Yu, How to Break MD5 and Other Hash Functions, Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 2005, ISBN 3-540-25910-4, 2005.

[120] X. Wang, Y.L. Yin, H. Yu, Collision search on SHA1, http://theory. csail.mit.edu/ ~yiqun/shanote.pdf, 2005.

[121] B. Waters, A. Juels, J. A. Halderman, E.W. Felten, New Client Puzzle Outsourcing Techniques for DoS Resistance, http://citeseer.ist.psu.edu/waters04new.html.

[122] Wikipedia, Information Security, http://en.wikipedia.org/wiki/Information_security.

[123] Wikipedia, Northeast Blackout of 2003, http://en.wikipedia.org/wiki/2003_North_America_blackout.

[124] Wikipedia, Parkerian Hexad, http://en.wikipedia.org/wiki/Parkerian_hexad.

[125] A.K. Wright, J.A. Kinast, J. McCarty, Low-Latency Cryptographic Protection for SCADA Communications, Proceedings of Applied Cryptography and Network Security, LNCS 3089, Springer, 2004.

[126] K. Zhang, Efficient Protocols for Signing Routing Messages, In Proceedings of Network and Distributed System Security Symposium, 1998.

[127] Y. Zhao, D. Li, An Improved Elegant Method to Re-initialize Hash Chains, http://eprint.iacr.org/2005/011.pdf, 2005.

Index

Anexe

A1. Rezultate obţinute pe parcursul stagiului doctoral

- **Lucrări publicate ca prim sau unic autor în temeiul tezei în publicaţii indexate**

- Groza, B., D. Petrica, *Cryptanalysis of an Authentication Protocol*, Proceedings of 7th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC'05, pp. 147-153, IEEE Comp. Soc., 2005.
- Groza, B., Petrica, D., Dragomir, T.L., *A time-memory trade to generate one-time passwords using quadratic residues over Zn*, Studies in Informatics and Control vol. 14 no. 3, 2005.
- Groza, B., *Using one-way chains to provide message authentication without shared secrets*, 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPeru'06, Lyon, France, pp. 82-87, IEEE Comp. Soc, 2006.
- Groza, B., D. Petrica, T.L. Dragomir, *Using the Discrete Squaring Function in the Delayed Message Authentication Protocol*, Proceedings of International Conference on Internet Surveillance and Protection, ICISP'06, Cap-Esterel, France, IEEE Comp. Soc., 2006.
- Groza, B., *An extension of the RSA trapdoor in a KEM/DEM Framework*, Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC'07, IEEE Comp. Soc., 2007.
- Groza, B., Dragomir, T.L., *On the use of one-way chain based authentication in secure control systems*,Second International Conference on Availability, Reliability and Security (ARES'07), International Workshop on Advances in Information Security (WAIS'07), Vienna, Austria, pp. 1214-1221, IEEE Comp. Soc., 2007.
- Groza, B., *Broadcast authentication protocol with time synchronization and quadratic residues chains*, Second International Conference on Availability, Reliability and Security (ARES'07), International Symposium on Frontiers in Availability, Reliability and Security (FARES'07), Vienna, Austria, pp. 550-557, IEEE Comp. Soc., 2007.
- Groza, B., Petrica, D., Barbu, S., Bilanin, M., *Implementation of an Authentication Protocol for Sending Audio-Video Information in Java*, SACI

2007, IEEE Comp. Int. Soc., 2007.

- Groza B., *On the use of the discrete power function for building public-key cryptosystems*, Applied Informatics & Communications, WSEAS Press, 2007, Greece, pp. 7-11 (Best Paper Award).
- Groza B., *Broadcast authentication with practically unbounded one-way chains*, JOURNAL OF SOFTWARE (JSW), Volume 3, Issue 2, 2008 (acceptata spre publicare in 2007 aparuta in Februarie 2008), ISSN : 1796-217X, Academy Publishers, Finlanda.
- Groza B., Murvay P.S., Silea I., Ionica T., *Cryptographic authentication on a 8051 based development board*, The Third International Conference on Internet Monitoring and Protection, ICIMP 2008, IEEE Comp. Soc.
- Groza B., Dragomir T.L., *Using a Cryptographic Authentication Protocol for the Secure Control of a Robot over TCP/IP*, IEEE-TTTC International Conference on Automation, Quality & Testing, Robotics, AQTR 2008 (THETA 16).
- Groza B., Pop D., Silea I., *Java Implementation of an Authentication Protocol with Application on Mobile Phones*, IEEE-TTTC International Conference on Automation, Quality & Testing, Robotics, AQTR 2008 (THETA 16).

➢ **Lucrări publicate ca prim sau unic autor în temeiul tezei în publicaţii neindexate**

- Groza, B., Petrică, D., *One time passwords for uncertain number of authentications,* CSCS-15, 15th International Conference on control systems and computer science, 25-27 May 2005, Politehnica University of Bucharest.
- Groza, B., Petrica, D., Dragomir, T.L., *Security based on cryptographic techniques for remote control systems*, SINTES, XII International Symposium on System Theory, Oct. 20-22, 2005, Craiova, Proceedings, Vol 4 Computer Engineering, ISBN 973-742-148-5, 973-742-154-X, pp. 729-734.
- Groza B., *Construction techniques for one-way chains and their use in authentication*, Control Engineering and Applied Informatics Journal, vol. 8, no. 1, 2006, ISSN 1454-8658, pp. 42-51.
- Groza B., *The Delayed Message Authentication Protocol with Chains Constructed on the Discrete Power Function*, 7th International Conference on Technical Informatics CONTI'2006, ISBN 973-625-319-X, pp. 33-36.
- Groza B., Petrica D., *On chained cryptographic puzzles*, 3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, SACI'2006, ISBN 963-7154-46-9, pp. 182-191.

- Groza B., Alexandroni A., Silea I., *An Overview of the NTLM Authentication and its Weaknesses in SharePoint Solutions*, The 8th International Conference on Technical Informatics, CONTI 2008.
- Groza B., Putanu E.A., Dragomir T.L. Petrica D., *Development of a Client-Server Platform for Simulation of Remote Control Systems from Matlab*, National Conference of Electrical Drives, CNAE 2008.

➢ **Granturi câştigate ca director în temeiul tezei finalizate prin rapoarte de cercetare**

- PROTOCOALE CRIPTOGRAFICE DE AUTENTIFICARE PRIN CODURI MAC CU CHEI INLANTUITE SI CU SINCRONIZARE TEMPORALA SAU CHALLENGE-RESPONSE SI PRIN SEMNATURI DIGITALE MULTIPLE-TIME SAU ONE-TIME IN ARBORI MERKLE / Grant Cercetare, MEdC-CNCSIS-TD-122/2007, nivel de finantare 10.773 RON, director Bogdan Ioan Groza, Universitatea Politehnica Timisoara, durata 1 an.
- PROTOCOALE DE SECURITATE SI TEHNICI CRIPTOGRAFICE BAZATE PE FUNCTII ONE-WAY PENTRU ASIGURAREA AUTENTICITATII INFORMATIEI / Grant Cercetare, MEdC-CNCSIS-TD-90/2006, nivel de finantare 14.490 RON, director Bogdan Ioan Groza, Universitatea Politehnica Timisoara, durata 1 an.

➢ **Cărţi publicate ca prim sau unic autor în temeiul tezei**

- Groza B., *Introducere in criptografia cu cheie publică*, 2007, 136 pagini, Editura Politehnica, ISBN 978-973-625-654-9.

➢ **Alte rezultate, activităţi**

- Recenzor invitat în toamna lui 2007, ca expert in protocoale de autentificare, în comitetul de organizare al workshopului IWSSI 2007, Innsburck, Austria, (http://www.comp.lancs.ac.uk/iwssi2007/) afiliat conferinţei de prestigiu în domeniu UBICOMP 2007.
- A urmat la Institul de cercetare B-IT din Bonn Germania, în septembrie 2007, cursuri de criptografie cu cheie publică ţinute de Pascal Paillier şi Gadiel Serrousi pe temele: "Reducţii de securitate pentru criptosisteme asimetrice" şi "Criptografie pe curbe eliptice".

- Best Paper Award pentru lucrarea: B. Groza, On the use of the discrete power function for building public-key cryptosystems, Applied Informatics & Communications, WSEAS Press, 2007, Greece, pp. 7-11.
- Membru în grantul CNCSIS: DEZVOLTAREA UNOR STRUCTURI AUTOMATE DE SPORIRE A DEPENDABILITATII SISTEMELOR DE CONDUCERE CU APLICATII IN SISTEMELE INDUSTRIALE (ENERGETICA, CHIMICA, AVIATIE, ROBOTICA), MEdC-CNCSIS-A-309/2005,2006, director Vânătoru Matei, Universitatea Craiova.
- Membru în diverse societăţi de specialitate naţionale şi internaţionale: IEEE (Institute of Electrical and Electronics Engineers), IACR (International Association for Cryptologic Research), SRAIT (Societatea Română de Automatică si Informatică Tehnică).