Capitolul 2

Servicii terţe de încredere (TTP)

2.1 Cerinţe ale serviciilor terţe de încredere

2.1.1 Noţiuni introductive

Schimbul de informaţie electronică între două entităţi implică prezenţa unui element de încredere care să asigure unele servicii, cum ar fi autenticitatea acelor entităţi. Mai mult, uneori nivelul de încredere aşteptat de entităţi nu se poate obţine decât cu ajutorul unei părţi terţe, care să faciliteze schimbul de informaţie.

Definiţia 2.1. *(X509) O entitate are încredere în alta când are siguranţa că partenerul se va comporta exact conform aşteptărilor sale.*

Un terţ de încredere (TTP - Trusted Third Party) este o entitate specifică care furnizează unul sau mai multe servicii electronice şi este considerată de încredere de către celelalte entităţi.

Exemplul 2.1. *Exemple de servicii furnizate prin intermediul terţilor de încredere: servicii de marcare temporală, servicii de arhivare electronică, servicii de administrare a cheilor şi certificatelor de chei, servicii suport pentru identificare, autentificare şi nonrepudiere, servicii de acordare de privilegii, servicii de notariat public electronic, servicii de directoare etc.*

Conform recomandărilor ISO 14516, un terţ de încredere trebuie:

- să ofere servicii clar definite;
- să respecte politici bine definite de utilizare şi de securitate, care sunt făcute public;
- să opereze într-un cadru perfect legal în ceea ce priveşte serviciile pe care le oferă şi entităţile cărora se adresează;
- să opereze în conformitate cu standardele naţionale şi internaţionale în vigoare;

- să fie independent şi imparţial pe timpul funcţionării;
- să urmărească cel mai bun cod de practici şi proceduri acceptat, pe care îl face public;
- să înregistreze şi să arhiveze toate elementele de evidenţă relevante pentru tipul de servicii oferite, astfel încât ulterior să se poată face auditarea;
- să permită arbitrarea independentă, fără a compromite securitatea serviciilor oferite sau a entităţilor sale client;
- să-şi asume responsabilitatea şi răspunderea în nişte limite definite pentru disponibilitatea şi calitatea serviciilor sale.

Un TTP poate furniza unul sau mai multe servicii. El trebuie însă să asigure faptul că serviciile pe care le oferă sunt în concordanţă cu politicile definite şi publicate de acesta. În funcţie de arhitectura furnizorului de servicii de încredere, pot exista cerinţe adiţionale şi de securitate care trebuie avute în vedere la administrarea şi operarea sa.

2.1.2 Cerinţe asupra terţilor de încredere

În general, cerinţele formulate asupra unui terţ de încredere diferă în funcţie de serviciile furnizate.

Pentru a câştiga încrederea părţilor interesate, un TTP trebuie să demonstreze că (ISO 14516):

- există şi aplică o politică de securitate potrivită;
- problemele de securitate sunt rezolvate printr-o combinaţie de mecanisme şi proceduri de securitate corect implementate;
- operaţiile sunt realizate corect şi în strânsă legătură cu un set de roluri şi responsabilităţi clar definite;
- procedurile şi interfeţele de comunicare cu entităţile sunt potrivite cu scopul lor şi sunt aplicate corect;
- regulile şi regulamentele sunt corect aplicate şi sunt consistente cu nivelul de încredere dorit;
- calităţile operaţiilor şi practicilor de lucru au fost corect acreditate;
- respectă obligaţiile contractuale în raport cu utilizatorii săi;
- există o înţelegere şi acceptare clară a responsabilităţilor;

- compatibilitatea cu legile şi regulamentele este menţinută şi auditată;
- ameninţările cunoscute şi măsurile de contracarare a acestora sunt clar identificate;
- sunt îndeplinite cerinţele organizaţionale şi de personal;
- credibilitatea sa poate fi verificată;
- este monitorizat permanent de o autoritate administrativă care îi supervizează activitatea.

Cerinţe funcţionale

D. Lekkas ş.a. au realizat ([11]) un studiu privind o serie de proiecte de securitate derulate la nivel european, toate bazate pe utilizarea TTP-urilor. Aici s-au identificat o serie de cerinţe funcţionale care pot fi definite la nivelul unui TTP:

1. **Autentificarea**: Identificarea corectă a entităţilor implicate în tranzacţiile electronice.
   Se obţine în general utilizând mecanisme de criptografie cu chei publice şi semnătură electronică. Stocarea cheilor private de autentificare se face de obicei: pentru utilizatori pe smartcarduri dedicate, iar pentru serviciile furnizate de TTP-uri pe dispozitive speciale de tip HSM (*Hardware Secure Module*).

2. **Integritatea datelor**: Păstrarea lor nealterată pe timpul comunicării între entităţi. Alterarea datelor poate fi accidentală sau intenţionată. Integritatea se poate realiza utilizând mecanisme de semnătură electronică sau funcţii de dispersie.

3. **Confidenţialitatea**: Criptarea mesajelor schimbate între entităţi în cadrul tranzacţiilor electronice constituie o cerinţă de bază. Se obţine utilizând în general algoritmi de criptare cu chei simetrice, iar în unele situaţii – mecanisme de criptare asimetrică.

4. **Non-repudierea**: O entitate nu poate nega o acţiune (cum ar fi expedierea sau recepţionarea unui mesaj) sau existenţa unor informaţii la un moment dat.
   Această cerinţă poate fi asigurată cu tehnici de semnătură digitală (non-repudierea originii mesajelor) sau de marcare temporală (existenţa datelor la un anumit moment).

5. **Disponibilitatea**: Este de asemenea una din cerinţele fundamentale ale unui TTP. Ea este corelată cu politica TTP-ului şi SLA-ului (*Service Level Agreament*) acceptat.
   Disponibilitatea poate fi asigurată prin mecanisme specifice de HA (*High - Availability*) având la bază redundanţa echipamentelor şi mecanisme de DR (*Disaster - Recovery*).

6. **Uşurinţa de utilizare**: Interfaţa sistemului cu utilizatorii constituie o cerinţă importantă în condiţiile în care interacţionează în mod direct cu aceştia. Unele servicii oferite de TTP nu interacţionează în mod direct cu utilizatorii, ci prin intermediul unor aplicaţii care implementează protocoale specifice.

7. **Mobilitatea**: Necesară – în unele situaţii – pentru utilizatorii mobili. Aceştia trebuie să poată contacta un anumit TTP indiferent de localizarea lor în raport cu acesta.

8. **Anonimitatea**: O entitate poate fi înregistrată la un TTP, însă (în funcţie de opţiunile sale) identitatea sa trebuie să nu fie dezvăluită celorlaltor utilizatori.

9. **Marcarea temporală**: Mărcile temporale sigure (de încredere) ataşate documentelor electronice sunt necesare în anumite tranzacţii desfăşurate între entităţi.
   În general serviciile de marcare temporală (TSP - *Time Stamp Providers*) sunt văzute ca servicii auxiliare ale serviciilor de securitate. Implementarea lor necesită însă un nivel mare de atenţie, datorită complexităţii şi cerinţelor speciale privind echipamentele hardware de sincronizare a timpului.

10. **Unicitatea**: Unicitatea unor documente electronice/mesaje poate fi o cerinţă funcţională care trebuie îndeplinită de un TTP.

11. **Inter-operabilitatea**: Schimbul mesajelor nu poate fi restricţionat la domeniul deservit de un singur TTP. În unele situaţii, mesajele electronice trebuie procesate însă de utilizatori din domenii deservite de TTP-uri diferite.
    Cerinţa poate fi asigurată prin acţiuni suplimentare executate între TTP-uri şi sunt specifice fiecărui tip de TTP. De exemplu, cross - certificarea a două CA - uri din domenii PKI diferite poate fi o condiţie necesară şi suficientă pentru asigurarea inter-operabilităţii utilizatorilor acelor domenii.
    Tot inter-operabilitatea presupune şi respectarea standardelor care guvernează domeniul de aplicabilitate.

12. **Acreditarea**: Procedurile de auditare si acreditare pentru TTP-uri sunt esenţiale în special pentru asigurarea unui nivel de încredere cerut în relaţiile cu utilizatorii. Acreditarea se poate face la nivel local, naţional sau internaţional iar nivelul ei depinde de legile şi standardele existente în domeniul respectiv.

13. **Politica de securitate**: Fiecare TTP trebuie să ofere utilizatorilor săi o politică de securitate bine definită, în concordanţă cu legislaţia şi restricţiile naţionale precum şi cu cerinţele de securitate definite.
    Disputele dintre entităţi vor fi arbitrate în concordanţă şi cu politicile de securitate definite la nivelul TTP-urilor implicate (direct sau indirect) în tranzacţiile electronice.

14. **Managementul cheilor**: Utilizatorii pot cere unui TTP diverse servicii de gestiune privind cheile lor de semnătură şi criptare. Generarea cheilor, distribuţia acestor chei, mecanisme de recuperare de chei (key - recovery), backup pentru chei, key - escrow, reînnoirea automată sau la cerere a cheilor (la expirare sau în caz de compromitere) pot fi cerinţe funcţionale la nivelul unui TTP.

15. **Publicarea datelor**: Serviciile de publicare din TTP-uri sunt necesare pentru pu-blicarea unor informaţii folosite de utilizatorii sistemelor. Distribuirea cheilor publice sau a certificatelor digitale pentru utilizatori, distribuirea listelor de certificate revocate sunt informaţii esenţiale proceselor de certificare şi validare.
Serviciile de publicare pot fi implementate prin intermediul unor protocoale cum ar fi $LDAP$, $HTTP$, $X.500$, DNS etc.

16. **Scalabilitatea şi modularitatea**: Serviciile furnizate de TTP-uri trebuie să fie scalabile şi uşor gestionabile în implementări pe scară largă. Structura modularizată permite adăugarea sau scoaterea cu uşurinţă a unor componente de funcţionalitate la nivelul TTP-ului.

17. **Compatibilitatea şi portabilitatea**: Presupune compatibilitatea implementărilor TTP-urilor cu standardele, tehnologiile şi platformele software/hardware cerute.

Cerinţe privind politica de securitate

Politica de securitate a unui TTP trebuie să acopere toate aspectele de securitate privind administrarea TTPului şi operarea serviciilor sale, fiind un instrument vital pentru generarea încrederii către entităţile client.

O politică de securitate înglobează toate elementele principale de funcţionare ale terţului de încredere. Orice politică de securitate ar trebui să conţină:

1. O componentă privind politica de securitate generală, în care care sunt stabilite toate aspectele non-tehnice privind securitatea şi încrederea în serviciile terţului;

2. O componentă privind politica de securitate tehnică, în care sunt stabilite aspectele tehnice privind securitatea şi funcţionalitatea terţului. Aici sunt descrise rutinele, procedurile şi aspectele tehnice ale sistemului.

Conţinutul efectiv al politicii depinde de serviciile oferite de TTP. În orice politică de securitate sunt incluse următoarele elemente:

- conceptele generale privind serviciile furnizate;
- definirea entităţilor participante;
- cerinţele de securitate (confidenţialitatea, integritatea, disponibilitatea, autenticitatea, responsabilitatea);

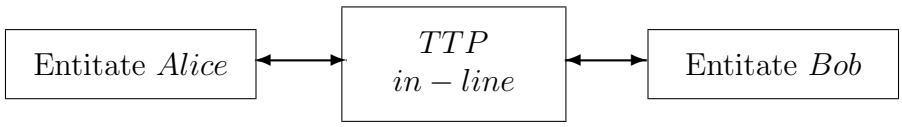
- infrastructura organizaţională si asignarea responsabilităţilor în cadrul TTP-ului;
- obligaţiile si răspunderile TTP-ului şi a celorlaltor entităţi implicate;
- ciclul de viaţă al cheilor criptografice necesare (mecanismul de generare a cheilor, protejarea cheilor, timpul lor de viaţă, reînnoirea cheilor, distrugerea cheilor etc.);
- mecanismele de asigurare a securităţii în cadrul sistemului;
- procedurile de operare şi scenariile lor de aplicare;
- definirea rolurilor necesare în cadrul operării serviciilor furnizate;
- definirea claselor pentru clasificarea informaţiilor;
- aspectele legate de personal, în special pentru personalul din funcţii - cheie;
- strategiile de gestiune a riscurilor;
- tratarea incidentelor.

2.2 Configuraţii cu terţi de încredere

Din punct de vedere al poziţionării terţului de încredere în cadrul comunicaţiei dintre entităţile client, precum şi al implicării sale directe sau indirecte la protocolul de comunicaţie, se pot realiza trei configuraţii diferite:

1. *Configuraţie cu TTP in-line*:

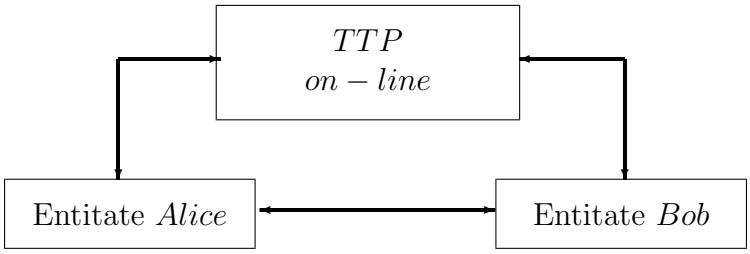
   Un terţ de încredere in-line poate exista atunci când două entităţi aparţinând la două domenii de securitate diferite doresc să schimbe între ele mesaje securizate. În această situaţie, terţul este poziţionat între cele două entităţi, acţionând ca un intermediar în toate interacţiunile dintre ele şi facilitând schimburile lor securizate de informaţii.



   Configuraţiile cu TTP-uri in-line pot include servicii cum ar fi autentificarea sau controlul privilegiilor (terţii de încredere jucând un rol în furnizarea de servicii de non-repudiere), controlul accesului, recuperarea de chei, confidenţialitate şi integritate pentru datele transmise.

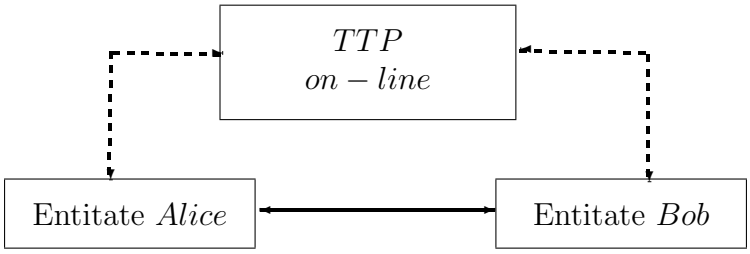
2. *Configuraţie cu TTP on-line*: Un TTP on-line se poate utiliza în situaţia când cel puţin una din entităţi cere terţului respectiv furnizarea sau înregistrarea unor informaţii de securitate, iar terţul este implicat doar în câteva din schimburile securizate dintre ele (de obicei în prima fază a comunicaţiei).

   Ulterior TTP nu mai participă la tranzacţiile dintre entităţi şi nu este poziţionat pe calea lor de comunicaţie.



   Astfel de configuraţii pot include servicii de autentificare, certificare, controlul privilegiilor etc.
   Terţul de încredere poate juca de asemenea un rol în furnizarea de servicii de nonrepudiere, controlul accesului, managementul cheilor, marcare temporală, confidenţialitate şi integritate pentru datele transmise.

3. *Configuraţie cu TTP off-line*: Terţul de încredere off-line nu interacţionează direct cu entităţile în timpul schimburilor de mesaje securizate. În schimb, entităţile comunică între ele folosind date generate anterior de către TTP.



   TTP-urile off-line pot include servicii de autentificare, certificare, controlul privilegiilor, non-repudiere, distribuirea cheilor, recuperarea cheilor etc.

2.3 Inter-operarea serviciilor TTP

Un TTP poate avea acorduri de încredere stabilite cu alte TTP-uri pentru a forma o reţea care să permită entităţilor sale să comunice securizat cu entităţile acestea. În situaţiile în care TTP-ul nu poate oferi anumite servicii solicitate, se pot stabili protocoale de

încredere care să permită altor TTP-uri să subcontracteze şi să asigure aceste servicii suplimentare.

La analizarea cerinţelor de interconectare trebuie avut în vedere dacă relaţia legală dintre un TTP şi abonaţii săi este diferită de cea dintre acel TTP şi alte entităţi interesate (de exemplu utilizatori care verifică semnături digitale bazate pe certificatele digitale emise de o Autoritate de Certificare).
Fiecare terţ de încredere furnizează servicii către entităţile din domeniul său conform cu politica de securitate proprie.

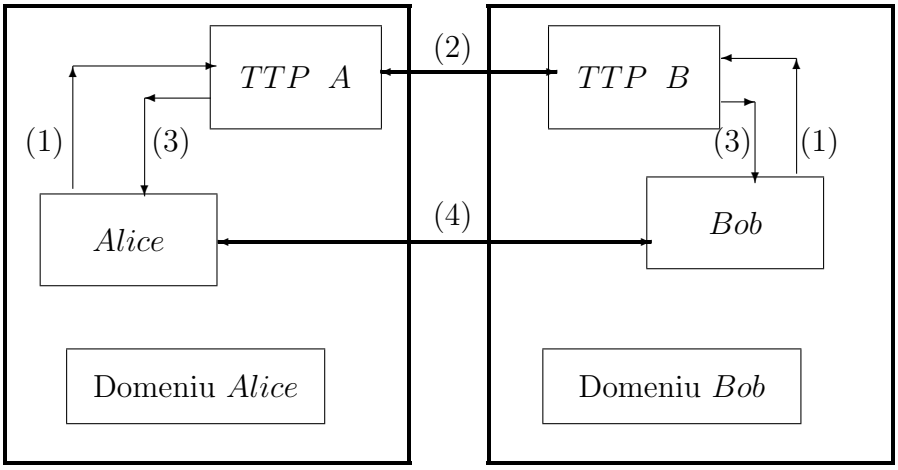
Există următoarele posibilităţi de interconectare:

1. Interconectare TTP - *Utilizator*: presupune mecanisme şi mijloace prin care un utilizator interacţionează cu un TTP pentru a cere/primi un serviciu. Fiecare utilizator poate interacţiona cu un TTP în mod diferit, în funcţie de serviciul solicitat.

2. Interconectare *Utilizator* - *Utilizator*: după ce un TTP şi-a terminat joburile în relaţia cu entităţile sale, toate comunicaţiile dintre entităţi se pot face fără a mai fi nevoie de asistenţa TTP-ului.

   Relaţia între entităţi, precum şi formalizarea contractuală a acestei relaţii, se bazează pe încrederea acestora în TTP şi pe mecanismele de interconectare dintre TTP-uri.

3. Interconectare TTP - TTP:
   Figura de mai jos prezintă un exemplu de astfel de interfaţă (ISO 14516):



   (a) *Alice* cere de la TTP-ul A o cheie secretă pentru a comunica cu *Bob* (1).

   (b) TTP-ul A transferă cheia secretă către *Alice* (3) şi către TTP-ul B (2).

   (c) TTP-ul B transmite apoi cheia către *Bob* (3).

   (d) Având stabilită o cheie comună, cele două entităţi pot comunica securizat (4).

O variantă a acestui protocol, care utilizează tehnologia cu chei publice:

(a) *Alice* cere TTP-ului A un certificat pentru a comunica securizat cu *Bob* (1).

(b) TTP-ul A emite un certificat lui *Alice* (3).

(c) Analog, *Bob* cere către TTP-ul B emiterea unui certificat (1) şi-l obţine (3).

(d) De asemenea, cele două TTP-uri îşi emit una către cealaltă câte un certificat (2).

(e) Acum, cele două entităţi pot comunica securizat având stabilită între ele infrastructura de chei de încredere publice (4).

2.4 Servicii de Non-Repudiere

Repudierea este una dintre problemele de securitate fundamentale existente în tranzacţiile efectuate prin documente. Necesitatea serviciilor de non-repudiere nu provine numai din faptul că părţile pot încerca să se înşele una pe cealaltă. Este o realitate bine cunoscută faptul că diverse circumstanţe neaşteptate pot conduce la situaţia în care două entităţi implicate într-o tranzacţie ajung – în timp – să aibă puncte diferite de vedere cu privire la ce s-a întâmplat în cadrul relaţiei dintre ele. O eroare de comunicaţie pe reţea în timpul derulării unui protocol este un exemplu reprezentativ.

Putem defini o tranzacţie de bază ca fiind transferarea unui mesaj M de la *Alice* (Emitent) la *Bob* (Receptor):

$$A \longrightarrow B \ : \ M$$

Chiar şi într-o astfel de tranzacţie simplă, ar putea apare următoarele cazuri de dispută:

- *Alice* susţine că a trimis mesajul M lui *Bob*, iar *Bob* acuză faptul că nu l-a primit;
- *Bob* susţine că a primit mesajul M de la *Alice*, iar *Alice* acuză faptul că nu l-a trimis;
- *Alice* susţine că a trimis mesajul M înainte de un moment de timp T, în timp ce *Bob* declară că că nu a primit mesajul înainte de momentul T.

Serviciile de non-repudiere ajută entităţile implicate într-o astfel de tranzacţie să rezolve posibilele dispute care pot apare cu privire la anumite evenimente sau acţiuni care s-au întâmplat (sau nu s-au întâmplat) în cadrul tranzacţiei.

Definiţia 2.2. *Un protocol de non-repudiere este un flux de tranzacţii în care entităţile implicate schimbă dovezi electronice, capabile să ofere apoi servicii de non-repudiere.*

Principalele dovezi de non-repudiere – prezente în toate protocoalele propuse – sunt:

- *Non-Repudierea Originii*: un protocol de non-repudiere oferă non-repudierea originii (NRO), dacă şi numai dacă poate genera o dovadă privind originea mesajului – destinată receptorului mesajului – şi care prezentată unui judecător, acesta poate stabili fără dubiu că iniţiatorul a trimis acel mesaj.
- *Non-Repudierea Recepţiei*: un protocol de non-repudiere oferă non-repudierea recepţiei (NRR), dacă şi numai dacă poate genera o dovadă privind primirea mesajului – destinată iniţiatorului mesajului – şi care prezentată unui judecător, acesta poate stabili fără dubiu că destinatarul a primit acel mesaj.

Un protocol de non-repudiere este corect (fair) dacă la finalizarea sa:

- *Alice* deţine o dovadă de tip NRR, iar *Bob* deţine o dovadă de tip NRO; sau
- nici unul dintre ei nu deţine o informaţie de acest tip.

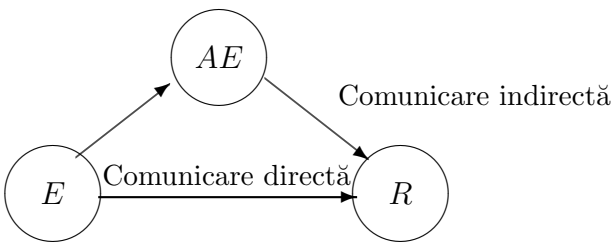
Protocoalele de non-repudiere corecte au constituit o temă importantă de cercetare în literatura de specialitate. Studiul [15] împarte protocoalele de non-repudiere în:

1. protocoale cu corectitudine tare;
2. protocoale cu corectitudine adevărată;
3. protocoale cu corectitudine probabilistică.

Cele mai multe protocoale de non-repudiere propuse iau în considerare cazul în care sunt implicate doar două entităţi, care trebuie să obţină dovezile de non-repudiere a originii, respectiv a recepţiei unui mesaj. Există şi scheme de protocoale care asigură dovezile necesare de non-repudiere pentru scenarii cu mai mult de două entităţi care doresc să schimbe mesaje între ele. Un studiu asupra acestui tip de protocoale este realizat în [25].

Într-o tranzacţie electronică, un transfer de mesaj poate fi transferat de la un emitent E (*Alice*) către un receptor R (*Bob*) în două feluri:

1. E trimite mesajul direct către R (comunicare directă); sau
2. E trimite mesajul către un *Agent de Expediere* intermediar AE, care apoi trimite mesajul către R (comunicare indirectă).



În varianta de comunicare directă, dacă *Emitentul* şi *Receptorul* nu au încredere unul în celălalt, pentru a se putea contoriza corect acţiunile lor, sunt necesare următoarele servicii de non-repudiere (ISO 13888 − 3):

- *Non-Repudierea Originii* (NRO): asigură protecţia împotriva refuzului *Emitentului* de a recunoaşte transmiterea mesajului. Dovada transmiterii mesajului va fi generată de *Emitent* sau un TTP şi va fi păstrată de *Receptor*.

- *Non-repudierea Recepţiei* (NRR): asigură protecţia împotriva refuzului *Receptorului* de a recunoaşte primirea mesajului. Dovada primirii mesajului este generată de *Receptor* sau de un TTP şi va fi păstrată de *Receptor*.

În varianta de comunicare intermediată de un *Agent de Expediere* (AE), pentru a se putea rezolva eventualele dispute între *Emitent* şi *Agentul de Expediere*, respectiv între *Agentul de Expediere* şi *Receptor*, sunt necesare următoarele servicii de non-repudiere ISO 13888 − 3):

- *Non-repudierea depunerii* (NRS): asigură dovada că *Emitentul* a depus mesajul pentru expediere. Dovada depunerii mesajului este generată de *Agentul de Expediere* şi va fi păstrată de *Receptor*.

- *Non-repudierea expedierii* (NRD): asigură dovada că mesajul a fost expediat de *Receptor*. Dovada de expediere este generată de *Agentul de Expediere* şi va fi păstrată de *Emitent*.

În general, protocoalele care stau la baza unui serviciu de non-repudiere trebuie să îndeplinească următoarele cerinţe:

- *Corectitudine*: Nici una din cele două entităţi implicate nu trebuie să poată deţine vreun avantaj faţă de cealaltă privind obţinerea dovezilor de non-repudiere.
  Repudierea poate fi prevenită numai dacă fiecare entitate obţine în egală măsură informaţia de care are nevoie: *Bob* trebuie să obţină mesajul util şi dovada de non-repudiere a originii acestui mesaj, iar *Alice* trebuie să obţină dovada de non-repudiere a recepţiei mesajului transmis.
  În caz contrar, cei doi utilizatori nu trebuie să aibă acces la nici una din aceste informaţii.

- *Eficienţă*: Non-repudierea se obţine în general prin folosirea unor servicii de tip TTP. Gradul de implicare al acestora este esenţial în determinarea eficienţei unui protocol de non-repudiere.

- *Oportunitate*: Din diverse motive, o tranzacţie din protocolul de non-repudiere poate fi întârziată sau chiar stopată intenţionat de una dintre părţile implicate. Acest lucru nu trebuie să dezavantajeze cealaltă parte.

- *Politică*: Toate regulile şi toţi parametrii necesari în cadrul serviciului de non-repudiere trebuie definite corect şi complet.
- *Transparenţa TTP-ului*: În anumite situaţii este de dorit ca implicarea TTP-ului în cadrul protocolului (sau a rezolvării disputelor) să fie invizibilă. Astfel, dovezile obţinute prin implicarea TTP-ului vor fi similare celor obţinute fără ajutorul acestuia.
- *Verificabilitatea TTP-ului*: Proprietate necesară în situaţia când TTP-ul nu este considerat de încredere de ambele entităţi.

2.4.1 Protocoale de non-repudiere bazate pe TTP-uri

Există mai multe abordări şi propuneri privind protocoale care să asigure cerinţele definite mai sus pentru un serviciu de non-repudiere. Unele presupun obţinerea dovezilor de non-repudiere fără a implica terţe părţi în comunicarea dintre *Alice* şi *Bob*; cele mai multe protocoale se bazează însă pe conectarea la un TTP.

Rolul TTP-urilor în protocoalele de non-repudiere

În funcţie de mecanismele de non-repudiere folosite şi de politica aplicată, TTP-urile pot participa sub diverse roluri la generarea, verificarea, validarea sau transferarea dovezilor de non-repudiere şi la arbitrarea disputelor.

Din punct de vedere al implicării TTP-ului în cadrul derulării unui protocol de non-repudiere, pot fi identificate trei situaţii (ISO $10181-4$):

1. *Protocoale de non-repudiere bazate pe TTP-uri in-line.* Acest tip de terţi de încredere acţionează ca intermediari în toate tranzacţiile efectuate între entităţi. Toate mesajele schimbate în cadrul protocolului trec pe la TTP.
2. *Protocoale de non-repudiere bazate pe TTP-uri on-line.* TTP-urile participă activ la generarea şi verificarea dovezilor de non-repudiere.
3. *Protocoale de non-repudiere bazate pe TTP-uri off-line.* În această situaţie, terţii de încredere nu participă activ în cadrul serviciului de non-repudiere (vor fi invocaţi doar în anumite situaţii de excepţie).

Un TTP implicat în gestionarea dovezilor de non-repudiere se poate afla în una din situaţiile de mai jos ([30]):

- *Ca Autoritate de Certificare.* Un CA generează certificate digitale pentru cheile utilizatorilor, autentificându-le astfel pentru a fi folosite în protocoalele de non-repudiere. CA-ul furnizează de asemenea listele de certificate revocate, pentru a putea fi determinată validitatea cheilor folosite.

Autorităţile de Certificare sunt folosite întotdeauna în situaţiile când pentru generarea dovezilor de non-repudiere sunt utilizate semnăturile digitale. În general, TTP-urile cu rol de CA sunt utilizate off-line în protocoalele de non-repudiere. Ele pot fi folosite şi ca TTP-uri on-line, dacă semnăturile electronice folosesc formate avansate de semnătură electronică şi conţin informaţii bazate pe servicii on-line de validare a certificatelor.

- *Ca Notar Electronic.* Similar cazului non-electronic, un TTP cu rol de notar electronic poate fi utilizat pentru asigurarea unor servicii de non-repudiere.

  Dacă pentru generarea dovezilor de non-repudiere sunt folosite mecanisme bazate pe criptografia simetrică, TTP-ul implicat în protocol poate fi activat pentru a genera dovezile de non-repudiere în numele participanţilor. Dacă dovezile de non-repudiere se obţin pe bază de semnături digitale, notarul ar trebui să furnizeze mărci temporale privind momentul generării dovezilor.

  În general TTP-urile cu rol de Notar Electronic sunt utilizate în protocoalele de non-repudiere într-o arhitectură on-line.

- *Ca Autoritatea de Expediere.* O astfel de Autoritate constituie un terţ de încredere în ceea ce priveşte expedierea mesajelor.

  Acest tip de TTP-uri este utilizat în cadrul protocoalelor de non-repudiere într-o arhitectură in-line.

- *Ca Autoritatea de Arbitrare.* Un TTP cu acest rol nu va fi implicat în protocoale decât în situaţiile în care apar dispute, principalul său scop fiind judecarea şi rezolvarea acestora. Judecarea disputelor presupune evaluarea dovezilor puse la dispoziţie de participanţi şi luarea în consideraţie a unei politici de non-repudiere.

Protocoale de non-repudiere fără TTP-uri

Protocoalele de acest tip pot asigura doar probabilistic cerinţele de non-repudiere.

Chiar dacă în implementare se aleg corect parametrii protocolului, gradul de risc este foarte mic iar probabilitatea de nesoluţionare a disputelor este neglijabilă, totuşi aceste protocoale sunt ineficiente, fiind greu de aplicat.

Exemplul 2.2. *Primele protocoale fără TTP apar la jumătatea anilor* 80*, dezvoltate iniţial pentru schimbul de secrete (de exemplu chei criptografice) între entităţi. Ideea de bază era ca fiecare entitate să transmită pe rând biţi succesivi din informaţia secretă care trebuia furnizată celeilalte entităţi, până la epuizarea informaţiei. Dacă o entitate stopa procesul, cealaltă proceda similar; în acest mod, efortul de calcul pentru aflarea restului de informaţie era sensibil echivalent pentru ambele entităţi. În cadrul acestor protocoale de schimb de date pot fi identificate mecanisme clare de non-repudiere.*

Primele protocoale pure de non-repudiere propuse au fost cele bazate pe TTP-uri. Abia mai târziu au apărut şi protocoale de non-repudiere care nu necesitau existenţa unui TTP.

Pentru exemplificare prezentăm protocolul probabilist de non-repudiere propus de Markowitch şi Roggeman ([18]). Protocoale similare care asigură non-repudierea şi nu folosesc TTP-uri au mai fost realizate în [13], [26], [22].

Scopul protocolului Markowitch - Roggeman este de a obţine dovezile de non-repudiere NRO şi NRR fără a utiliza un terţ de încredere. Protocolul îsi propune să asigure transmiterea unui mesaj M de la *Alice* către *Bob* şi să asigure dovezile de non-repudiere a originii şi recepţiei mesajului.

Protocolul este descris de următoarele tranzacţii:

1. În prima rundă, *Alice*:
   (a) Criptează mesajul M folosind cheia simetrică k şi obţine $c = E_k(M)$;
   (b) Generează dovada originii lui c:
   $$EOO_c = Sig_{Alice}(Bob, l, c)$$
   unde $l = Hash(M, k)$;
   (c) Trimite lui *Bob* (EOO_C, l, c).
2. *Bob* răspunde cu dovada recepţiei mesajului criptat c:
   $$EOR_C = Sig_{Bob}(Alice, l, c)$$
3. În rundele $i = 2, \ldots, n-1$
   (a) *Alice* trimite lui *Bob* cuplul $(EOO_{r_{i-1}}, r_{i-1})$ unde r_{i-1} este o valoare aleatoare de rundă, iar
   $$EOO_{r_{i-1}} = Sig_{Alice}(Bob, i, r_{i-1})$$
   (b) *Bob* răspunde cu dovada recepţiei valorii r_{i-1}:
   $$EOR_{r_i} = Sig_{Bob}(Alice, i, r_i)$$
4. În ultima rundă, *Alice* trimite cheia simetrică k şi dovada $EOO_{r_{n-1}}$ a originii acesteia:
   $$EOO_{r_{n-1}} = Sig_{Alice}(Bob, n, k)$$
5. *Bob* răspunde cu dovada recepţiei cheii k:
   $$EOR_{r_{n-1}} = Sig_{Bob}(Alice, n, k)$$

După epuizarea rundei n şi primirea lui $EOR_{r_{n-1}}$, *Alice* anunţă terminarea protocolului, dezvăluind practic numărul de runde ales şi faptul că în această ultimă rundă a trimis cheia simetrică k prin care *Bob* poate obţine mesajul $M = D_k((c)$. Dovada de non-repudiere a originii va fi:

$$NRO = (EOO_c, EOO_{r_{n-1}}),$$

iar dovada de non-repudiere a recepţiei va fi $NRR = (EOR_c, EOR_{r_{n-1}})$.

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c: dovedeşte că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;
  - $EOO_{r_{n-1}}$: dovedeşte că *Alice* i-a trimis şi cheia simetrică de decriptare.
- *Alice* poate dovedi non-repudierea recepţiei:
  - EOR_c: dovedeşte că *Bob* i-a confirmat primirea mesajului criptat cu o cheie simetrică;
  - $EOR_{r_{n-1}}$: dovedeşte că *Bob* i-a confirmat şi primirea cheii simetrice de decriptare, adică obţinerea mesajului M.

Protocolul se bazează pe păstrarea secretului privind numărul de runde şi cheia secretă k, alese de *Alice* până la epuizarea întregului set de runde. *Bob*, neştiind dinainte numărul de runde fixat de *Alice*, în ultima rundă nu va şti că a primit cheia simetrică decât după ce va fi anunţat de *Alice*, furnizând la rândul lui informaţia necesară de non-repudiere.

Observaţia 2.1.

- *La fiecare din ultimele $n-1$ runde, înainte de a trimite dovada non-repudierii recepţiei corespunzătoare, Bob ar putea încerca decriptarea mesajului primit în prima rundă; dacă reuşeşte, poate alege să nu mai trimită această dovadă, ceea ce l-ar pune în avantaj faţă de Alice. Această problemă se poate rezolva printr-o implementare care să contorizeze timpii de răspuns ai lui Bob. Astfel, dacă apare o întrziere mai mare în cadrul unui răspuns de la Bob, protocolul eşuează. Pentru asta, trebuie ales un algoritm de criptare în care operaţia de decriptare să dureze un timp suficient ca să poată fi detectat de Alice. Soluţia este ineficientă practic din mai multe motive (stadiul actual face ca timpii de calcul să fie insezizabili; sau, dacă entităţile utilizează o reţea de comunicaţie instabilă unde pot apare întârzieri nepredictibile, fără ca ele să fie induse de Bob).*
- *Pentru ca protocolul să fie funcţional, valorile aleatoare $r_1, \ldots, r_{n-1}$ trebuie să fie independente, egal distribuite, de lungimi aproximativ echivalente cu lungimea lui k.*

- *Alice trebuie să aleagă un algoritm de criptare simetric şi/sau un mod de utilizare al acestui algoritm care să nu permită decriptarea doar a unei părţi a mesajului. În [18] sunt propuse astfel de moduri şi mecanisme de prevenire.*

- *Oricare dintre entităţi poate întrerupe protocolul în rundele intermediare; în această situaţie, nici Alice şi nici Bob nu deţine un avantaj, deoarece nu va avea dovada privind trimiterea/recepţia cheii simetrice de decriptare.*

  Există însă o probabilitate ca Bob să intuiască corect numărul de ordine al rundei finale şi să întrerupă protocolul înainte de epuizarea corectă a acestei runde finale, ceea ce îi conferă un avantaj. În această situaţie Bob va avea dovada completă a non-repudierii originii în timp ce lui Alice îi lipseşte dovada recepţiei de către Bob a cheii de decriptare, deci nu poate dovedi că Bob a avut acces la mesaj.

Protocoale de non-repudiere bazate pe TTP-uri in-line

În 1996 Coffey si Saidha au propus un protocol de non-repudiere ([9]) bazat pe un TTP in-line utilizat pe post de *Server de Non-Repudiere* (*NRS - Non-Repudiation Server*). Acest terţ colectează dovezile de non-repudiere şi le transmite apoi entităţilor care îşi dispută tranzacţia. Practic terţul de încredere funcţionează ca un *Agent de Expediere* (AE) pentru mesajele schimbate între entităţile participante.

Protocolul utilizează semnăturile digitale (ca mecanism criptografic pentru generarea dovezii de non-repudiabilitate) şi criptografia cu chei publice (pentru schimbul de mesaje).

Pentru ca protocolul să fie funcţional, Serverul de Non-Repudiere trebuie să aibă următoarele caracteristici:

- este independent de cele două entităţi, fiind însă de încredere pentru acestea;

- nu distribuie nici o informaţie legată de dovezi către vreo entitate participantă până când nu deţine şi dovada corespunzătoare pentru a putea fi distribuită către cealaltă entitate;

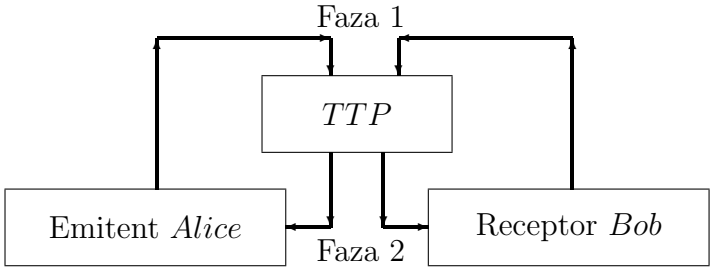
- primeşte dovada de non-repudiere a originii direct de la *Alice* şi cooperează cu *Bob* pentru a genera dovada de non-repudiere a recepţiei;

- odată ce deţine toată informaţia necesară de non-repudiere pentru ambele entităţi, nu se va opune procesului de distribuţie a acestei informaţii către cele două entităţi.

Protocolul Coffrey - Saidha se desfăsoară în două faze:

1. (Faza 1) Generarea dovezilor de non-repudiabilitate; are loc la TTP.

2. (Faza 2) Distribuirea dovezilor de non-repudiabilitate.



După ce se consumă *Faza 1* şi Serverul de Non-Repudiere este în posesia celor două dovezi, el va distribui în *Faza 2* aceste dovezi către părţile comunicante.

Toată comunicarea dintre *Alice* şi *Bob* are avea loc prin intermediul Serverului de Non-Repudiere, iar mesajele schimbate sunt semnate digital şi însoţite de o informaţie privind momentul semnării. De aceea – pe lângă Serverul de Non-Repudiere – protocolul necesită şi prezenţa unei *Autorităţi de Marcare Temporală* care să marcheze momentul semnării şi să poată astfel demonstra că semnarea mesajelor s-a făcut în perioada de valabilitate a certificatelor digitale corespunzătoare cheilor de semnătură.

Protocolul Coffey-Saitha cu TTP in-line este descris formal de următoarele tranzacţii:

1.	$Alice \longrightarrow TSA:$	$PK_{TSA}(NROp)$
2.	$TSA \longrightarrow Alice:$	$PK_{Alice}(NRO)$
3.	$Alice \longrightarrow NRS:$	$"NR-req"$
4.	$NRS \longrightarrow Alice:$	$PK_{Alice}(n_1)$
5.	$Alice \longrightarrow NRS:$	$PK_{NRS}(Sig_{Alice}(n_1, NRO, NRRp))$
6.	$NRS \longrightarrow Bob:$	$PK_{Bob}(Sig_{NRS}(n_2, NRRp))$
7.	$Bob \longrightarrow TSA:$	$PK_{TSA}(NRRps)$
8.	$TSA \longrightarrow Bob:$	$PK_{Bob}(NRR)$
9.	$Bob \longrightarrow NRS:$	$PK_{NRS}(Sig_{Bob}(n_2, NRR))$
10.	$NRS \longrightarrow Bob:$	$PK_{Bob}((NRO))$
11.	$NRS \longrightarrow Alice:$	$PK_{Alice}(NRR)$

Descrierea şi analiza protocolului:

- Toate mesajele intermediare M_i sunt criptate cu cheia publică a entităţii destinaţie: $PK_{Bob}(M_i)$. Astfel numai *Bob* – în calitate de destinatar – poate accesa mesajul transmis, utilizând pentru decriptare, cheia sa privată.

- În paşii 1 şi 2, *Alice* – ca entitate iniţiatoare – generează prin intermediul terţului de încredere dat de Autoritatea de Marcă Temporală (TSA – *Timestamp Authority*) dovada de non-repudiere a originii (NRO).

  Alice trimite către TSA dovada parţială a originii:

$$NROp = Sig_{Alice}(Alice, Bob, M)$$

şi primeşte de la TSA marca temporală peste această semnatură:

$$NRO = Sig_{TSA}((NROp, TSA, ts_1)$$

unde TSA şi ts_1 reprezintă identificatorul TSA-ului şi respectiv timpul aplicat.

- La pasul 3 *Alice* iniţiază o sesiune de lucru cu Serverul de Non-Repudiere (NRS), transmiţându-i o cerere de non-repudiere $NR - req$.
- În paşii 4 şi 5 *Alice* transmite către NRS dovada de non-repudiere a originii (NRO) obţinută la primii doi paşi, precum şi o dovadă parţială de non-repudiere a recepţiei:

  $$NRRp = (Bob, Alice, h(NRO))$$

  unde h este o funcţie de dispersie criptografică. Pentru a evita atacuri de tip replay, se foloseşte o secvenţă de tip provocare - răspuns bazată pe un nonce n_1, generat de serverul NRS la pasul 4.

  Fiind criptat de NRS cu cheia publică a lui *Alice*, nonce-ul transmis nu poate fi accesat decât de *Alice*.
- În pasul 6, serverul NRS iniţiază generarea dovezii de non-repudiere a recepţiei transmiţându-i lui *Bob* un nonce n_2 (necesar mai târziu) şi dovada parţială de non-repudiere $NRRp$ primită de la *Alice*.
- În paşii 7 şi 8, *Bob* generează – prin intermediul terţului de încredere dat de TSA – dovada de non-repudiere a recepţiei (NRR). *Bob* trimite spre TSA o dovadă parţială de non-repudiere a recepţiei, semnată de el:

  $$NRRps = Sig_{Bob}(NRRp) = Sig_{Bob}(Bob, Alice, h(NRO))$$

  şi primeşte de la TSA marca temporală peste această semnătură:

  $$NRR = Sig_{TSA}(NRRps, TSA, ts_2) = Sig_{TSA}(Sig_{Bob}(Bob, Alice, h(NRO))).$$

- La pasul 9 *Bob* trimite către Serverul de Non-Repudiere (NRS), dovada de non-repudiere a recepţiei NRR, însoţită de nonce-ul n_2 (pentru a evita atacurile de tip replay).
- În acest moment, Faza 1 s-a încheiat şi NRS – având ambele dovezi de non-repudiere – este în măsură să treacă la Faza 2 a protocolului: transmiterea acestor dovezi către *Alice* şi *Bob*.

  Astfel, în paşii 10 şi 11, *Bob* va primi dovada de non-repudiere a originii mesajului (NRO) şi – împreună cu acesta – şi mesajul propriu zis M; similar, *Alice* primeşte dovada de non-repudiere a recepţiei mesajului (NRR).

Observaţia 2.2.

1. *Ideea principală a protocolului Coffey - Saidha este aceea că dovezile de non-repudiere a originii şi respectiv a recepţiei nu sunt trimise celor două entităţi până ce acestea nu depun aceste dovezi semnate la Serverul de Non-Repudiere.*

2. *Dovezile de non-repudiere deţinute de Alice şi Bob nu conţin semnătura NRS-ului. Acestea sunt:*

   - $NRO = Sig_{TSA}(Sig_{Alice}(Alice, Bob, M), TSA, ts_1)$;
   - $NRR = Sig_{TSA}(Sig_{Bob}(Bob, Alice, h(NRO)), TSA, ts_2)$.

3. *Cele două entităţi nu comunică niciodată direct; schimbul de mesaje (mesajul M şi dovezile de non-repudiere) se face numai prin intermediul Serverului de Non-Repudiere.*

4. *Mesajul M ajunge la Bob odată cu dovada de non-repudiere a originii NRO şi nu înainte ca NRS să deţină dovada de non-repudiere a recepţiei NRR.*

5. *În cazul soluţionării unei dispute în care Alice susţine că Bob a primit mesajul M, arbitrul cere NRS-ului dovada de non-repudiere a recepţiei (NRR) precum şi dovada de non-repudiere a originii (NRO).*
   Dacă aceste dovezi nu pot fi oferite, afirmaţia lui Alice este respinsă. Altfel, arbitrul verifică semnătura lui Alice şi marca temporală aplicată de TSA pe această semnătură. De asemenea verifică valoarea $h(NRO)$ din cadrul dovezii NRR, semnă-tura lui Bob şi marca temporală aplicată peste aceasta.
   În caz de succes, arbitrul aprobă afirmaţia lui Alice.

6. *În cazul soluţionării unei dispute în care Bob susţine că Alice a trimis mesajul M, arbitrul cere NRS-ului sau lui Bob dovada NRO. Dacă aceasta nu poate fi oferită, afirmaţia lui Bob este respinsă.*
   Altfel, arbitrul verifică dacă mesajul din NRO satisface informaţia oferită de Bob. De asemenea, verifică semnătura lui Alice şi marca temporală aplicată de TSA pe această semnătură. În caz de success, arbitrul aprobă afirmaţia lui B.

7. *TSA-ul participă activ la generarea dovezilor de non-repudiere; cei doi participanţi precum şi Serverul de Non-Repudiere, trebuie să aibă încredere în acesta şi să verifice semnăturile digitale ale TSA-ului aplicate la marcarea temporală a mesajelor (pentru a se asigura că dovezile generate sunt valabile la soluţionarea disputelor ulterioare).*

8. *Alice şi Bob nu trebuie să aibă încredere unul în celălalt, dar trebuie să aibă încredere în cele două TTP-uri.*

9. *Atacurile de tip replay în relaţia NRS - Entitate sunt ocolite prin utilizarea unor secvenţe de tip provocare -răspuns, bazate pe informaţii de tip nonce transmise de NRS.*

Concluzie: Utilizarea de TTP-uri in-line (pentru asigurarea dovezilor necesare în serviciile de non-repudiere) poate fi o soluţie viabilă. Există însă câteva dezavantaje majore care descurajează punerea în practică a acestui tip de arhitectură:

1. TTP-ul trebuie să gestioneze baze de date destul de mari în care să stocheze mesajele pe care le primeşte pentru a le retransmite.

2. La nivelul TTP-ului, "gâtuirea" tranzacţiilor este maximă.

3. Gestiunea centralizată de informaţii sensibile în cantităţi mari poate constitui o problemă, necesitând un nivel suplimentar de confidenţialitate la nivelul terţului.

Protocoale de non-repudiere bazate pe TTP-uri on-line

În cazul protocoalelor de non-repudiere bazate pe TTP-uri on-line, TTP-ul nu acţionează ca Agent de Expediere (intermediar pentru fiecare tranzacţie între entităţi). El intervine totuşi în cadrul fiecărei sesiuni a protocolului.

Dintre protocoalele de acest tip vom detalia pe cel propus de Zhou şi Gollmann ([32]) în 1996. Aici TTP-ul funcţionează ca un director de publicare read-only, de unde entităţile participante îşi obţin informaţii necesare pentru dovezile de non-repudiere.

Protocolul este descris de următoarele tranzacţii:

1. *Alice* (ca entitate emitentă):

   (a) criptează mesajul M folosind cheia simetrică k : $c = E_k(M)$;

   (b) generează dovada originii lui c:

   $$EOO_c = Sig_{Alice}(Bob, l, t, c) \quad \text{unde } l = Hash(M, k)$$

   (c) trimite lui *Bob* perechea (c, EOO_c).

2. *Bob* (ca entitate receptoare) răspunde cu dovada recepţiei mesajului criptat c:

   $$EOR_c = Sig_{Bob}(Alice, l, t, c).$$

3. *Alice* trimite unui terţ de încredere TTP cheia simetrică k şi dovada depunerii cheii la TTP:

   $$Sub = Sig_{Alice}(Bob, l, t, k)$$

4. TTP:

   (a) trimite spre ambele entităţi confirmarea faptului că deţine cheia k:

   $$Con = Sig_{TTP}(Alice, Bob, l, t, k)$$

   (b) trimite lui Bob cheia simetrică k.

   (c) Bob poate recompune mesajul $M = D_k(c)$.

Dovada de non-repudiere a originii va fi:

$$NRO = (EOO_c, Con)$$

Dovada de non-repudiere a recepţiei va fi:

$$NRR = (EOR_c, Con)$$

În cazul unei dispute:

- Bob poate dovedi non-repudierea originii:
  - EOO_c – dovedeşte că $Alice$ i-a trimis mesajul criptat cu o cheie simetrică;
  - Con – dovedeşte că Bob a putut obţine de la TTP cheia simetrică de decriptare, deci a putut obţine şi mesajul.
- $Alice$ poate dovedi non-repudierea recepţiei:
  - EOR_c – dovedeşte că Bob i-a confirmat primirea mesajului (criptat cu o cheie simetrică);
  - Con – dovedeşte că TTP-ul a publicat cheia simetrică, deci Bob a avut acces la această cheie, pentru decriptarea şi obţinerea lui M.
- TTP-ul poate dovedi că $Alice$ i-a trimis cheia simetrică, prezentând Sub.

Observaţia 2.3.

- *După epuizarea primului pas, dacă Bob nu răspunde cu dovada primirii mesajului criptat EOR_c sau comunicaţia se întrerupe, Alice va opri protocolul fără a transmite cheia simetrică de decriptare către TTP.*
  Astfel Bob nu va avea acces la această cheie şi – implicit – nici la mesaj.
- *Toate mesajele din cadrul protocolului sunt legate între ele prin intermediul etichetei l.*

- *După primirea dovezii de confirmare a mesajului criptat, Alice trebuie să compare eticheta din confirmare cu cea trimisă odată cu mesajul criptat. Altfel va pierde o eventuală viitoare dispută.*

- *Dacă Alice nu trimite către TTP cheia simetrică k, protocolul se încheie şi niciunul din parteneri nu este în avantaj.*
  Bob nu are mesajul şi nici dovada completă a originii lui (îi lipseste Con), iar lui Alice îi lipseşte Con, deci nu are dovada că TTP-ul a publicat cheia simetrică de decriptare şi nu poate dovedi non-repudierea recepţiei.

- *După primirea cheii k şi a lui Sub, TTP-ul va publica tuplul $(Alice, Bob, l, k, Con)$ într-o intrare specifică lui Alice, de unde Bob poate obţine cheia de decriptare.*

- *Protocolul este protejat la atacuri de tip denial-of-service. Entităţile rău intenţionate nu pot trimite chei false în numele lui Alice, deoarece acestea trebuie semnate de ea.*

Ca o concluzie, în obţinerea de servicii de non-repudiere, TTP-urile on-line pot constitui o soluţie mai bună decât cele in-line. Motive:

1. TTP-ul nu acţionează ca Agent de Expediere ci ca un Agent de Certificare pentru cheile de decriptare.

2. Entităţile nu schimbă mesajele prin intermediul TTP-ului, însă acesta participă activ în cadrul fiecărei instanţe a protocolului.

3. În cadrul schimbului de mesajelor există si un parametru (opţional) de timp t.

   Pentru a evita o încărcare prea mare a TTP-ului, parametrul t – agreat de *Alice* şi *Bob* încă din primele două mesaje – specifică o perioadă de timp cât TTP-ul va păstra disponibilă informaţia de cheie şi dovada publicării ei.

Protocoale de non-repudiere bazate pe TTP-uri off-line

Un TTP este off-line într-un protocol de non-repudiere dacă nu intervine decât în situaţii în care apar probleme în cadrul protocolului.

TTP-urile de acest tip au fost introduse în protocoale unde în general nu apar probleme. Din acest motiv protocoalele cu TTP off-line se numesc şi *protocoale optimiste*.

Primele protocoale care folosesc TTP-uri off-line sunt descrise de Asokan ([2],[3]). De asemenea, Micalli propune ([21]) un protocol de certificat e-mail cu cerinţe de non-repudiabilitate bazat pe TTP-uri off-line transparente (invizibile)[1].

[1]Un TTP off-line are proprietatea de a fi transparent dacă la sfârşitul protocolului, analizând numai dovezile produse, va fi imposibil de afirmat că TTP-ul a intervenit sau nu în cadrul derulării acestuia.

Totuşi, primele protocoale dedicate clar ideii de non-repudiere cu TTP-uri off-line sunt descrise de Kremer, Markowitch ([14]) şi de Jhou ([31]).

În general toate protocoalele de non-repudiere cu TTP-uri off-line conţin cel puţin două sub-protocoale diferite:

- Un protocol principal (main) utilizat în cazurile normale – în care entităţile se comportă corect una faţă de cealaltă.
  TTP-ul nu intervine în acest protocol.

- Un protocol de recuperare (recovery) pentru situaţiile cu probleme.
  În acest caz este necesară intervenţia TTP-ului pentru furnizarea dovezilor de non-repudiere.

În plus, unele protocoale conţin şi:

- un protocolul adiţional de ieşire forţată (abort), care poate fi declanşat de una din entităţi în anumite situaţii; în urma acestuia se va apela ulterior protocolul de recuperare (recovery) pentru rezolvarea disputelor.

Vom prezenta ca studiu de caz protocolul de non-repudiere cu TTP off-line propus de Kremer-Merkowitch ([14])).

El se desfăşoară între *Alice* (ca entitate emitentă) şi *Bob* (ca entitate receptoare), iar dacă apar probleme de non-repudiere este implicat şi TTP-ul.

Dovezile de non-repudiere sunt obţinute pe baza mecanismului de semnătură digitală.

În cadrul protocolului sunt generate următoarele dovezi:

- dovada originii mesajului criptat:

$$EOO = Sig_{Alice}(Bob, TTP, h(c))$$

- dovada expedierii cheii k de criptare a mesajului (criptată cu cheia publică a TTP-ului): $Sub = Sig_{Alice}(Bob, PK_{TTP}(k))$.

- dovada de non-repudiere a recepţiei mesajului criptat şi a cheii de criptare a mesajului (criptată cu cheia publică a TTP-ului):

$$NRR = Sig_{Bob}(Alice, TTP, h(c), PK_{TTP}(k))$$

- dovada originii cheii k de criptare a mesajului: $EOO_k = Sig_{Alice}(Bob, k)$.

- dovada cererii de recuperare:

$$Rec = Sig_{Bob}(Y)$$

- dovada de confirmare a cheii de criptare:

$$Con_k = Sig_{TTP}(Alice, Bob, k)$$

Schema protocolului Kremer-Merkowitch este:

1. *Alice*:
   (a) Generează o cheie de sesiune k şi criptează mesajul M: $c = E_k(M)$.
   (b) Calculează $PK_{TTP}(k)$ folosind cheia publică a TTP-ului;
   (c) Construieşte dovezile EOO (privind originea mesajului criptat) şi Sub (privind expedierea cheii de sesiune criptată).
   (d) Trimite lui *Bob* aceste informatii: $(c, PK_{TTP}(k), EOO, Sub)$.
2. *Bob* trimite către *Alice* dovada NRR de non-repudiere a recepţiei informaţiilor.
3. *Alice* trimite lui *Bob* cheia de sesiune k (necesară decriptării mesajului M) şi dovada EOO_k privind originea acestei chei.

   Dacă *Bob* nu primeşte informaţiile din pasul 3 al protocolului (din cauza lui *Alice* sau a canalului de comunicaţie), el apelează la protocolul de recuperare (recovery) a mesajului; lucru necesar deoarece *Bob* nu are încă acces la M.
4. *Bob* trimite spre TTP o cerere de recuperare Y:

$$(Y, h(c), PK_{TTP}(k), Rec, Sub, NRR, EOO)$$

5. TTP-ul:
   (a) Decriptează cheia de sesiune k:
   (b) Trimite lui *Alice*: cheia k şi dovada NRR (semnată de *Bob*) care atestă faptul că *Bob* are mesajul criptat şi poate primi acum şi cheia de decriptare.
   (c) Trimite lui *Bob* cheia de sesiune k decriptată şi confirmarea dovezii Con_k pentru această cheie.

Observaţia 2.4.

- *În primul pas Alice trimite mesajul criptat şi cheia simetrică k – criptată cu cheia publică a TTP-ului. Acest lucru va permite TTP-ului – în faza de recuperare – să extragă cheia k şi să o furnizeze lui Bob.*
  După acest mesaj, nici Alice şi nici Bob nu posedă dovezi complete de non-repudiere.

- *La terminarea pasului 2, deşi Bob nu poate accesa încă mesajul, trimite lui Alice dovada completă de non-repudiere a recepţiei.*
  Dacă Bob nu primeşte ulterior cheia de decriptare (adică pasul 3 nu mai are loc), el o poate obţine de la TTP prin protocolul de recuperare.

  De asemenea, dacă cheia simetrică criptată primită de la Alice la pasul 1 este eronată (de exemplu a primit $PK_{TTP}(k_1)$ cu $k_1 \neq k$), atunci Bob poate demonstra că a primit un $PK_{TTP}(k_1)$, iar dovada trimisă de el lui Alice devine invalidă.

- *Dovada completă de non-repudiere a expedierii este compusă din tripletul $\{EOO, Sub, EOO_k\}$ şi este obţinută de Bob abia după completarea pasului 3 din protocol.*

  Dacă pasul 3 nu are loc (din diverse motive), dovada de non-repudiere a expedierii va fi completată prin protocolul de recuperare şi este formată din tripletul $\{EOO, Sub, Con_k\}$.
  În acest caz, Bob primeşte şi cheia simetrică de decriptare.

- *Bob poate încerca să trişeze şi să lanseze protocolul de recuperare mai devreme.*

  Imediat după pasul 1 el poate întrerupe protocolul principal fără să mai trimită lui Alice dovada NRR. Pentru a furniza corect dovezile de non-repudiere, TTP-ul oferă totdeauna ambelor entităţi dovezile necesare.

  De asemenea este necesar ca TTP-ul să valideze NRR-ul primit de la Bob în pasul 4 (primul pas al protocolului de recuperare). Validarea presupune de fapt verificarea semnăturii lui Bob din acest NRR.
  Acest lucru este necesar pentru a putea fi sigur că înainte ca Bob să primească cheia de decriptare şi dovada expedierii acestei chei, Alice a primit dovada corectă de non-repudiere a recepţiei.

Din aceste observaţii rezultă că niciuna dintre cele două entităţi implicate nu este avantajată în ceea ce priveste expedierea şi recepţia informaţiei utile şi a dovezilor corespunzătoare necesare.

Există însă un dezavantaj pentru *Alice*; acesta apare în situaţia în care *Bob* refuză să completeze pasul 2, caz în care *Alice* nu va primi dovada recepţiei.

În această situaţie:

- *Bob* poate lansa imediat protocolul de recuperare. Nici o problemă, deoarece ambele entităţi primesc toate informaţiile pentru a încheia corect sesiunea protocolului.

- *Bob* poate aştepta o perioadă de timp (mai mică sau mai mare) şi abia după aceea va lansa protocolul de recuperare.
  În acest caz *Alice* trebuie să păstreze deschisă sesiunea de protocol, până când *Bob* se hotărăşte să lanseze protocolul de recuperare.

Acest lucru îl plasează pe *Bob* pe o poziţie avantajată faţă de *Alice*, deoarece *Bob* nu poate fi niciodată în situaţia de a depinde de *Alice*. Se spune despre un astfel de protocol că *nu este oportun*.

Concluzie:
Arhitectura bazată pe TTP-uri off-line constituie cea mai bună soluţie pentru asigurarea dovezilor necesare de non-repudiere deoarece:

- Implicarea TTP-ului poate fi doar ocazională. În majoritatea timpului, tranzacţiile se desfăşoară doar între entităţile emiţător şi receptor.
- TTP-ul nu trebuie să stocheze informaţiile primite de la entităţi decât în anumite situaţii ceea ce diminuează necesarul de resurse.
- Caracterul off-line al protocoalelor elimină dezavantajele cu care se confruntă serviciile on-line (redundanţa serviciului şi a conectivităţii, ameninţări cu diverse atacuri cum ar fi cele de tip Denial-of-Service, intruziune etc.).

2.5 Servicii de Marcă Temporală

Definiţia 2.3. *Marcarea temporală a documentelor este un mecanism prin care se poate determina dacă un document electronic a fost creat înainte de un moment de timp dat.*

Exemplul 2.3. *Fără un astfel de mecanism nu putem avea încredere în documente semnate electronic cu primitive criptografice de semnătură care au fost depăşite tehnologic. Fără cunoaşterea momentului la care a fost creată semnătura, aceasta poate fi oricând repudiată.*

După semnarea documentului, certificatul semnatarului poate fi revocat pe motiv de compromitere a cheii private, ceea ce permite acestuia să afirme că nu el a fost cel care a semnat documentul.

Validarea unei semnături trebuie să se poată realiza şi după revocarea sau expirarea certificatului de semnare.

Multe din serviciile şi tehnicile de securitate existente (semnăturile electronice, autentificarea, arhivarea electronică a documentelor, protocoale de non-repudiere etc.) sunt bazate pe posibilitatea de a putea stabili cu exactitate momentul de timp când anumite date au fost supuse unor operaţii.

Atunci când există termene de predare/primire, există necesitatea de a dovedi că un document a fost predat la termen, acesta constituind un alt caz de utilizare a mărcilor de timp.

Orice formă de notariere electronică include şi o marcă temporală. Cu toate că mărcile temporale constituie o soluţie pentru diverse aplicaţii, în multe cazuri adevărata problemă o constituie obţinerea lor într-o manieră sigură şi eficientă ([19]).

Marcarea temporală a unui document se face prin adăugarea unei amprente (ştampile) de timp la documentul respectiv.

În general, majoritatea schemelor de marcare temporală folosesc:

1. un rezumat al documentului care trebuie marcat (marca temporală se calculează pe rezumatul documentului)
2. un terţ de încredere de tip on-line numit *Autoritate de Marcă Temporală* (TSA – *Time Stamping Authority*).

TSA - ul generează mărcile temporale pentru documente şi garantează că parametrul de timp inclus este corect. Entităţile care verifică marca temporală trebuie să aibă încredere în TSA.

2.5.1 Clasificarea schemelor de marcă temporală

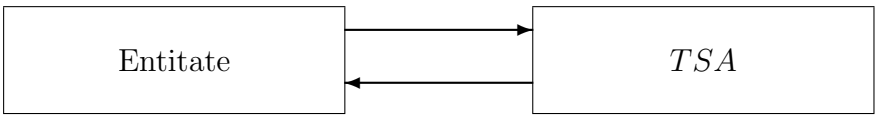
Schemele diferă între ele prin două aspecte de bază:

- mecanismul prin care se generează marca temporală (pornind de la document).
- modalitatea prin care se asigură încrederea în TSA.

Conform clasificării prezentate în [6], schemele de marcare temporală pot fi clasificate în:

1. scheme simple;
2. scheme cu înlănţuire (legături);
3. scheme distribuite.

Indiferent de schema folosită, obţinerea unei mărci temporale se realizează după modelul următor:



Scheme simple

Definesc doar un mecanism prin care se pot obţine mărcile temporale; ele generează mărci temporale independente (nu includ informaţii din alte mărci generate anterior).
Exemple de utilizare al acestui tip de schemă sunt cele propuse de standardele elaborate de IETF în TSP – *Time-Stamping Protocol* (RFC 3161) şi de ISO în ISO $18014-2$.

Soluţia cea mai utilizată de generare a amprentelor de timp se bazează pe criptografie, folosind semnăturile digitale.

Astfel, pentru obţinerea unei mărci temporale pentru un document X:

1. O entitate finală *Alice* calculează o amprentă $h(X)$, pe care o trimite la TSA.
2. TSA-ul generează o marcă temporală, ataşând timpul curent T la $h(X)$ şi semnează marca de timp cu cheia sa privată: $Sig_{TSA}(T, h(X))$.
3. TSA-ul trimite lui *Alice* această marcă generată $Sig_{TSA}(T, h(X))$.

Avantaje:

- Simplitate
- Clientul poate verifica singur marca de timp: orice entitate care verifică marca temporală trebuie de fapt să verifice semnătura generată de TSA.

Dezavantaje:

- Necesită încredere absolută în Autoritatea de Marcare Temporală; este motivul principal pentru care schemele simple prezintă un nivel scăzut de securitate. Haber şi Stornetta au arătat ([12]) că, dacă TSA modifică în mod fraudulos parametrul T pentru a antedata un document, nimeni nu poate detecta acest lucru.
- Posibilitatea de compromitere a cheii private de semnătură; în acest caz toate mărcile temporale generate cu cheia respectivă devin invalide.

Pentru generarea mărcilor de timp independente (cu scheme simple), $ISO\ 18014 - 2$ oferă alte două soluţii: utilizarea codurilor de autentificare (MAC) şi utilizarea obiectelor arhivate.
Aceste metode se bazează pe realizarea unor blocuri MAC (în care TSA-ul foloseşte o cheie simetrică pentru autentificare) sau pe stocarea unor obiecte ce reprezintă o legătură între documentul care trebuie marcat şi timpul marcării.

Ambele metode au dezavantajul că la verificarea mărcilor trebuie implicată şi Autoritatea de Marcare Temporală.

Schemele cu înlănţuire

Aceste scheme încearcă să micşoreze necesarul de încredere acordat TSA-ului şi problemele de compromitere a cheii private a acestuia prin legarea între ele a amprentelor de timp generate.

Schemele includ în marca temporală generată anumite informaţii referitoare la mărcile temporale generate anterior. În acest fel se poate crea un lanţ de mărci temporale care depind una de cealaltă şi stopează posibilitatea TSA-ului de a antedata documente.

Securitatea sistemului creşte, dar procesul de verificare a mărcilor necesită participarea TSA-ului, ceea ce complică implementarea Autorităţii de Marcare şi modelul de funcţionare al schemelor.

Există mai multe abordări de implementare pentru schemele înlănţuite:

- scheme cu înlăţuire liniară;
- scheme cu înlănţuire bazate pe arbori de autentificare Merkle;
- scheme cu înlănţuire bazate pe arbori binari: $ISO\ 18014-3$, schemele Haber-Stornetta ([12]), schema Bayer ([4]), schema Benaloh - De Mare ([5]), schema Buldas ([8]) etc.

Schemele cu legături funcţionează în general în trei paşi:

1. *Agregarea*: toate documentele primite de TSA într-un interval scurt de timp sunt considerate simultane şi vor fi colectate de TSA pentru a fi prelucrate împreună. Prin agregare se obţine un şir binar (marca temporală) calculat pe baza documentelor simultane.

   Scopul pasului de agregare este de a scădea încărcarea TSA-ului.

2. *Înlănţuirea*: urmăreşte obţinerea unui lanţ ordonat într-un singur sens (one way) între valorile obţinute la pasul de agregare.
   Rezultatul pasului de agregare curent este folosit pentru a fi "legat" de rezultatul pasului de agregare anterior.

   Prin acest mecanism se obţine o autentificare temporală relativă: rezultatele rundelor de agregare pot fi comparate între ele în ceea ce priveşte ordinea.

   Operaţia de înlănţuire (de legare) se poate face de exemplu folosind funcţii de dispersie. Dacă se doreste doar garantarea ordinii documentelor, valoarea efectivă de timp nu este neapărat necesară în schemele cu înlănţuire.

3. *Publicarea*: pentru a-i putea audita activitatea şi a creşte astfel securitatea, TSA-ul publică periodic (de exemplu lunar) ultima legătură obţinută la pasul de înlănţuire. În acest fel TSA-ul nu mai poate interveni în lanţul creat până la ultima legătură publicată, pentru a "antedata" de exemplu un document.

Schema propusă de Haber-Stornetta este una din cele mai simple şi folosite scheme cu legături. Ea se bazează pe utilizarea unei funcţii de dispersie criptografică (pentru înlănţuirea amprentelor de timp generate) şi pe semnătură digitală (pentru autentificarea mărcilor).

Astfel, pentru cel de-al n-lea document, $H_n = h(X)$ primit de TSA pentru marcare, amprenta de timp generată va fi

$$Sig_{TSA}(n, T_n, ID_n, H_n, L_n)$$

unde T_n este timpul curent, ID_n este identificatorul clientului, iar L_n este informaţia de legătură cu mărcile generate anterior, fiind definit recursiv de formula:

$$L_n = (T_{n-1}, ID_{n-1}, H_{n-1}, h(L_{n-1}))$$

În opinia autorilor singura soluţie de a ataca schema este aceea de a falsifica un lanţ suficient de lung de legături.
Pentru a preîntâmpina acest lucru, pasul de publicare devine obligatoriu .

Publicarea nu este eficientă dacă se face pentru fiecare marcă generată.
De aceea schemele cu legături sunt totuşi vulnerabile în intervalul petrecut de la ultima publicare până la următoarea publicare. În acest interval, un TSA compromis poate antedata orice document şi poate reface lanţul de legături.

O soluţie de auditare ar putea fi realizată doar cu colaborarea tuturor clienţilor din acest interval, ceea ce nu este deloc practic.

Schemele distribuite

Sunt obţinute prin utilizarea mai multor TSA-uri pentru generarea amprentelor de timp. În acest mod încrederea într-un singur TSA poate fi redusă.

Pentru falsificarea unei amprente de timp a unui document trebuie să colaboreze mai multe TSA-uri.

Schemele distribuite pot fi obţinute prin combinarea schemelor simple şi a schemelor înlănţuite. De exemplu, o schemă distribuită se poate obţine prin serializarea unei scheme simple cu o schemă înlănţuită care rulează la două TSA-uri diferite.

Schemele distribuite prezintă cel mai mare grad de securitate, dar au o complexitatea ridicată. În consecinţă fiabilitatea sistemului scade, deoarece toate componentele sale trebuie să fie funcţionale.

Exemple de scheme distribuite putem găsi în [1], [7].

2.5.2 Problemele schemelor de marcă temporală

Nici una din schemele de marcă temporală prezentate nu este perfectă, fiecare din ele având avantajele dar şi dezavantaje sale. În [20] sunt indicate o serie de probleme cu care se confruntă schemele actuale de marcă temporală:

- Schemele simple nu prezintă control asupra parametrului de timp. Din această cauză nu există posibilitatea de a contracara un comportament fraudulos al TSA-ului, acesta putând antedata documentele.

- Caracterul on-line caracteristic TSA-urilor face ca acestea să devină vulnerabile la problemele specifice acestui tip de serviciu: atacuri de intruziune, disponibilitatea serviciilor, resurse de reţea redundante etc.

- În cazul schemelor cu înlănţuire, TSA-ul poate antedata documente în intervalul dintre paşii de publicare. Singura soluţie de verificare a procesului ar fi prin identificarea şi colaborarea tuturor utilizatorilor sistemului din acest interval, ceea ce face schema nepractică.

- Metoda cea mai folosită pentru generarea mărcilor de timp se bazează pe semnături digitale. În cazul schemelor simple, compromiterea cheii private invalidează toate mărcile generate. Pentru schemele cu înlănţuire, sunt invalidate toate mărcile care nu au fost încă incluse în legătura publicată.

- La schemele cu legături, pentru verificarea lanţurilor de amprente, acestea trebuie arhivate pe o perioadă suficient de mare, iar operaţia de verificare necesită implicarea TSA-ului.

- Datorită faptului că în general procesul de marcare temporală se bazează pe un singur serviciu TSA, schemele devin vulnerabile la atacuri de tip Denial-of-Service. Este foarte greu de implementat soluţii de balansare a cererilor bazate pe servicii TSA redundante, deoarece mărcile de timp trebuiesc corelate.

- Pasul de publicare necesar schemelor de înlănţuire nu este întotdeauna uşor de realizat. În general pentru procesul de publicare sunt preferate diverse mijloace cum ar fi reviste sau ziare cu circulaţie publică, iar în acest caz publicarea nu poate fi automatizată.

- Nu toate schemele sunt echivalente ca nivel de securitate. Anumite scheme sunt succeptibile la diverse atacuri.

- Tehnic, schemele de marcare sunt bazate pe primitive criptografice cum ar fi funcţiile de dispersie criptografică sau criptografia cu chei publice.

  Compromiterea acestora implică automat şi compromiterea schemelor de marcare.

- Inter-operabilitatea constituie o altă problemă a schemelor. Pentru foarte multe scheme, formatul cererilor, al mărcilor de timp şi al legăturilor nu sunt bine specificate.

  Standardul TSP (RFC 3161) tratează doar schemele simple, propunând un format $ASN.1$ pentru cereri şi mărcile temporale. În [29] este o propunere mai generală bazată pe formate XML (care acoperă şi mecanisme de înlănţuire), însă nu există încă un standard în acest sens.

  În acest moment majoritaea aplicaţiilor sunt bazate pe standardul TSP, iar unele autorităţi de marcare temporală folosesc şi elemente de înlănţuire.

- Schemele distribuite prezintă cel mai bun grad de securitate, însă acestea sunt greu şi scump de implementat practic. Verificarea mărcilor presupune implicarea tuturor TSA-urilor din cadrul sistemului de marcare.

- Cerinţele actuale privind realizarea semnăturilor digitale obligă folosirea mărcilor temporale pentru a garanta validitatea semnăturilor electronice pe termen lung

(ETSI 101 733). Infrastructurile PKI sunt afectate de necesitatea unui serviciu TSA de tip on-line deoarece scade din scalabilitatea şi usurinţa de folosire.

2.5.3 Standardizarea serviciilor de marcă temporală

ISO (*International Organization for Standardization*) a abordat problematica serviciilor de marcare temporală în suita de standarde ISO 18014.

Conform acestora, entităţile implicate într-un serviciu de marcare temporală sunt:

1. o Autoritate de Marcare Temporală (TSA), definită ca terţ de încredere care oferă dovezi că anumite date au existat la un moment de timp dat şi garantează corectitudinea parametrului de timp.
2. un client: entitatea care este în posesia datelor şi doreste ca acestea să fie marcate temporal.
3. un verificator: entitatea care confirmă validitatea unei mărci temporale.

Conform standardului ISO $18014-1$ există două protocoale de marcă temporală:

- **Protocolul de emitere**:
  1. Clientul trimite o cerere de marcare temporală.
  2. TSA verifică formatul cererii şi generează marca temporală.
  3. TSA returnează spre client marca generată.

  Marca temporală conţine cel puţin un parametru de timp şi o amprentă a datelor pentru care se generează marca, legate între ele printr-o tehnică criptografică.
- **Protocolul de verificare**:
  1. Verificatorul lansează procedura de verificare a unei mărci temporale.
  2. Dacă şi de câte ori este necesar, el solicită Clientului şi TSA-ului informaţii suplimentare.
  3. Pe baza tuturor informaţiilor primite, Verificatorul controlează marca temporală.

Celelalte două părţi ale standardului ISO prezintă câteva modalităţi de legare a amprentei datelor cu parametrul de timp, folosind diferite tehnici criptografice.

- ISO $18014-2$ defineşte trei mecanisme de generare a amprentelor de timp independente, specifice schemelor simple:

1. folosind semnăturile electronice pentru asigurarea autenticităţii şi integrităţii mărcilor emise – model preluat din PKIX TSP (RFC 3161);
2. pe bază de MAC-uri şi cu chei simetrice: asigură garantarea integrităţii şi autenticităţii mărcilor;
3. pe bază de obiecte arhivate; în acest caz, TSA-ul întoarce clientului ca marcă temporală numai o referinţă cu un identificator al datelor marcate.

Ultimele două soluţii necesită ca verificatorul să obţină de la TSA anumite informaţii pentru verificare:

- În cazul bazat pe MAC-uri, verificatorul trimite TSA-ului marca iar TSA-ul întoarce răspunsul cu privire la integritatea şi autenticitatea acesteia.

- La utilizarea de obiecte arhivate, verificatorul cere TSA-ului direct parametrul de timp asociat cu identificatorul primit.

În toate situaţiile, verificatorul trebuie să aibă încredere deplină în TSA.

- ISO $18014-3$ defineşte mecanisme de generare a amprentelor de timp cu legătură.

  Aici sunt construiţi cei trei paşi specifici schemelor înlănţuite: agregarea, înlănţuirea şi publicarea.

Niciuna din cele trei componente ale standardului ISO nu face referire la schemele distribuite.

$PKIX$ Time-Stamp Protocol (TSP)

TSP-ul propus de $IETF$ $PKIX$ în standardul Internet RFC 3161 este cel mai răspândit protocol utilizat în aplicaţiile şi serviciile de marcare temporală existente. El foloseşte schema simplă de generare a amprentelor de timp şi se bazează pe mecanismul de semnătură digitală pentru garantarea autenticităţii şi integrităţii mărcilor de timp emise de TSA.

Standardul defineşte TSA-ul ca fiind "*un TTP care creează mărci temporale pentru a indica faptul că anumite date au existat la un moment particular de timp*".

Modelul tranzacţional propus este de tip client - server. Clientul face o cerere către TSA, în care transmite amprenta $h(X)$ corespunzătoare documentului care trebuie marcat şi obţine marca temporală $Sig_{TSA}(h(X), T)$.

Pe lângă detaliile legate de formatul cererilor şi răspunsurilor, standardul defineşte şi conceptul de politică de marcare temporală.
Aceasta reprezintă un set de reguli de funcţionare a TSA, elementele care ţin de securitatea serviciului, mecanismele de gestiune a cheii private a TSA, algoritmii de dispersie criptografică acceptaţi, algoritmul de semnătură folosit, acurateţea parametrului de timp etc.

Un TSA poate funcţiona după una sau mai multe politici care trebuie să fie clar definite, publicate şi unic identificabile prin intermediul unui *ObjectIdentifier* (OID). Clientul poate alege în cerere una din politicile TSA-ului sau îl poate lăsa pe acesta să folosească una din politicile sale.

Standardul stabileşte şi câteva cerinţe de securitate asupra TSA-ului:

- foloseşte o sursă de timp exactă şi de încredere;
- include informaţia de timp de încredere în fiecare marcă de timp emisă;
- acceptă din partea clientului numai cereri de marcare bazate pe una dintre politicile sale de funcţionare;
- verifică şi acceptă din partea clientului numai funcţii de dispersie considerate de încredere la momentul cererii;
- nu asociază parametrul de timp niciodată cu datele ci doar cu amprenta acestora;
- semnează fiecare token de timp emis ca răspuns şi foloseşte o cheie dedicată în acest sens;
- identifică în mod unic fiecare marcă de timp emisă, folosind numere întregi;
- păstrează caracterul anonim al clientului; marca de timp generată nu trebuie să identifice în nici un fel clientul.

Standardul propune mai multe mecanisme posibile de transport bazate pe protocoalele $HTTP$, FTP, $e-Mail$ sau TCP/IP. În practică, standardele de securitate care folosesc protocolul TSP pentru obţinerea mărcilor temporale necesare aleg de obicei varianta $HTTP$.

Protocolul descris în $RFC3161$ prezintă câteva slăbiciuni care se pot transforma în falii de securitate ([27]):

1. Nu se precizează nici un mecanim de protecţie împotriva unui comportament fraudulos al TSA-ului. Este folosită schema simplă de generare a mărcilor, schemă care este cea mai puţin sigură din acest punct de vedere. Verificatorii trebuie să accepte TSA-ul ca fiind de încredere în totalitate.

   Implementările TSA pot fi îmbunătăţite prin adăugarea unor mecanisme specifice schemelor înlănţuite, însă standardul nu conţine nici un astfel de element pentru a asigura un nivel de interoperabilitate în acest sens.

2. Standardul nu prevede alte metode mai rapide – comparativ cu semnătura – pentru obţinerea mărcilor temporale. Operaţia de semnare este mare consumatoare de timp ceea ce face sistemul vulnerabil la atacuri de tip Denial-of-Service.

O soluţie ar putea fi utilizarea unor acceleratoare criptografice specializate, care să sporească viteza de rezolvare a cererilor din partea TSA.

3. Compromiterea cheii private invalidează toate token-urile de timp generate cu ea.

   Siguranţa sistemului depinde practic de securitatea acestei chei. De aceea, implementările TSA trebuie să aibă în vedere măsuri speciale de protecţie a cheii, bazate pe dispozitive speciale de tip HSM (*Hardware Secure Module*) şi pe mecanisme de acces dual de tip "K-din-N" cu roluri atent definite şi controlate, ceea ce presupune costuri ridicate de implementare şi operare pentru TSA.

4. Dacă nu este folosit corect, protocolul poate fi vulnerabil la atacuri de tip *replay*.

   Pot fi mai multe scenarii în care poate fi executat atacul: un document trebuie datat de mai multe ori, sau datarea unui document nu s-a putut face dintr-un anumit motiv şi trebuie retransmis etc.
   Un atacator poate întoarce în aceste situaţii un răspuns cu un parametru de timp învechit. Protocolul conţine suport pentru mecanismul clasic de protecţie la acest atac, bazat pe informaţia de tip nonce, însă utilizarea sa este opţională.

   Cealaltă soluţie (sugerată în standard) – care foloseşte o fereastră glisantă de timp – este nepractică deoarece se bazează pe o sursă locală de timp, care poate fi total dereglată şi – în plus – complică destul de mult implementările clienţilor.

5. Standardul nu reglementează suficient de bine componenta de politică a TSA.

   Aceasta este doar amintită, nefiind detaliate aspecte utile cum ar fi: ce elemente minime trebuie să conţină o politică, numărul de politici pe care trebuie să le accepte un server, cum poate afla un client politicile oferite de un TSA etc.

   Forul de standardizare European $ETSI$ reglementează o parte din aceste aspecte prin specificaţia ETSI 102 023.

6. Standardul nu prevede mecanisme proprii de autentificare a clienţilor.

   De exemplu, pot exista cerinţe ca un serviciu TSA să fie disponibil numai unor clienţi care plătesc un abonament. În astfel de scenarii trebuie implementate mecanisme proprietare de gestiune a clienţilor respectivi.

   O soluţie poate fi bazată pe un serviciu de tip proxy care să intermedieze relaţia clienţilor cu TSA-ul şi care va implementa mecanismele de identificare, autentificare şi contorizare a cererilor sosite. Autentificarea se va face prin mecanisme externe protocolului TSP – de exemplu folosind protocolul TLS (RFC 5246).

7. Răspunsurile de eroare sunt vulnerabile la atacuri de tip Man-in-the-Middle deoarece nu sunt semnate de TSA. Un atacator poate intercepta astfel de atacuri şi poate modifica informaţiile din răspuns (poate schimba, de exemplu motivul erorii).

Această vulnerabilitate a protocolului poate fi exploatată şi sub forma unui atac de tip Denial-of-Service, caz în care toate răspunsurile corecte ale TSA-ului vor fi convertite în răspunsuri de eroare.

2.6 Servicii de Arhivare Digitală

Definiţia 2.4. *Un serviciu de arhivare digitală are ca obiectiv principal salvarea documentelor digitale şi păstrarea lor în siguranţă pentru o anumită perioadă de timp.*

Exemplul 2.4. *Perioada de utilizare a unor informaţii semnate electronic poate fi mai lungă decât perioada de valabilitate a certificatelor de chei publice utilizate la verificarea semnăturii sau în raport cu perioada de criptanaliză a algoritmilor criptografici folosiţi la generarea semnăturii.*

Orice mecanism de arhivare electronică trebuie să ia în considerare: durata de viaţă a mediilor de stocare, mecanisme de disaster-recovery, creşterea capacităţii de criptanalizare a algoritmilor dar şi a puterii de calcul, modificările care apar la nivel de tehnologie hardware sau software.

Sistemele de arhivare de încredere deţin mecanisme de stocare a datelor într-un mod care să asigure şi dovezile necesare privind integritatea lor. Dovezile de integritate sunt generate/obţinute periodic pentru a forma un lanţ continuu care să garanteze integritatea datelor de la momentul arhivării acestora până la data verificării lor.

Există mai multe scheme care tratează problema arhivării pe termen lung. În general acestea urmăresc disponibilitatea datelor pe termen lung, însă pierd din vedere alte aspecte importante cum ar fi migrarea formatelor datelor în timp, astfel încât acestea să poată fi compatibile cu noile implementări hardware si software.

2.6.1 Cerinţele unui serviciu de arhivare pe termen lung

Cerinţele de funcţionare formulate asupra unui serviciu de arhivare pot să difere în raport cu utilizatorii care folosesc serviciul respectiv şi cu necesităţile acestora.

Câteva cerinţe generale specifice unui serviciu de arhivare pe termen lung ($RFC4810$):

1. Un serviciu de arhivare trebuie să permită clienţilor săi realizarea operaţiilor de bază privind:

   (a) depunerea datelor pentru a fi arhivate;

   (b) obţinerea datelor din arhivă;

   (c) ştergerea datelor din arhivă (opţional).

   Datele pot fi transmise în diverse forme (date brute, date semnate şi/sau criptate etc.) şi folosind diverse formate de prezentare (documente PDF, XML, Microsoft Office Word, PPT etc.).

La recepţionarea datelor care urmează să fie arhivate, serviciul furnizează clientului un identificator unic asociat cu datele respective.
Acesta va fi apoi folosit pentru identificarea, obţinerea sau ştergerea datelor din arhivă. Autentificarea clienţilor în vederea realizării acestor operaţii poate fi o cerinţă suplimentară la nivelul serviciului. În acest sens, anumite operaţii pot fi de uz general (de exemplu, obţinerea datelor), sau pot fi permise numai entităţilor autorizate (ştergerea datelor).

2. Garantarea integrităţii datelor arhivate.
Integritatea trebuie asigurată pe toată perioada de arhivare, iar serviciul este responsabil de asigurarea dovezilor care pot demonstra această proprietate.

   Dovezile pot fi generate intern de serviciul de arhivare sau pot fi obţinute de la un serviciu de încredere extern.

3. Funcţionarea în concordanţă cu o politică de arhivare.
Aceasta defineşte caracteristicile de implementare pentru serviciul respectiv şi are mai multe componente: politica de mentenanţă a obiectelor arhivate, politica de autorizare a clienţilor serviciului, politica de securitate a serviciului de arhivare etc.

   Un serviciu de arhivare poate avea în uz mai multe politici.

4. Pe lângă operaţiile principale (depunerea, obţinerea sau ştergerea datelor din arhivă), serviciul trebuie să permită clienţilor săi specificarea altor elemente care privesc datele arhivate: informaţiile de tip "meta-data" asociate cu datele arhivate, politica de arhivare folosită, perioada de arhivare a datelor, posibilitatea de extindere a acestei perioade etc.

5. Asigurarea confidenţialităţii datelor arhivate.
Un client poate cere ca datele respective să rămână confidenţiale chiar şi faţă de serviciul de arhivare. Această cerinţă este destul de dificil de îndeplinit, deoarece serviciul trebuie să asigure în timp operaţiile de conservare a datelor originale (necriptate), fără a avea acces la ele.

6. Metode de transfer a datelor arhivate şi a dovezilor asociate către un alt serviciu de arhivare.
Acest lucru este necesar de exemplu la încetarea activităţii serviciului înainte de terminarea perioadei de arhivare a datelor şi prin cedarea serviciilor sale către un alt serviciu.

   Un alt caz de acest gen poate fi în situaţia când un client doreste să-şi mute datele arhivate de la un serviciu de arhivare la altul.

7. În unele situaţii obiectele de date arhivate nu pot fi independente ci fac parte dintr-un grup.

Ca exemple: o semnătură digitală împreună cu documentul semnat, sau un document împreună cu o serie de traduceri ale sale. În aceste cazuri operaţiile de arhivare trebuie realizate nu pentru un singur obiect, ci pentru un grup de obiecte arhivate.

8. Eficienţa serviciului.
   Serviciile de arhivare pot avea cantităţi mari de date arhivate pe care trebuie să le gestioneze într-un mod eficient. Operaţiile de conservare a datelor şi de (re)generare a dovezilor de integritate pot fi costisitoare ca timp, putere de calcul şi resurse de stocare. O soluţie eficientă în acest sens este furnizată în *RFC* 4998.

9. Asigurarea non-repudierii originii datelor poate fi de asemenea un obiectiv al unui serviciu de arhivare.
   Pot fi folosite mecanisme specifice protocoalelor de non-repudiere, cu sau fără implicarea altor *TTP*-uri.

2.6.2 Tehnici pentru asigurarea integrităţii datelor

Anumite tehnici criptografice (funcţiile de dispersie criptografică, semnăturile digitale) pot asigura integritatea datelor, dar securitatea lor depinde de rezistenţa la procedeele de criptanaliză – care evoluează şi se perfecţionează în timp.

Cheile de semnătură, lungimile amprentelor (prin aplicarea funcţiilor de dispersie) devin la un moment dat ineficiente în raport cu puterea de calcul.

Când discutăm de arhivare pe termen lung trebuie să avem în vedere perioade de la câţiva ani la câteva decenii. În acest timp, cu siguranţă mecanismele criptografice amintite îsi pot pierde proprietăţile de securitate necesare pentru garantarea integrităţii datelor.

Tehnici de bază folosite de sistemele de arhivare

Una din primele metode folosite pentru asigurarea integrităţii datelor a fost *replicarea*.

Metoda presupune generarea mai multor replici (clone) ale datelor iar integritatea acestora este validată prin compararea replicilor şi luarea unei decizii pe bază de vot majoritar. Dacă nu există erori în faza iniţială de replicare, orice modificare survenită la nivelul datelor poate fi detectată (dacă nu a fost realizată pe un număr majoritar de replici).

Sistemele de arhivare *PASIS* ([28]) şi *SafeStore* ([16]) folosesc această idee a descentralizării datelor. Un atacator trebuie să compromită un număr suficient de replici pentru a deveni o ameninţare reală.

Metodele bazate pe descentralizare şi replicare asigură foarte bine existenţa datelor arhivate şi accesul la ele, însă nu oferă mecanisme de protecţie faţă de învechirea algoritmilor criptografici. De exemplu, dacă arhiva conţine documente semnate acestea nu pot fi protejate pe termen lung. În plus, implementările pot deveni scumpe iar verificările sunt consumatoare de timp (dacă sistemul arhivează cantităţi mari de date).

Funcţiile de dispersie criptografică pot constitui o soluţie privind integritatea datelor. La introducerea unui document în arhivă este generată şi salvată şi amprenta acestuia. Verificarea integrităţii documentului presupune recalcularea acestei amprente şi confruntarea ei cu informaţia salvată iniţial.
Soluţiile *Plutus* ([17]) şi $SNAD$ ([23]) folosesc acest mecanism.

Tehnicile menţionate nu pot rezolva problemele privind integritatea documentelor pe termen lung. Ele pot fi însă folosite pentru a construi alte mecanisme mai sigure în acest sens.

O abordare posibilă foloseşte o combinare a mecanismelor de replicare cu cele de dispersie.
Fiecare obiect de date arhivat este replicat pe mai multe arhive. Verificarea integrităţii unui obiect se poate face prin calcularea amprentelor tuturor replicilor obiectului şi trimiterea lor către un auditor. Acesta va lua o decizie folosind o schemă simplă de vot majoritar.
Proiectul $LOCKSS$ ([24]) foloseşte un astfel de mecanism.

Deşi sunt mult mai costisitoare ca timp de calcul, semnăturile electronice constituie un mecanism puternic care poate fi folosit pentru garantarea integrităţii datelor.
Fiecare obiect arhivat poate fi semnat pe baza unei chei private a sistemului de arhivare. Verificarea integrităţii presupune validarea semnăturii respective pe baza certificatului digital asociat.

Problema principală este legată atât de rezistenţa în timp a algoritmilor de semnătură folosiţi, dar şi de valabilitatea certificatului folosit la verificare. În plus, compromiterea cheii private a sistemului de arhivare va compromite întreaga arhivă.

Formatele de semnătură electronică avansate propuse de $ETSI$ (*European Telecommunications Standards Institutei*) iau în calcul aspectul rezistenţei semnăturilor electronice în timp. Formele de semnătură $CAdES-A$ (ETSI 101 733) şi $XAdES-A$ (ETSI 101 903) conţin câmpuri speciale pentru arhivarea pe termen lung bazată pe marcarea temporală periodică.

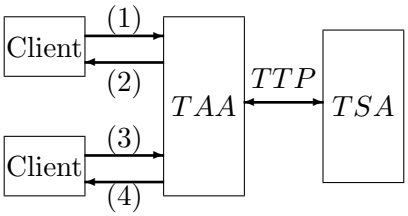
În vederea micsorării complexităţii operaţiilor de generare/verificare a informaţiei de control (dovezilor), semnăturile electronice pot fi înlocuite cu mecanisme de tip MAC, dar riscul privind compromiterea cheii secrete rămâne la fel de mare.

Trusted Archive Protocol (TAP)

TAP – *Trusted Archive Protocol* ([10]) este o propunere dezvoltată de grupul de lucru $ITEF-PKIX$ care foloseşte mărci temporale pentru asigurarea integrităţii datelor arhivate.

TAP introduce conceptul de *Autoritate de Arhivare de Încredere* (TAA – *Trust Archive Authority*) ca fiind componenta centrală a unui sistem de arhivare de încredere. TAA acceptă date (de orice tip) pentru arhivare şi asigură dovezile necesare pentru garantarea integrităţii acestora.

Modelul arhitectural propus de *Trusted Archive Protocol* este:



unde:

1. Clientul trimite o cerere de arhivare a datelor (inclusiv datele care trbuie arhivate).

2. TAA răspunde la cererea de arhivare, trimiţând – printre altele – şi un token de arhivare.

3. Clientul face o cerere de obţinere a datelor (incluzând tokenul de arhivare).

4. TAA răspunde la cerere, incluzând tokenul de arhivare, înregistrarea de arhivare şi datele solicitate din arhivă.

Modelul foloseşte schema TTP (RFC 3161) pentru obţinerea mărcilor temporale necesare în vederea garantării integrităţii datelor. Protocolul este de tip client - server sincron, iar tranzacţiile din cadrul protocolului sunt bazate pe mesaje CMS (RFC 3852). Toate răspunsurile primite din partea TAA sunt semnate electronic, fiind încapsulate într-o structură de tip *SignedData*.

Toate datele pentru structurile definite în TAP sunt descrise folosind notaţia $ASN.1$ (ISO 8824).

Datele arhivate sunt păstrate – împreună cu dovezile de arhivare – într-un *Pachet de Arhivare*. Acesta conţine:

- Datele primite de la client.

- Un token de arhivare, generat de TAA la momentul arhivării datelor. El este furnizat clientului ca răspuns la depunerea datelor, pentru a putea identifica şi realiza operaţii ulterioare asupra datelor arhivate (obţinerea, ştergerea datelor din arhivă etc.).

- Înregistrarea de arhivare, cu informaţia criptografică actualizată de TAA prin care se asigură integritatea datelor. Aceasta este o structură de mărci temporale imbricate. Iniţial structura conţine marca temporală calculată pentru datele transmise pentru arhivare. La fiecare actualizare, înregistrarea de arhivare cea mai recentă devine subiectul unei noi marcări temporale.

Protocolul *TAP* nu tratează decât câteva aspecte legate de arhivele pe termen lung. O serie de alte elemente – cum ar fi formatul datelor sau migrarea acestuia în timp – nu sunt acoperite de protocol. La nivelul protocolului nu pot fi definite roluri, atribuţii sau alte elemente care să permită implementarea unor operaţii de autentificare, autorizare, taxare a serviciului etc.
De asemenea propunerea de protocol nu conţine nici un suport pentru informaţii de tip meta-data (cuvinte cheie, categoria datelor, formatul datelor etc.). Acestea sunt esenţiale pentru gestiunea datelor în cadrul unui sistem de arhivare.

Schema folosită pentru gestiunea dovezilor de arhivare (înregistrările de arhivare) prezintă şi ea câteva probleme:

- Numărul de mărci temporale necesar este destul de mare.
  Pentru fiecare obiect de date, *TAA* trebuie să obţină cel puţin o marcă temporală iniţială (asociată cu datele arhivate), după care aceasta trebuie reînnoită ori de câte ori este nevoie. Astfel, dimensiunile structurilor cu înregistrările de arhivare pot creşte foarte mult.

- Folosirea schemei de marcare descrisă de *RFC* 3161 implică un nivel de încredere deplin în *Autoritatea de Marcare Temporală* (*TSA*) folosită.

  Ar trebui avute în vedere şi alte mecanisme de marcare temporală, mai sigure din acest punct de vedere, cum ar fi de exemplu cele bazate pe schemele înlănţuite.

- Garantarea integrităţii unui document semnat digital se bazează tot pe validarea unei semnături (cea din marca temporală).

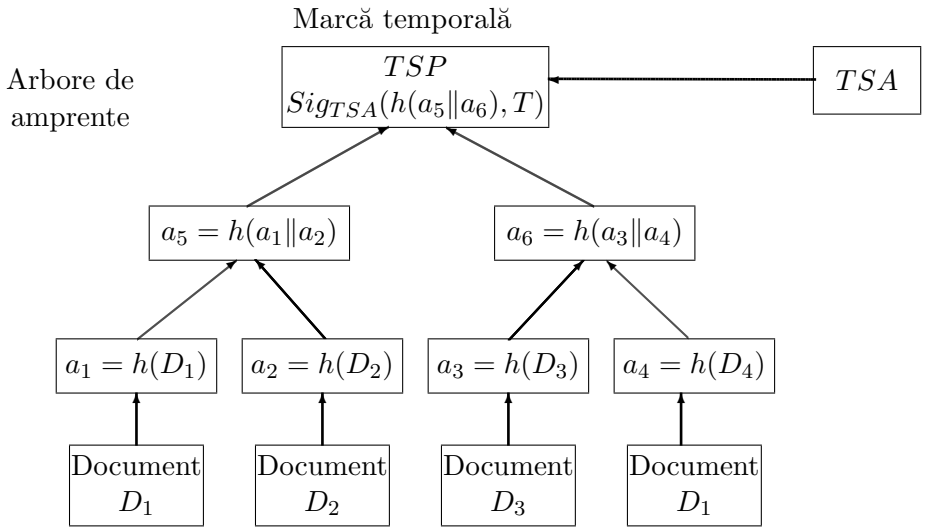
Evidence Record Syntax

Standardul Internet *ERS – Evidence Record Syntax* (*RFC* 4998) prezintă o propunere eficientă de generare a dovezilor necesare pentru garantarea integrităţii datelor într-un sistem de arhivare pe termen lung.

ERS se bazează pe conceptul de marcă temporală îmbunătăţită. Aceasta este o combinaţie realizată între arbori de dispersie şi o marcă temporală obişnuită. Fiecare arbore este construit pentru un grup de date arhivate. Frunzele arborilor conţin amprentele obiectelor de date arhivate, iar celelalte noduri conţin amprente ale informaţiei obţinute prin concatenarea dispersiilor descendenţilor. După construirea fiecărui arbore, marca temporală este aplicată numai nodului rădăcină.
În acest fel marca de timp nu protejează un singur obiect, ci un grup întreg de obiecte arhivate.

Pentru formatul mărcilor temporale se poate folosi sintaxa propusă în *RFC* 3161. Structurile de date astfel obţinute poartă numele de *mărci temporale îmbunătăţite*.
Modelul este exemplificat în figura de mai jos.

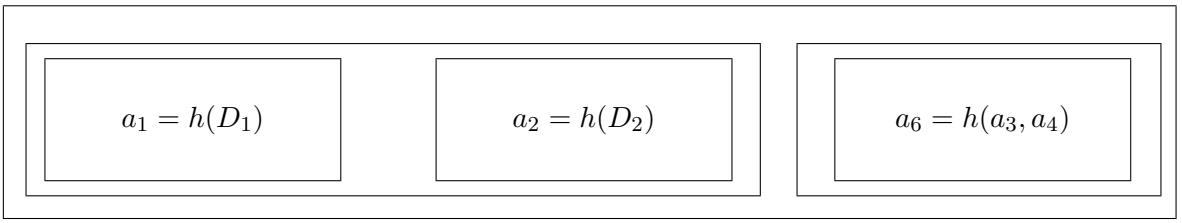
Verificarea unei mărci temporale îmbunătăţite pentru un obiect de date D_i presupune urmărirea drumului de la amprenta documentului verificat până la nodul rădăcină din arbore şi validarea mărcii temporale aplicate acestuia.



Astfel, pentru verificarea integrităţii documentului D_1 se va testa $h(h(a_1\|a_2), a_6) \in TSA$.

Arborele de amprente marcat temporal constituie punctul de pornire pentru generarea dovezilor.
Pentru fiecare document din arhivă nu se salvează întreg arborele, ci numai o formă redusă. Arborele redus pentru un document este de fapt o listă de liste cu amprente selectate din arborele iniţial, suficiente pentru a putea reface lanţul până la marca temporală.

De exemplu, pentru documentul D_1, arborele redus este



Periodic mărcile temporale şi arborii de dispersie trebuie regeneraţi (re-înnoiţi). Motivul este deprecierea în timp a proprietăţilor de securitate ai algoritmilor şi parametrilor folosiţi la generarea mărcilor de timp şi a nodurilor arborelului (funcţii de dispersie, algoritmi de semnătură, lungimi de chei).

Regenerarea se face doar pentru mărcile temporale aplicate nodurilor rădăcină sau pentru tot arborele de dispersie.

Reînnoirea mărcilor temporale se face în situaţia când algoritmul de semnătură sau cheia privată folosite de TSA devin slabe. Având în vedere că marca temporală veche

include deja toate amprentele documentelor din arbore, procesul de reînnoire este foarte eficient, iar documentele referite nu mai trebuie încărcate de pe mediile de stocare pentru a li se recalcula amprentele.
Sistemul de arhivare va calcula doar amprenta mărcii temporale vechi şi o va insera într-un arbore nou, pentru care va obţine o marcă temporală actualizată.

Reînnoirea unui arbore de amprente se face în situaţia când algoritmul de dispersie utilizat devine nesigur. Operaţia de înlocuire a funcţiei de dispersie este costisitoare, deoarece necesită refacerea amprentelor tuturor documentelor implicate.
Sistemul de arhivare trebuie să păstreze lanţul complet de dovezi generate pentru fiecare document de la momentul iniţial al depunerii sale în arhivă. Din acest motiv regenerarea fiecărui arbore va ţine cont atât de amprenta documentului cât şi de amprenta ultimului arbore redus calculat.

Pentru siguranţa sistemului de arhivare este recomandat să se genereze două structuri arborescente paralele, folosind funcţii de dispersie diferite (de exemplu $SHA-256$ şi $RIPEMD-160$). Dacă una din ele va fi compromisă, cealaltă oferă răgazul necesar pentru reînnoire.

Ştergerea unui document din arhivă nu influenţează restul arhivei. Toate celelalte documente pot fi verificate folosind pentru fiecare, arborele său redus.

Relativ la formatul datelor pentru structurile definite în ERS (RFC 4998), acestea sunt descrise folosind notaţia $ASN.1$ (ISO 8824).

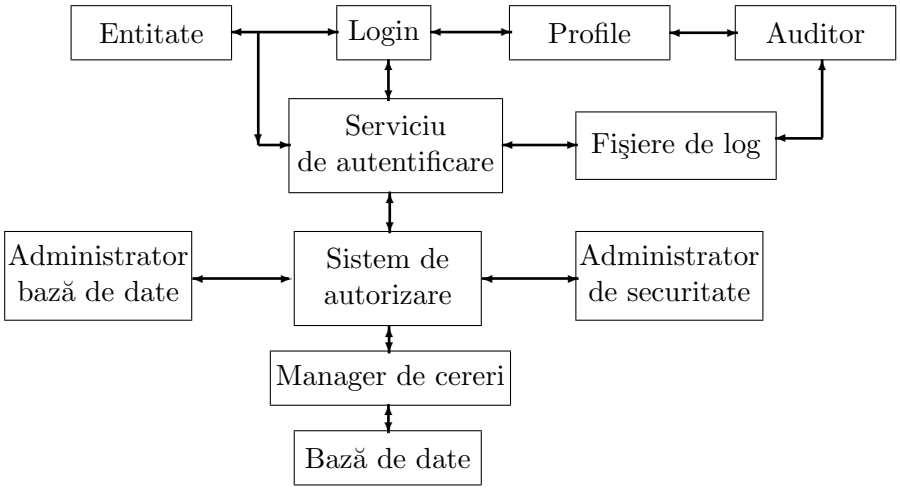
2.7 Servicii de Directoare

Există multe situaţii în care serviciile de securitate se bazează pe informaţii cum ar fi certificate de chei publice, liste de certificate revocate, certificate de atribute sau copii de registre electronice furnizate prin intermediul unor servicii de directoare.
Pentru accesarea acestor informaţii pot fi folosite metode de identificare unică şi clară a obiectelor din cadrul directoarelor.

Figura de mai jos (ISO 14516) prezintă arhitectura unui serviciu de directoare.

După logarea şi autentificarea cu succes, fiecare interogare pentru obţinerea de date din director este mediată de *Sistemul de Autorizare*. Dacă drepturile de acces ale unei entităţi sunt în concordanţă cu regulile de autorizare existente, se va obţine accesul.
Altfel, entitatea va primi un mesaj de eroare.
Încercările de acces eşuate (autorizări nereuşite) vor fi jurnalizate.

Managerul de cereri gestionează interogările autorizate. Scopul său este de a procesa interogarea, de a accesa baza de date şi de a întoarce răspunsul către entitate.



Nu este obligatoriu ca toate informaţiile să fie localizate într-o singură bază de date locală. Pentru administrarea securităţii serviciului de directoare pot fi definite următoarele roluri (ISO 14516):

1. Administratorul de securitate este responsabil pentru definirea regulilor de autorizare conforme cu politica de securitate adoptată.

   În acest sens, regulile de acces pot fi diferite, în funcţie de scopul serviciului de directoare (de exemplu, serviciul poate fi de natură publică sau poate fi restricţionat unei comunităţi controlate de entităţi care plătesc pentru acest serviciu).

2. Administratorul bazei de date este responsabil de întreţinerea acestei componente a serviciului de directoare. El are drepturi de acces la baza de date şi poate scrie, citi sau şterge informaţie din baza de date.

3. Auditorul revede periodic jurnalele din fişierele de loguri pentru a detecta posibilele intruziuni sau atacuri de securitate.

2.8 Servicii de Notariat Electronic

Serviciile publice de notariat electronic sunt servicii de nivel înalt care folosesc anumite servicii de bază: servicii de marcă temporală, de certificare, de directoare, de arhivare digitală şi non-repudiere etc.
Un document va fi trimis TTP-ului care-l va atesta (sau certifica) utilizând semnături digitale sau alte mecanisme. Un serviciu de directoare poate fi o componentă a serviciului de unde se pot obţine documentele certificate.

Un serviciu public de notariat poate atesta şi certifica anumite clase de documente (de exemplu faptul că un document a existat la un anumit moment de timp sau că este autentic). Un astfel de serviciu poate fi folosit pentru medierea unei dispute între două entităţi, fiind autorizat în acest sens de o anumită autoritate.

Serviciile de notariat electronic lucrează ca un notar public tradiţional. Ele pot stoca documente semnate digital şi marcate temporal. Aceste documente vor fi înregistrate într-un registru electronic propriu serviciului notarial.

Există însă multe aspecte delicate legate de aceste tipuri de servicii cum ar fi cele legate de autoritatea şi responsabilitatea notarului electronic, de evidenţă, jurisdicţie şi – mai ales – legale.

Definiţia 2.5. *O Autoritate Notarială (NA) este un terţ de încredere care înregistrează date la un anumit moment de timp şi poate verifica corectitudinea unei informaţii înregistrate în raport cu politica sa de securitate.*

Autoritatea Notarială acţionează ca un serviciu de înregistrare; acest rol poate fi însă extins pentru a funcţiona şi ca serviciu de validare. Atunci când Autoritatea Notarială face verificări asupra unui document ea poate adăuga informaţii noi la documentul respectiv. În acest mod entităţile care se încred în NA pot fi sigure că documentul a fost verificat de către Autoritate la un moment dat de timp şi este în concordanţă cu o anumită politică a Autorităţii Notariale.

Exemplul 2.5. *O Autoritate Notarială poate notariza un certificat digital de chei publice în raport cu o politică de validare.*

În acest caz NA verifică dacă certificatul din cerere este sau nu valid şi determină starea de revocare la momentul respectiv. Autoritatea verifică calea de certificare până la un punct de încredere, folosind mecanisme de validare bazate pe CRL-uri, ARL-uri sau pe servicii de validare on-line de tip $OCSP$ (RFC 2560) sau $SCVP$ (RFC 5055). Informaţiile privind starea certificatului se vor introduce împreună cu o marcă temporală de încredere într-un token notarial final.

Exemplul 2.6. *Autoritatea Notarială poate notariza şi date formatate. NA verifică corectitudinea datelor şi crează tokenul notarial.*

În acest caz, corectitudinea datelor nu se referă neapărat la valabilitatea unei semnături digitale. Particularitatea acestui aspect depinde de politica de verificare a Autorităţii Notariale şi de tipul documentului.

De exemplu, documentul poate conţine una sau mai multe semnături (în acest caz, corectitudinea documentului se bazează pe verificarea acestora), poate conţine presupuneri (corectitudinea se bazează pe valoarea lor de adevăr) sau documentul poate fi de fapt un contract (iar corectitudinea sa este legată de valabilitatea legală a documentului).

Autoritatea Notarială poate semna fiecare token notarial emis, folosind o cheie generată exclusiv în acest scop şi care are menţionată această calitate în certificatul de chei publice corespunzător.

2.9 Servicii de Gestiune a Cheilor

Un serviciu complet de gestiune a cheilor se bazează pe următoarele servicii primare (ISO 17770 − 1): generarea, înregistrarea, certificarea, distribuirea, instalarea, stocarea, derivarea, arhivarea, revocarea cheilor, precum şi scoaterea din evidenţă şi distrugerea cheilor.

În plus, se pot folosi şi alte servicii de securitate înrudite cum ar fi: controlul accesului, autentificarea, autorizarea sau marcarea temporală.

Un terţ de încredere on-line poate acţiona ca un serviciu de gestiune a cheilor, ca suport pentru serviciile care folosesc tehnici criptografice.
În funcţie de modul cum este generat materialul de cheie, serviciul poate acţiona ca un serviciu de distribuire de chei (când cheia este generată de TTP), sau un serviciu de translatare a cheilor (când cheia este generată de una dintre entităţi şi transmisă către celelaltor entităţi prin intermediul TTP-ului).

2.9.1 Serviciul de generare a cheilor

Este folosit pentru generarea într-o formă sigură a cheilor necesare anumitor algoritmi criptografici. Generarea de informaţie secretă şi/sau nepredictibilă, cu anumite proprietăţi, este fundamentală pentru generarea cheilor.

Obţinerea aleatorismului reprezintă una din componentele cele mai dificile ale unui sistem de generare de chei criptografice. Numerele aleatoare pot fi generate fie utilizând un generator de numere pseudo-aleatoare securizat criptografic sau o sursă cu adevărat aleatoare ($TRNG$ - *True Random Number Generator*).
Gestiunea numerelor aleatoare presupune generarea lor, validarea gradului de aleatorism, generarea şi validarea parametrilor de domeniu, generarea perechilor de chei şi validarea cheilor publice. Recomandări utile privind numerele aleatoare şi implicaţiile acestora în generarea de chei pot fi găsite în RFC 1750.

Atât pentru tehnicile simetrice cât si pentru cele asimetrice, trebuie avut în vedere următoarele aspecte:

- cheile posibil slabe pentru algoritmul ţintit;
- folosirea întregului spaţiu de chei.

2.9.2 Serviciul de înregistrare a cheilor

Aici TTP-ul este o autoritate de înregistrare acreditată, care realizează înregistrarea cheilor pentru entităţi. Fiecare cheie înregistrată este asociată cu o entitate.

Acest serviciu presupune menţinerea unui registru de chei si a informaţiilor asociate, într-o manieră securizată potrivită (de exemplu: un registru de chei publice pentru cheile publice ale entităţilor).

Cheile publice trebuie să fie certificate de una sau mai multe Autorităţi de Certificare. Pentru a creşte disponibilitatea acestui serviciu, cheile certificate ar trebui distribuite pe mai multe directoare accesibile şi de încredere – caz în care directoarele ar trebui să se reactualizeze pentru a păstra consistenţa.

Serviciile oferite de o autoritate de înregistrare sunt înregistrarea şi de-înregistrarea cheilor.

2.9.3 Serviciul de certificare a cheilor

În acest caz, TTP-ul este o Autoritate de Certificare acreditată care creează certificate de chei. CA-ul marchează temporal şi semnează cheile publice sau atribute, pentru a le valida şi autentifica într-o infrastructură de chei de încredere.

Entităţile care folosesc certificatele trebuie să aibă încredere într-o Autoritate de Certificare comună.

Cheile certificate pot fi generate fie de un serviciu de generare al TTP - ului fie de către entitatea utilizatoare. Serviciul poate include de asemenea reînnoirea certificatelor expirate (re-certificarea).

Sunt importante:

- obţinerea dovezii asupra posesiei cheii private din partea presupusului posesor;
- obţinerea dovezii asupra validităţii valorii cheii publice candidate (şi validitatea parametrilor – acolo unde este cazul).

2.9.4 Serviciul de distribuire a cheilor

Scopul este o distribuţie securizată a cheilor către entităţile autorizate.

În funcţie de politica de securitate a TTP-urilor, cheile pot fi trimise către alte servicii TTP, furnizate eventual de acelaşi terţ de încredere.

Distribuirea cheilor între TTP-uri şi între TTP-uri şi entităţi – mai ales dacă se folosesc canale nesecurizate – trebuie să folosească mecanisme şi tehnici criptografice de securizare.

Un tip mai special de serviciu de distribuire de chei îl reprezintă serviciul de translatare de chei. Rolul acestuia este acela de a translata cheile pentru ditribuţie între entităţi, astfel ca fiecare entitate să împartă în comun o cheie unică cu *Centrul de Translatare a Cheilor*.

2.9.5 Serviciul de stocare a cheilor

Furnizează stocarea securizată a cheilor utilizate pentru folosirea lor curentă sau pe termen scurt, sau pentru backup într-o locaţie fizică separată, pentru a asigura confidenţialitatea şi integritatea cheilor.

Este esenţial ca orice încercare de compromitere să fie detectată.

2.9.6 Serviciul de derivare a cheilor

Este utilizat pentru a crea un număr potenţial larg de chei (plecând de la o cheie iniţială secretă numită *cheie de derivare*), date variabile nesecrete şi un proces de transformare.

Cheia de derivare necesită mijloace speciale de protecţie, iar procesul de transformare trebuie să fie ireversibil şi nepredictibil – pentru ca în cazul compromiterii unei chei derivate să nu fie compromiee cheia de derivare sau alte chei derivate.

2.9.7 Serviciul de arhivare a cheilor

Acest serviciu este similar cu cel de stocare a cheilor, cu deosebirea că stocarea cheilor se face pe termen lung.

Serviciul este destinat pentru cheile care trebuie obţinute după o perioadă mare de timp pentru o anumită operaţie.

2.9.8 Serviciul de revocare a cheilor

Scopul acestui tip de serviciu este de a asigura dezactivarea securizată a cheilor – atunci când se ştie (sau se suspectează) că acele chei sunt compromise.

O listă de chei revocate ar trebui să fie distribuită regulat. Revocarea unei chei poate fi cerută de proprietarul cheii, de o altă persoană autorizată sau de o entitate de încredere – dacă există orice suspiciune că acea cheie ar putea fi compromisă.

Conform $ISO/IEC\ 11770-1$, fiecare intrare într-o listă de revocare a cheilor trebuie să includă timpul revocării, timpul cererii şi timpul la care cheia era cunoscută sau supectă de a fi compromisă.

În anumite cazuri, revocarea poate fi datorată anumitor restricţii de timp. Trebuie să fie un interval mic de timp între momentul cererii de revocare şi anunţarea listei de chei revocate. Un TTP poate fi responsabil numai cu revocarea cheilor pentru clienţii săi.

2.9.9 Serviciul de distrugere a cheilor

În acest caz, TTP-ul este o autoritate de înregistrare acreditată care furnizează servicii de distrugere a cheilor care nu mai sunt utile.

Acest TTP trebuie să furnizeze întâi un serviciu de scoatere din evidenţă, pentru a şterge asocierea dintre o cheie şi entitatea sa.
După acest pas, se trece la distrugerea efectivă a cheii.

Distrugerea cheii presupune distrugerea tuturor informaţiilor care ţin de ea, astfel încât să nu existe nici un mijloc de a recupera cheia respectivă. Acest lucru presupune şi distrugerea tuturor copiilor arhivate ale cheii – nu înainte de a verifica faptul că nu mai există informaţie arhivată protejată de această cheie.

Bibliografie

[1] A. Ansper, A. Buldas, M. Saarepera, J. Willemson – *Improving the availability of time-stamping services*, The 6th Australasian Conference on Information Security and Privacy ACISP, 2001.

[2] N. Asokan, M. Schunter, M. Waidner – *Optimistic protocols for fair exchange*, T. Matsumoto (Ed.), 4th ACM Conference on Computer and Communications Security, ACM Press, Zurich, 1997.

[3] N. Asokan, V. Shoup, M. Waidner – *Optimistic fair exchange of digital signatures*, Advances in Cryptology: Proc. of Eurocrypt 99, Vol. 1403 of LNCS, Springer Verlag, 1998.

[4] D. Bayer, S. A. Haber, W. S. Stornetta – *Improving the efficiency and reliability of digital time-stamping*, In Sequences 91: Methods in Communication, Security, and Computer Science, Springer-Verlag, 1992.

[5] J. Benaloh, M. De Mare – *Ecient Broadcast Time-Stamping*, Technical Report TR-MCS-91-1, Clarkson University, Department of Mathematics and Computer Science, April 1991.

[6] I. Bica – *Protocoale şi instrumente pentru confidenţialitatea şi autenticitatea documentelor în reţele de calculatoare*, Teză de doctorat, Academia Tehnică Militară, Bucureşti, 2004.

[7] A. Bonnecaze, P. Liardet, A. Gabillon, K. Blibech – *A Distributed Time Stamping Scheme*, Proc. of the IEEE Conference on Signal and Image Technology and Internet Based Systems, Cameroon, 2005.

[8] A. Buldas, P. Laud, H. Lipmaa, J. Villemson – *Time-stamping with binary linking schemes*, Advances in Cryptology CRYPTO 98, Springer-Verlag, August 1998.

[9] T. Coffey, P. Saidha – *Non-repudiation with mandatory proof of receipt*, ACMCCR: Computer Communication Review, 1996.

[10] S. Chokhani, C. Wallace – *Trusted Archive Protocol*, PKIX Internet Draft, February 2003.

[11] D. Lekkas, S. Katsikas, D. Spinellis, P. Gladychev, A. Patel – *User Requirements of Trusted Third Parties in Europe*. In Proc. of the Joint IFIP WG 8.5 and WG 9.6 Working Conference on User Identification & Privacy Protection, Stockholm, 1999

[12] S. A. Haber, W. S. Stornetta – *How to time-stamp a digital document*, Journal of Cryptology, Springer-Verlag, 1991.

[13] Y. Han – *Investigation of non-repudiation protocols*, ACISP: Information Security and Privacy: Australasian Conference, Vol 1172 of Lecture Notes in Computer Science, Springer-Verlag, 1996.

[14] S. Kremer, O. Markowitch – *Optimistic non-repudiable information exchange*, J. Biemond (Ed.), 21st Symp. On Information Theory in the Benelux, 2000

[15] S. Kremer, O. Markowitch, J. Zhou – *An intensive survey of fair nonrepudiation protocols*. Computer Communications 25, 2002.

[16] R. Kotla, M. Dahlin, L. Alvisi – *SafeStore: A durable and practical storage system*. In Proceedings of the 2007 USENIX Annual Technical Conference, USENIX, June 2007

[17] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, K. Fu – *Plutus: scalable secure file sharing on untrusted storage*. In Proc. of the Second USENIX Conference on File and Storage Technologies (FAST), San Francisco, CA, 2003.

[18] O. Markowitch, Y. Roggeman – *Probabilistic non-repudiation without trusted third party*, Proc. of 2nd Workshop on Security in Communication Networks, 1999.

[19] C. Marinescu, N. Ţapuş – *A Survey of the Problems of Time-Stamping or Why It Is Necessary to Have Another Time-Stamping Scheme*, Proc. of the IASTED Conference on Software Engineering 2007, SE2007, Austria.

[20] C. Marinescu – *Semnarea electronică a datelor. Schemă hibrid de realizare a ştampilelor digitale de timp*, Teză de doctorat, Universitatea Politehnică Bucureşti, 2008.

[21] S. Micali – *Certified E-mail with invisible post offices*, Invited presentation at the RSA 97 conference, 1997.

[22] J Mitsianis – *A new approach to enforcing non-repudiation of receipt*, Manuscript, 2001.

[23] E. L. Miller, D. D. E. Long, W. E. Freeman, B. C. Reed – *Strong security for network-attached storage*. In Proc. of the 2002 Conference on File and Storage Technologies (FAST), Monterey, CA, 2002.

[24] P. Maniatis, M. Roussopoulos, T. J. Giuli, D. S. H. Rosenthal, M. Baker – *The LOCKSS peer-to-peer digital preservation system*, ACM Trans. on Computer Systems (TOCS), Vol. 23, 2005.

[25] J. Onieva, J. Zhou, J. Lopez – *Multi-Party Non-repudiation: A Survey*, ACM Computing Surveys, 41(1), December 2008

[26] P. Syverson – *Weakly secret bit commitment: Applications to lotteries and fair exchange*, Proc. of the 1998 IEEE Computer Security Foundations Workshop (CSFW11), 1998.

[27] M. Togan – *Contribuţii privind dezvoltarea unor servicii de terţ de încredere în reţelele de calculatoare*, Teză de doctorat, Academia Tehnică Militară, Bucureşti, 2009.

[28] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R Ganger, H. Kiliccote, P. K. Khosla – *Survivable storage systems*, IEEE Computer, August 2000.

[29] K. Wouters, B. Preneel, A. I. Gonzlez-Tablas, A. Ribagorda – *Towards an XML format for time-stamps*, Proc. of the 2002 ACM Workshop on XML security, 2002.

[30] N. Zang, Q. Shi – *Achieving non-repudiation of receipt*, The Computer Jurnal, 1996.

[31] J. Zhou, R. Deng, F. Bao – *Evolution of fair non-repudiation with TTP*, ACISP: Information Security and Privacy: Australasian Conference, Vol. 1587 of LNCS, Springer-Verlag, 1999

[32] J. Zhou, D. Gollmann – *Observations on Non-repudiation*, Proc. of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology, Korea, 1996.