Servicii terțe de încredere (*TTP*)

Prof. Dr. Adrian Atanasiu

Universitatea București

March 1, 2016

Noțiuni introductive

Schimbul de informație electronică între două entități implică prezența unui element de încredere care să asigure unele servicii (ex: autenticitatea). Mai mult, uneori nivelul de încredere așteptat de entități nu se poate obține decât cu ajutorul unei părți terțe, care să faciliteze schimbul de informație.

Noțiuni introductive

Schimbul de informație electronică între două entități implică prezența unui element de încredere care să asigure unele servicii (ex: autenticitatea). Mai mult, uneori nivelul de încredere așteptat de entități nu se poate obține decât cu ajutorul unei părți terțe, care să faciliteze schimbul de informație.

Definiție

(X.509) O entitate are încredere în alta când are siguranța că partenerul se va comporta exact conform așteptărilor sale.
Un terț de încredere (TTP - Trusted Third Party) este o entitate specifică care furnizează unul sau mai multe servicii electronice și este considerată de încredere de către celelalte entități.

Cerințe funcționale

- **Autentificarea**: Identificarea corectă a entităților implicate în tranzacțiile electronice.

Cerințe funcționale

- **Autentificarea**: Identificarea corectă a entităților implicate în tranzacțiile electronice.
  Se obține în general utilizând mecanisme de criptografie cu chei publice și 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*).

Cerințe funcționale

- **Autentificarea**: Identificarea corectă a entităților implicate în tranzacțiile electronice.
  Se obține în general utilizând mecanisme de criptografie cu chei publice și 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*).
- **Integritatea datelor**: Păstrarea lor nealterată pe timpul comunicării între entități.
  Alterarea poate fi accidentală sau intenționată.

Cerințe funcționale

- **Autentificarea**: Identificarea corectă a entităților implicate în tranzacțiile electronice.
  Se obține în general utilizând mecanisme de criptografie cu chei publice și 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*).
- **Integritatea datelor**: Păstrarea lor nealterată pe timpul comunicării între entități.
  Alterarea poate fi accidentală sau intenționată.
  Integritatea se poate realiza utilizând mecanisme de semnătură electronică sau funcții de dispersie.

- **Confidențialitatea**: Criptarea mesajelor schimbate între entități în cadrul tranzacțiilor electronice.
  Se obține utilizând în general algoritmi de criptare cu chei simetrice, iar în unele situații – mecanisme de criptare asimetrică.

- **Confidențialitatea**: Criptarea mesajelor schimbate între entități în cadrul tranzacțiilor electronice.
  Se obține utilizând în general algoritmi de criptare cu chei simetrice, iar în unele situații – mecanisme de criptare asimetrică.
- **Non-repudierea**: O entitate nu poate nega o acțiune (cum ar fi expedierea sau recepționarea unui mesaj) sau existența unor informații la un moment dat.
  Această cerință poate fi asigurată cu tehnici de semnătură digitală (non-repudierea originii mesajelor) sau de marcare temporală (existența datelor la un anumit moment).

- **Confidențialitatea**: Criptarea mesajelor schimbate între entități în cadrul tranzacțiilor electronice.
  Se obține utilizând în general algoritmi de criptare cu chei simetrice, iar în unele situații – mecanisme de criptare asimetrică.
- **Non-repudierea**: O entitate nu poate nega o acțiune (cum ar fi expedierea sau recepționarea unui mesaj) sau existența unor informații la un moment dat.
  Această cerință poate fi asigurată cu tehnici de semnătură digitală (non-repudierea originii mesajelor) sau de marcare temporală (existența datelor la un anumit moment).
- **Disponibilitatea**: Este corelată cu politica *TTP*-ului și *SLA*-ului (*Service Level Agreament*) acceptat.

- **Confidențialitatea**: Criptarea mesajelor schimbate între entități în cadrul tranzacțiilor electronice.
  Se obține utilizând în general algoritmi de criptare cu chei simetrice, iar în unele situații – mecanisme de criptare asimetrică.
- **Non-repudierea**: O entitate nu poate nega o acțiune (cum ar fi expedierea sau recepționarea unui mesaj) sau existența unor informații la un moment dat.
  Această cerință poate fi asigurată cu tehnici de semnătură digitală (non-repudierea originii mesajelor) sau de marcare temporală (existența datelor la un anumit moment).
- **Disponibilitatea**: Este corelată cu politica *TTP*-ului și *SLA*-ului (*Service Level Agreament*) acceptat.
  Este asigurată prin mecanisme specifice de *HA* (*High - Availability*) și *DR* (*Disaster - Recovery*).

- **Ușurința de utilizare**: Interfața sistemului cu utilizatorii este importantă dacă interacționează direct cu aceștia.

- **Ușurința de utilizare**: Interfața sistemului cu utilizatorii este importantă dacă interacționează direct cu aceștia.
  Unele servicii oferite de *TTP* interacționează doar prin intermediul unor aplicații care implementează protocoale specifice.

- **Ușurința de utilizare**: Interfața sistemului cu utilizatorii este importantă dacă interacționează direct cu aceștia.
  Unele servicii oferite de *TTP* interacționează doar prin intermediul unor aplicații care implementează protocoale specifice.
- **Mobilitatea**: Necesară – în unele situații – pentru utilizatorii mobili.
  Aceștia trebuie să poată contacta un anumit *TTP* indiferent de localizarea lor în raport cu acesta.

- **Ușurința de utilizare**: Interfața sistemului cu utilizatorii este importantă dacă interacționează direct cu aceștia.
  Unele servicii oferite de *TTP* interacționează doar prin intermediul unor aplicații care implementează protocoale specifice.
- **Mobilitatea**: Necesară – în unele situații – pentru utilizatorii mobili.
  Aceștia trebuie să poată contacta un anumit *TTP* indiferent de localizarea lor în raport cu acesta.
- **Anonimitatea**: O entitate poate fi înregistrată la un *TTP*, însă (în funcție de opțiunile sale) identitatea sa trebuie să nu fie dezvăluită celorlaltor utilizatori.

- **Ușurința de utilizare**: Interfața sistemului cu utilizatorii este importantă dacă interacționează direct cu aceștia.
  Unele servicii oferite de *TTP* interacționează doar prin intermediul unor aplicații care implementează protocoale specifice.
- **Mobilitatea**: Necesară – în unele situații – pentru utilizatorii mobili.
  Aceștia trebuie să poată contacta un anumit *TTP* indiferent de localizarea lor în raport cu acesta.
- **Anonimitatea**: O entitate poate fi înregistrată la un *TTP*, însă (în funcție de opțiunile sale) identitatea sa trebuie să nu fie dezvăluită celorlaltor utilizatori.
- **Marcarea temporală**: Mărcile temporale sigure (de încredere) atașate documentelor electronice sunt necesare în anumite tranzacții desfășurate între entități.

- **Unicitatea**: Unicitatea unor documente/mesaje poate fi o cerință care trebuie îndeplinită de un *TTP*.

- **Unicitatea**: Unicitatea unor documente/mesaje poate fi o cerință care trebuie îndeplinită de un *TTP*.
- **Inter-operabilitatea**: Schimbul mesajelor nu poate fi restricționat la domeniul deservit de un singur *TTP*.

- **Unicitatea**: Unicitatea unor documente/mesaje poate fi o cerință care trebuie îndeplinită de un *TTP*.
- **Inter-operabilitatea**: Schimbul mesajelor nu poate fi restricționat la domeniul deservit de un singur *TTP*.
  În unele situații, mesajele electronice trebuie procesate de utilizatori din domenii deservite de *TTP*-uri diferite.
  De exemplu, cross - certificarea a două *CA* - uri din domenii *PKI* diferite poate fi o condiție necesară și suficientă pentru asigurarea inter-operabilității utilizatorilor acelor domenii.

- **Unicitatea**: Unicitatea unor documente/mesaje poate fi o cerință care trebuie îndeplinită de un *TTP*.
- **Inter-operabilitatea**: Schimbul mesajelor nu poate fi restricționat la domeniul deservit de un singur *TTP*.
  În unele situații, mesajele electronice trebuie procesate de utilizatori din domenii deservite de *TTP*-uri diferite.
  De exemplu, cross - certificarea a două *CA* - uri din domenii *PKI* diferite poate fi o condiție necesară și suficientă pentru asigurarea inter-operabilității utilizatorilor acelor domenii.
  Tot inter-operabilitatea presupune și respectarea standardelor care guvernează domeniul de aplicabilitate.

- **Unicitatea**: Unicitatea unor documente/mesaje poate fi o cerință care trebuie îndeplinită de un *TTP*.
- **Inter-operabilitatea**: Schimbul mesajelor nu poate fi restricționat la domeniul deservit de un singur *TTP*.
  În unele situații, mesajele electronice trebuie procesate de utilizatori din domenii deservite de *TTP*-uri diferite.
  De exemplu, cross - certificarea a două *CA* - uri din domenii *PKI* diferite poate fi o condiție necesară și suficientă pentru asigurarea inter-operabilității utilizatorilor acelor domenii.
  Tot inter-operabilitatea presupune și respectarea standardelor care guvernează domeniul de aplicabilitate.
- **Acreditarea**: Procedurile de auditare si acreditare pentru *TTP*-uri sunt esențiale pentru asigurarea unui nivel de încredere cerut în relațiile cu utilizatorii.
  Se poate face la nivel local, național sau internațional.

- **Politica de securitate**: Fiecare *TTP* trebuie să ofere utilizatorilor săi o politică de securitate bine definită, în concordanță cu legislația și restricțiile naționale precum și cu cerințele de securitate definite.
  Disputele dintre entități vor fi arbitrate în concordanță și cu politicile de securitate definite la nivelul *TTP*-urilor implicate (direct sau indirect) în tranzacțiile electronice.

- **Politica de securitate**: Fiecare *TTP* trebuie să ofere utilizatorilor săi o politică de securitate bine definită, în concordanță cu legislația și restricțiile naționale precum și cu cerințele de securitate definite.
  Disputele dintre entități vor fi arbitrate în concordanță și cu politicile de securitate definite la nivelul *TTP*-urilor implicate (direct sau indirect) în tranzacțiile electronice.
- **Managementul cheilor**: Utilizatorii pot cere unui *TTP* diverse servicii de gestiune privind cheile lor de semnătură și criptare.

- **Politica de securitate**: Fiecare *TTP* trebuie să ofere utilizatorilor săi o politică de securitate bine definită, în concordanță cu legislația și restricțiile naționale precum și cu cerințele de securitate definite.
  Disputele dintre entități vor fi arbitrate în concordanță și cu politicile de securitate definite la nivelul *TTP*-urilor implicate (direct sau indirect) în tranzacțiile electronice.
- **Managementul cheilor**: Utilizatorii pot cere unui *TTP* diverse servicii de gestiune privind cheile lor de semnătură și criptare.
  Generarea cheilor, distribuția 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 cerințe funcționale la nivelul unui *TTP*.

- **Publicarea datelor**: Serviciile de publicare din *TTP*-uri sunt necesare pentru afișarea unor informații folosite de utilizatorii sistemelor: distribuirea cheilor publice sau a certificatelor digitale pentru utilizatori, distribuirea listelor de certificate revocate.

- **Publicarea datelor**: Serviciile de publicare din *TTP*-uri sunt necesare pentru afișarea unor informații folosite de utilizatorii sistemelor: distribuirea cheilor publice sau a certificatelor digitale pentru utilizatori, distribuirea listelor de certificate revocate.
- **Scalabilitatea și modularitatea**: Serviciile furnizate de *TTP*-uri trebuie să fie scalabile și ușor gestionabile în implementări pe scară largă.
  Structura modularizată permite adăugarea sau scoaterea cu ușurință a unor componente de funcționalitate la nivelul *TTP*-ului.

- **Publicarea datelor**: Serviciile de publicare din *TTP*-uri sunt necesare pentru afișarea unor informații folosite de utilizatorii sistemelor: distribuirea cheilor publice sau a certificatelor digitale pentru utilizatori, distribuirea listelor de certificate revocate.
- **Scalabilitatea și modularitatea**: Serviciile furnizate de *TTP*-uri trebuie să fie scalabile și ușor gestionabile în implementări pe scară largă.
  Structura modularizată permite adăugarea sau scoaterea cu ușurință a unor componente de funcționalitate la nivelul *TTP*-ului.
- **Compatibilitatea și portabilitatea**: Presupune compatibilitatea implementărilor *TTP*-urilor cu standardele, tehnologiile și platformele software/hardware cerute.

Configurație cu *TTP* in-line

Din punct de vedere al poziționării terțului de încredere în cadrul comunicației dintre entitățile client, precum și al implicării sale directe sau indirecte la protocolul de comunicație, se pot realiza trei configurații diferite:

Configurație cu *TTP* in-line

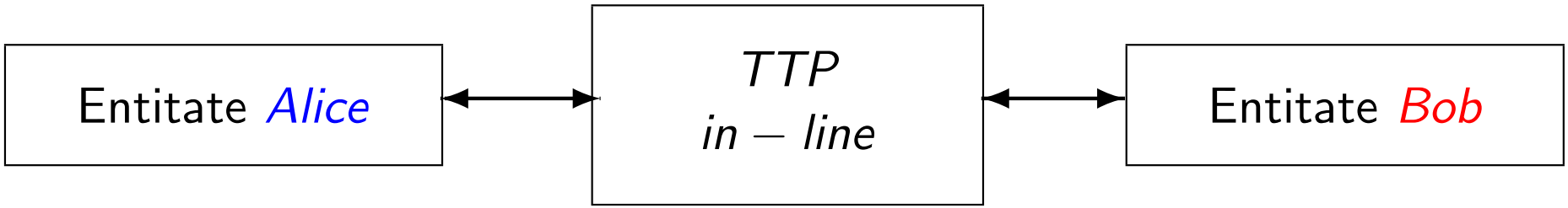
Din punct de vedere al poziționării terțului de încredere în cadrul comunicației dintre entitățile client, precum și al implicării sale directe sau indirecte la protocolul de comunicație, se pot realiza trei configurații diferite:

Două entități aparținând la două domenii de securitate diferite doresc să schimbe între ele mesaje securizate.

Configurație cu *TTP* in-line

Din punct de vedere al poziționării terțului de încredere în cadrul comunicației dintre entitățile client, precum și al implicării sale directe sau indirecte la protocolul de comunicație, se pot realiza trei configurații diferite:

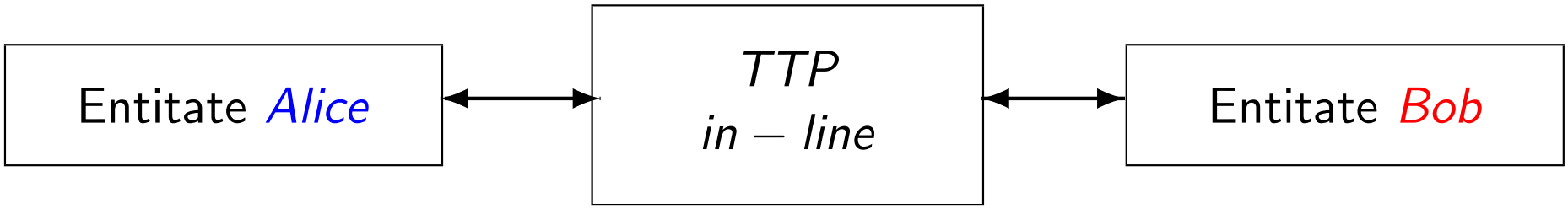
Două entități aparținând la două domenii de securitate diferite doresc să schimbe între ele mesaje securizate.



Configurație cu *TTP* in-line

Din punct de vedere al poziționării terțului de încredere în cadrul comunicației dintre entitățile client, precum și al implicării sale directe sau indirecte la protocolul de comunicație, se pot realiza trei configurații diferite:

Două entități aparținând la două domenii de securitate diferite doresc să schimbe între ele mesaje securizate.



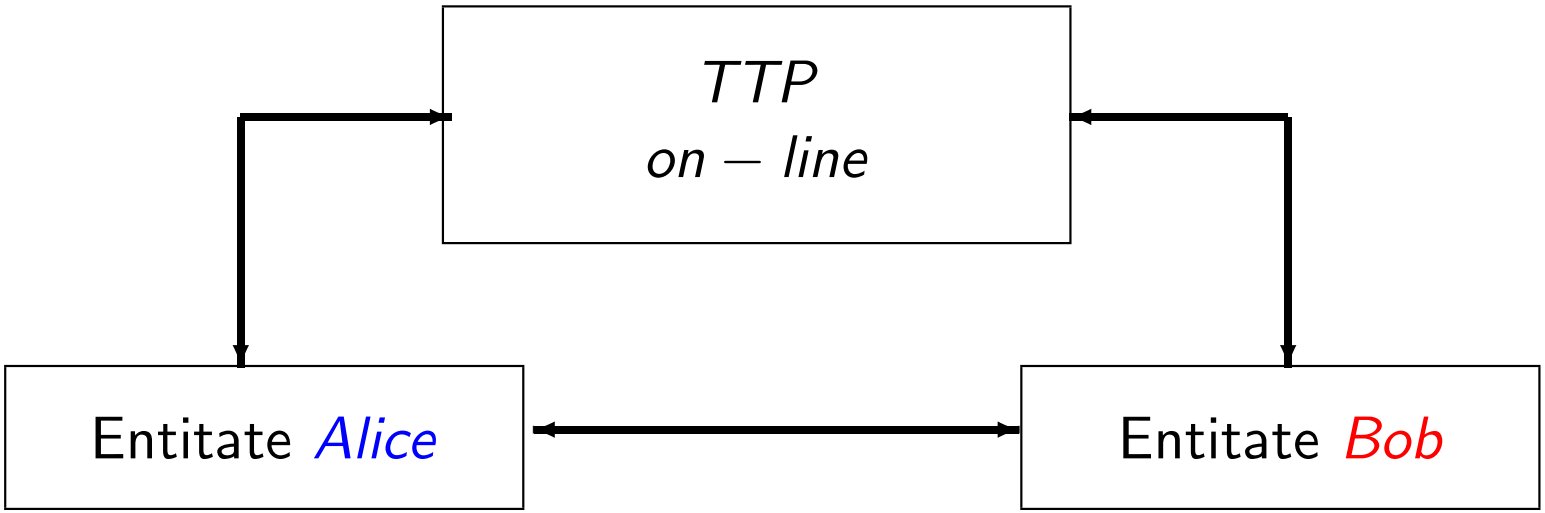
Pot include: autentificarea sau controlul privilegiilor, controlul accesului, recuperarea de chei, confidențialitate și integritate pentru datele transmise.

Configurație cu *TTP* on-line

TTP-ul este implicat doar în câteva din schimburile securizate dintre ele (de obicei în prima fază a comunicației).

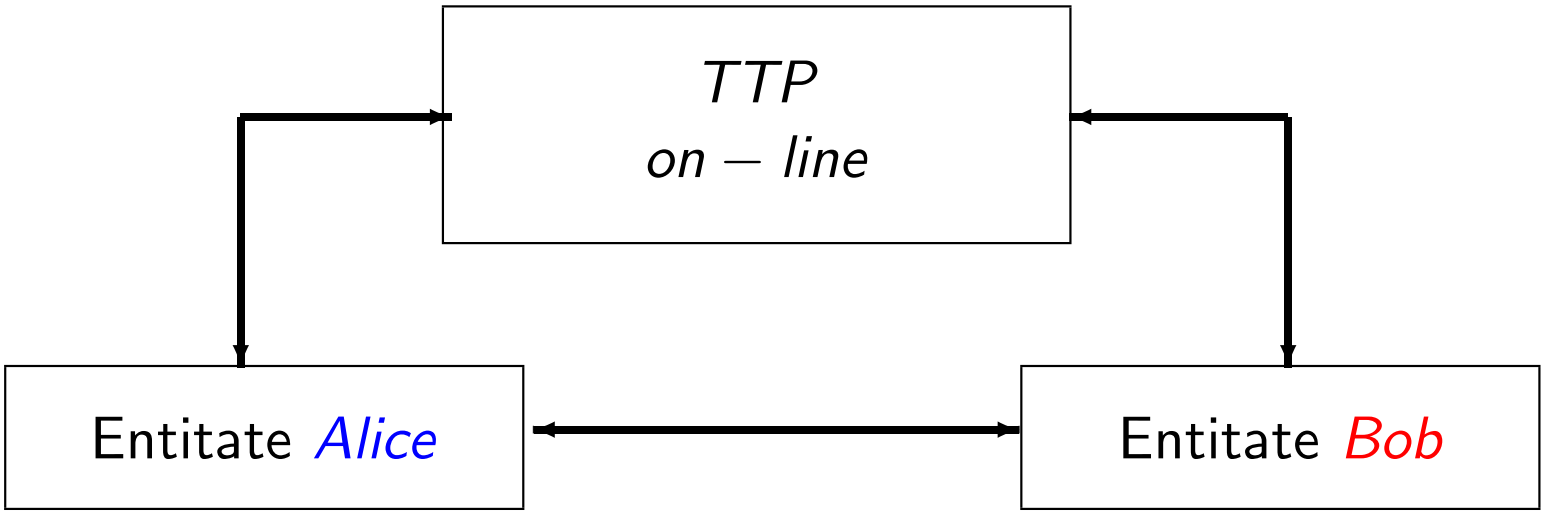
Configurație cu *TTP* on-line

TTP-ul este implicat doar în câteva din schimburile securizate dintre ele (de obicei în prima fază a comunicației).



Configurație cu *TTP* on-line

TTP-ul este implicat doar în câteva din schimburile securizate dintre ele (de obicei în prima fază a comunicației).



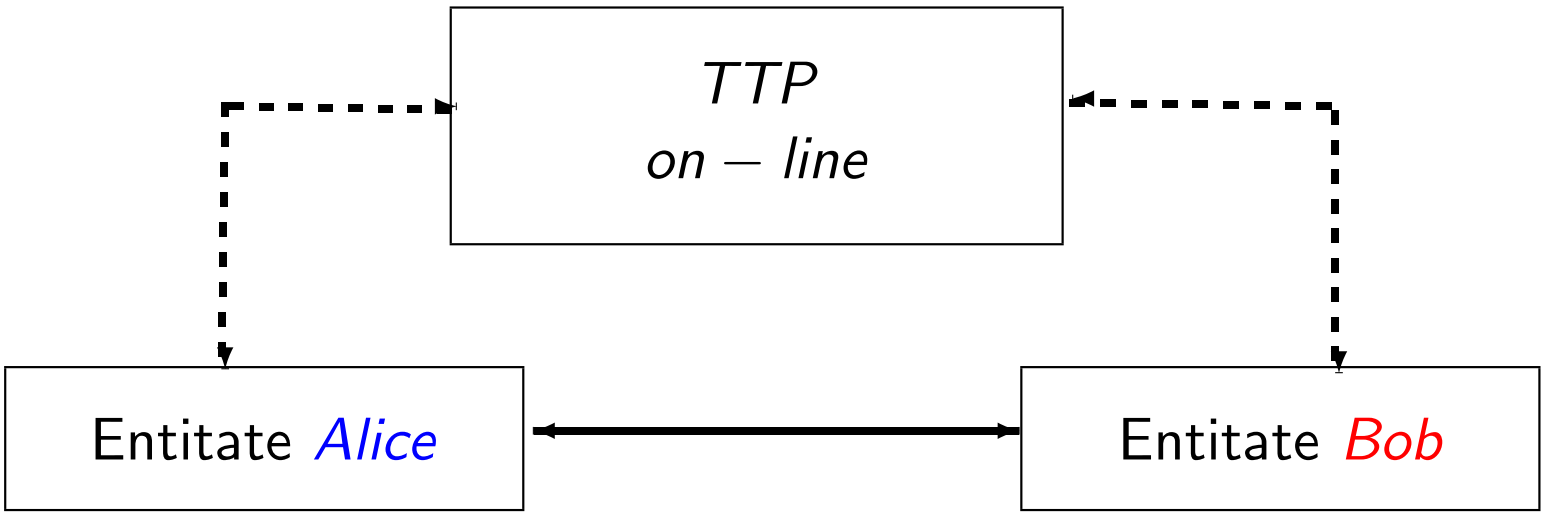
Pot include servicii de autentificare, certificare, controlul privilegiilor, non-repudiere, controlul accesului, managementul cheilor, marcare temporală, confidențialitate și integritate.

Configurație cu *TTP* off-line

Terțul de încredere off-line nu interacționează direct cu entitățile în timpul schimburilor de mesaje securizate. În schimb, entitățile comunică între ele folosind date generate anterior de către *TTP*.

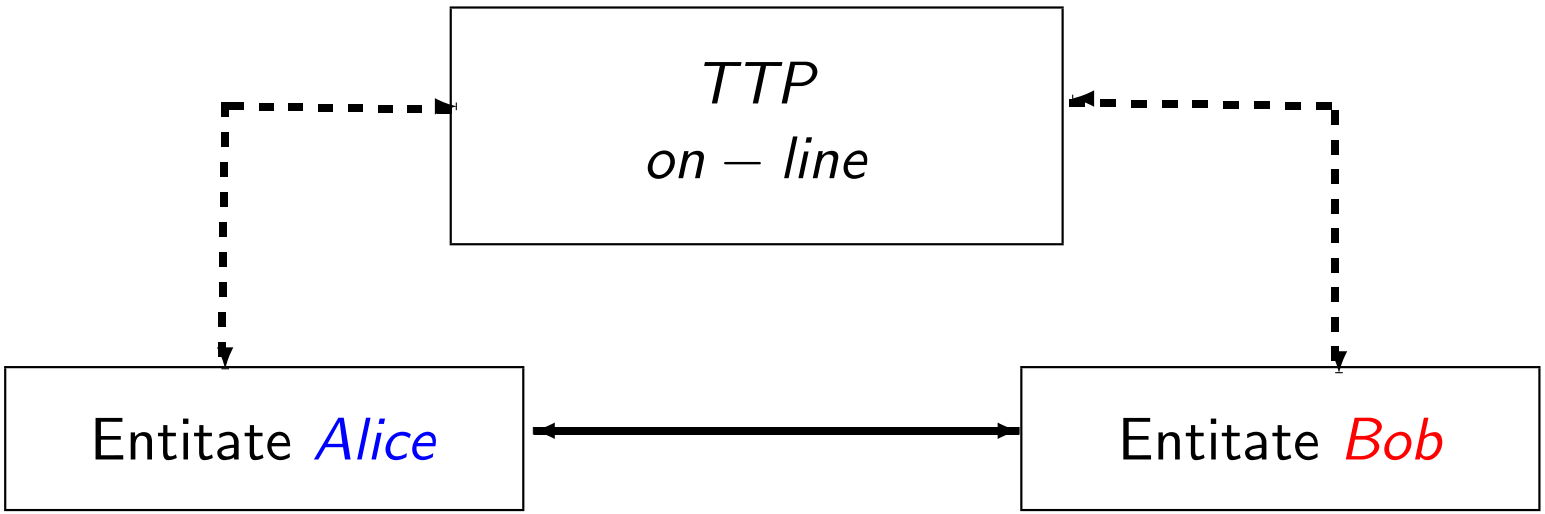
Configurație cu *TTP* off-line

Terțul de încredere off-line nu interacționează direct cu entitățile în timpul schimburilor de mesaje securizate. În schimb, entitățile comunică între ele folosind date generate anterior de către *TTP*.



Configurație cu *TTP* off-line

Terțul de încredere off-line nu interacționează direct cu entitățile în timpul schimburilor de mesaje securizate. În schimb, entitățile comunică între ele folosind date generate anterior de către *TTP*.



Pot include servicii de autentificare, certificare, controlul privilegiilor, non-repudiere, distribuirea cheilor, recuperarea cheilor etc.

Un *TTP* poate avea acorduri de încredere stabilite cu alte *TTP*-uri pentru a forma o rețea care să permită entităților sale să comunice securizat cu entitățile acestea.

Un *TTP* poate avea acorduri de încredere stabilite cu alte *TTP*-uri pentru a forma o rețea care să permită entităților sale să comunice securizat cu entitățile acestea.
Fiecare terț de încredere furnizează servicii către entitățile din domeniul său conform cu politica de securitate proprie.

Un *TTP* poate avea acorduri de încredere stabilite cu alte *TTP*-uri pentru a forma o rețea care să permită entităților sale să comunice securizat cu entitățile acestea.
Fiecare terț de încredere furnizează servicii către entitățile din domeniul său conform cu politica de securitate proprie.
Posibilități de interconectare:

- Interconectare *TTP - Utilizator*: presupune mecanisme și mijloace prin care un utilizator interacționează cu un *TTP* pentru a cere/primi un serviciu.
  Fiecare utilizator poate interacționa cu un *TTP* în mod diferit, în funcție de serviciul solicitat.

Interconectare *Utilizator - Utilizator*

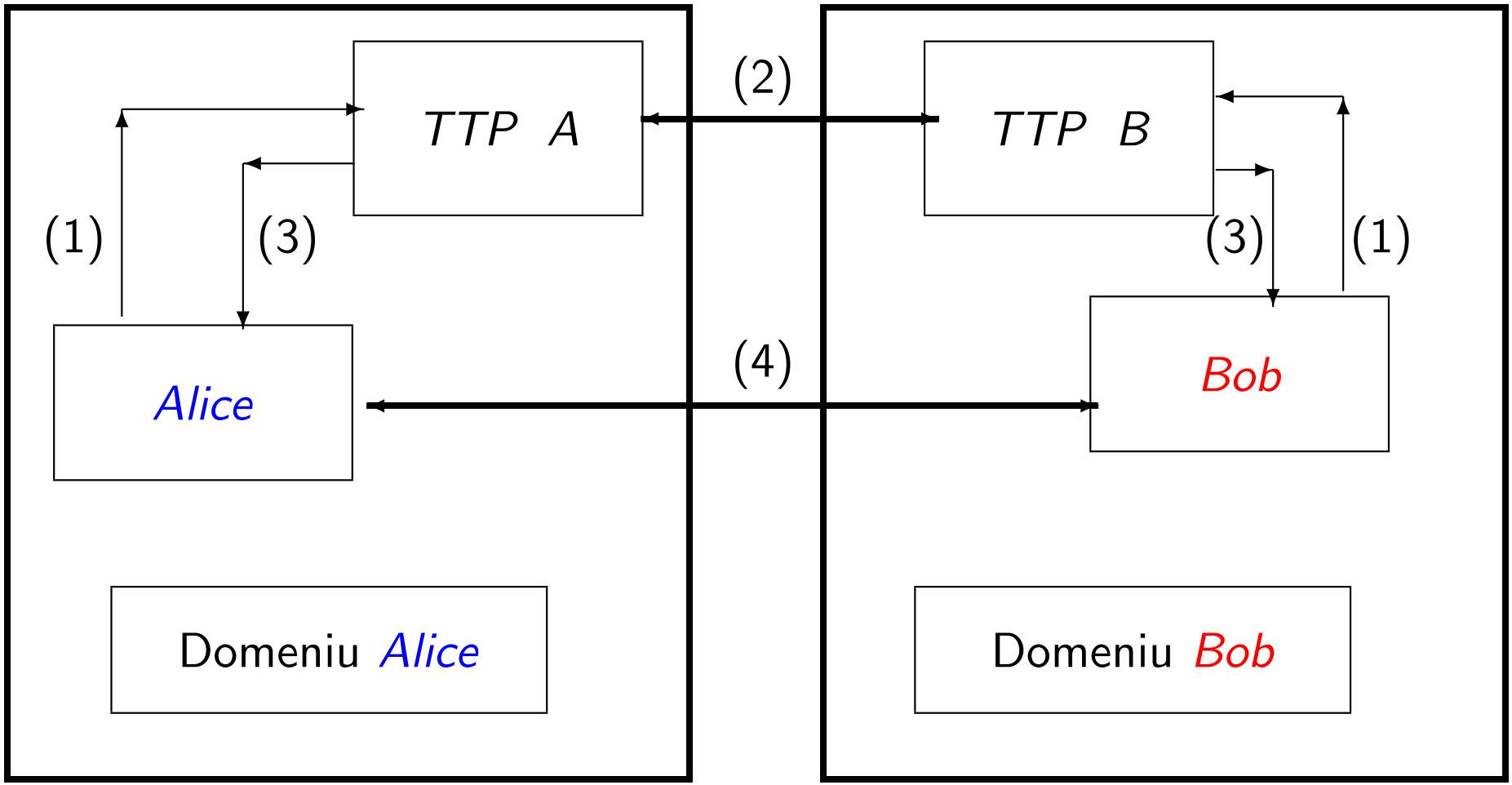
După ce un *TTP* și-a terminat joburile în relația cu entitățile sale, toate comunicațiile dintre entități se pot face fără a mai fi nevoie de asistența *TTP*-ului.

Interconectare *Utilizator - Utilizator*

După ce un *TTP* și-a terminat joburile în relația cu entitățile sale, toate comunicațiile dintre entități se pot face fără a mai fi nevoie de asistența *TTP*-ului.

Relația între entități, precum și formalizarea contractuală a acestei relații, se bazează pe încrederea acestora în *TTP* și pe mecanismele de interconectare dintre *TTP*-uri.

Interconectare *TTP* - *TTP*



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

1. *Alice* cere de la *TTP*-ul *A* o cheie secretă pentru a comunica cu *Bob* (1).
2. *TTP*-ul *A* transferă cheia secretă către *Alice* (3) și către *TTP*-ul *B* (2).

1. *Alice* cere de la *TTP*-ul *A* o cheie secretă pentru a comunica cu *Bob* (1).
2. *TTP*-ul *A* transferă cheia secretă către *Alice* (3) și către *TTP*-ul *B* (2).
3. *TTP*-ul *B* transmite apoi cheia către *Bob* (3).

1. *Alice* cere de la *TTP*-ul *A* o cheie secretă pentru a comunica cu *Bob* (1).
2. *TTP*-ul *A* transferă cheia secretă către *Alice* (3) și către *TTP*-ul *B* (2).
3. *TTP*-ul *B* transmite apoi cheia către *Bob* (3).
4. Având stabilită o cheie comună, cele două entități pot comunica securizat (4).

1. *Alice* cere de la *TTP*-ul *A* o cheie secretă pentru a comunica cu *Bob* (1).
2. *TTP*-ul *A* transferă cheia secretă către *Alice* (3) și către *TTP*-ul *B* (2).
3. *TTP*-ul *B* transmite apoi cheia către *Bob* (3).
4. Având stabilită o cheie comună, cele două entități pot comunica securizat (4).

Variantă care utilizează tehnologia cu chei publice:

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

1. *Alice* cere de la *TTP*-ul *A* o cheie secretă pentru a comunica cu *Bob* (1).
2. *TTP*-ul *A* transferă cheia secretă către *Alice* (3) și către *TTP*-ul *B* (2).
3. *TTP*-ul *B* transmite apoi cheia către *Bob* (3).
4. Având stabilită o cheie comună, cele două entități pot comunica securizat (4).

Variantă care utilizează tehnologia cu chei publice:

1. *Alice* cere *TTP*-ului *A* un certificat pentru a comunica securizat cu *Bob* (1).
2. *TTP*-ul *A* emite un certificat lui *Alice* (3).

1. *Alice* cere de la *TTP*-ul *A* o cheie secretă pentru a comunica cu *Bob* (1).
2. *TTP*-ul *A* transferă cheia secretă către *Alice* (3) și către *TTP*-ul *B* (2).
3. *TTP*-ul *B* transmite apoi cheia către *Bob* (3).
4. Având stabilită o cheie comună, cele două entități pot comunica securizat (4).

Variantă care utilizează tehnologia cu chei publice:

1. *Alice* cere *TTP*-ului *A* un certificat pentru a comunica securizat cu *Bob* (1).
2. *TTP*-ul *A* emite un certificat lui *Alice* (3).
3. Analog, *Bob* cere către *TTP*-ul *B* emiterea unui certificat (1) și-l obține (3).

1. *Alice* cere de la *TTP*-ul *A* o cheie secretă pentru a comunica cu *Bob* (1).
2. *TTP*-ul *A* transferă cheia secretă către *Alice* (3) și către *TTP*-ul *B* (2).
3. *TTP*-ul *B* transmite apoi cheia către *Bob* (3).
4. Având stabilită o cheie comună, cele două entități pot comunica securizat (4).

Variantă care utilizează tehnologia cu chei publice:

1. *Alice* cere *TTP*-ului *A* un certificat pentru a comunica securizat cu *Bob* (1).
2. *TTP*-ul *A* emite un certificat lui *Alice* (3).
3. Analog, *Bob* cere către *TTP*-ul *B* emiterea unui certificat (1) și-l obține (3).
4. De asemenea, cele două *TTP*-uri își emit una către cealaltă câte un certificat (2).

Non-Repudiere

Necesitatea serviciilor de non-repudiere nu provine numai din faptul că părțile pot încerca să se înșele una pe cealaltă.

Non-Repudiere

Necesitatea serviciilor de non-repudiere nu provine numai din faptul că părțile pot încerca să se înșele una pe cealaltă.

Diverse circumstanțe neașteptate pot conduce la situația în care două entități implicate într-o tranzacție ajung – în timp – să aibă puncte diferite de vedere cu privire la ce s-a întâmplat în cadrul relației dintre ele.

Non-Repudiere

Necesitatea serviciilor de non-repudiere nu provine numai din faptul că părțile pot încerca să se înșele una pe cealaltă.

Diverse circumstanțe neașteptate pot conduce la situația în care două entități implicate într-o tranzacție ajung – în timp – să aibă puncte diferite de vedere cu privire la ce s-a întâmplat în cadrul relației dintre ele.

O eroare de comunicație pe rețea în timpul derulării unui protocol este un exemplu reprezentativ.

O tranzacție de bază este transferarea unui mesaj M de la *Alice* (Emitent) la *Bob* (Receptor):

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

O tranzacție de bază este transferarea unui mesaj M de la *Alice* (Emitent) la *Bob* (Receptor):

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

Chiar și într-o astfel de tranzacție simplă, ar putea apare următoarele cazuri de dispută:

- *Alice* susține că a trimis mesajul M lui *Bob*, iar *Bob* acuză faptul că nu l-a primit;

O tranzacție de bază este transferarea unui mesaj M de la *Alice* (Emitent) la *Bob* (Receptor):

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

Chiar și într-o astfel de tranzacție simplă, ar putea apare următoarele cazuri de dispută:

- *Alice* susține că a trimis mesajul M lui *Bob*, iar *Bob* acuză faptul că nu l-a primit;
- *Bob* susține că a primit mesajul M de la *Alice*, iar *Alice* acuză faptul că nu l-a trimis;

O tranzacție de bază este transferarea unui mesaj M de la *Alice* (Emitent) la *Bob* (Receptor):

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

Chiar și într-o astfel de tranzacție simplă, ar putea apare următoarele cazuri de dispută:

- *Alice* susține că a trimis mesajul M lui *Bob*, iar *Bob* acuză faptul că nu l-a primit;
- *Bob* susține că a primit mesajul M de la *Alice*, iar *Alice* acuză faptul că nu l-a trimis;
- *Alice* susține 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 entitățile implicate într-o tranzacție să rezolve posibilele dispute care pot apare cu privire la anumite evenimente sau acțiuni care s-au întâmplat (sau nu s-au întâmplat) în cadrul tranzacției.

Serviciile de non-repudiere ajută entitățile implicate într-o tranzacție să rezolve posibilele dispute care pot apare cu privire la anumite evenimente sau acțiuni care s-au întâmplat (sau nu s-au întâmplat) în cadrul tranzacției.

Definiție

Un protocol de non-repudiere este un flux de tranzacții în care entitățile implicate schimbă dovezi electronice, capabile să ofere apoi servicii de non-repudiere.

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

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 și numai dacă poate genera o dovadă privind originea mesajului – destinată receptorului mesajului – și care prezentată unui judecător, acesta poate stabili fără dubiu că inițiatorul a trimis acel mesaj.

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 și numai dacă poate genera o dovadă privind originea mesajului – destinată receptorului mesajului – și care prezentată unui judecător, acesta poate stabili fără dubiu că inițiatorul a trimis acel mesaj.
- *Non-Repudierea Recepției*: un protocol de non-repudiere oferă non-repudierea recepției (*NRR*), dacă și numai dacă poate genera o dovadă privind primirea mesajului – destinată inițiatorului mesajului – și 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:

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

- *Alice* deține o dovadă de tip *NRR*, iar *Bob* deține o dovadă de tip *NRO*;

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

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

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

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

Protocoalele de non-repudiere se împart în:

1. protocoale cu corectitudine tare;

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

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

Protocoalele de non-repudiere se împart în:

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

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

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

Protocoalele de non-repudiere se împart în:

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

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

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

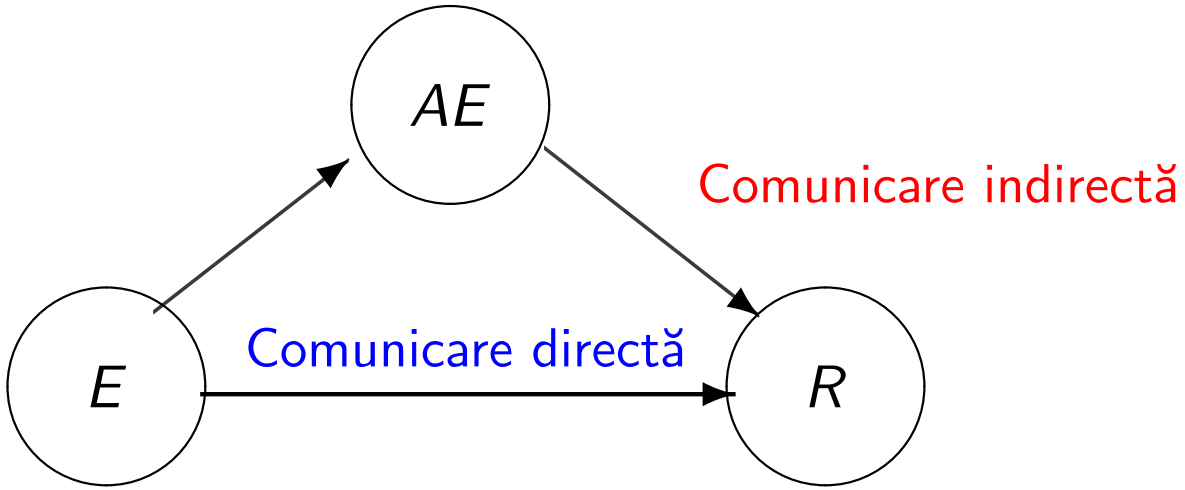
1. E trimite mesajul direct către R (comunicare directă);

Într-o tranzacție electronică, un 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).

Într-o tranzacție electronică, un 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* și *Receptorul* nu au încredere unul în celălalt, pentru a se putea contoriza corect acțiunile lor, sunt necesare următoarele servicii de non-repudiere:

În varianta de comunicare directă, dacă *Emitentul* și *Receptorul* nu au încredere unul în celălalt, pentru a se putea contoriza corect acțiunile lor, sunt necesare următoarele servicii de non-repudiere:

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

În varianta de comunicare directă, dacă *Emitentul* și *Receptorul* nu au încredere unul în celălalt, pentru a se putea contoriza corect acțiunile lor, sunt necesare următoarele servicii de non-repudiere:

- *Non-Repudierea Originii* (*NRO*): asigură protecția împotriva refuzului *Emitentului* de a recunoaște transmiterea mesajului. Dovada transmiterii mesajului va fi generată de *Emitent* sau un *TTP* și va fi păstrată de *Receptor*.
- *Non-repudierea Recepției* (*NRR*): asigură protecția împotriva refuzului *Receptorului* de a recunoaște primirea mesajului.

În varianta de comunicare directă, dacă *Emitentul* și *Receptorul* nu au încredere unul în celălalt, pentru a se putea contoriza corect acțiunile lor, sunt necesare următoarele servicii de non-repudiere:

- *Non-Repudierea Originii* (*NRO*): asigură protecția împotriva refuzului *Emitentului* de a recunoaște transmiterea mesajului. Dovada transmiterii mesajului va fi generată de *Emitent* sau un *TTP* și va fi păstrată de *Receptor*.
- *Non-repudierea Recepției* (*NRR*): asigură protecția împotriva refuzului *Receptorului* de a recunoaște primirea mesajului.Dovada primirii mesajului este generată de *Receptor* sau de un *TTP* și va fi păstrată de *Emitent*.

În varianta de comunicare intermediată de un *Agent de Expediere* (*AE*), pentru a se putea rezolva eventualele dispute între *Emitent* și *Agentul de Expediere*, respectiv între *Agentul de Expediere* și *Receptor*, sunt necesare serviciile:

În varianta de comunicare intermediată de un *Agent de Expediere* (*AE*), pentru a se putea rezolva eventualele dispute între *Emitent* și *Agentul de Expediere*, respectiv între *Agentul de Expediere* și *Receptor*, sunt necesare serviciile:

- *Non-repudierea depunerii* (*NRS*): asigură dovada că *Emitentul* a depus mesajul pentru expediere. Dovada depunerii mesajului este generată de *Agentul de Expediere* și 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* și *Agentul de Expediere*, respectiv între *Agentul de Expediere* și *Receptor*, sunt necesare serviciile:

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

Cerințe:

Protocoalele unui serviciu de non-repudiere trebuie să îndeplinească:

- *Corectitudine*: Nici una din cele două entități implicate nu trebuie să poată deține vreun avantaj față de cealaltă privind obținerea dovezilor de non-repudiere.

Cerințe:

Protocoalele unui serviciu de non-repudiere trebuie să îndeplinească:

- *Corectitudine*: Nici una din cele două entități implicate nu trebuie să poată deține vreun avantaj față de cealaltă privind obținerea dovezilor de non-repudiere.
  Repudierea poate fi prevenită numai dacă fiecare entitate obține în egală măsură informația de care are nevoie: *Bob* trebuie să obțină mesajul util și dovada de non-repudiere a originii acestui mesaj, iar *Alice* trebuie să obțină dovada de non-repudiere a recepției mesajului transmis.

Cerințe:

Protocoalele unui serviciu de non-repudiere trebuie să îndeplinească:

- *Corectitudine*: Nici una din cele două entități implicate nu trebuie să poată deține vreun avantaj față de cealaltă privind obținerea dovezilor de non-repudiere.
  Repudierea poate fi prevenită numai dacă fiecare entitate obține în egală măsură informația de care are nevoie: *Bob* trebuie să obțină mesajul util și dovada de non-repudiere a originii acestui mesaj, iar *Alice* trebuie să obțină dovada de non-repudiere a recepției mesajului transmis.

  În caz contrar, cei doi utilizatori nu trebuie să aibă acces la nici una din aceste informații.

Cerințe:

- *Eficiență*: Non-repudierea se obține în general prin folosirea unor servicii de tip *TTP*.
  Gradul de implicare al acestora este esențial în determinarea eficienței unui protocol de non-repudiere.

Cerințe:

- *Eficiență*: Non-repudierea se obține în general prin folosirea unor servicii de tip *TTP*.
  Gradul de implicare al acestora este esențial în determinarea eficienței unui protocol de non-repudiere.
- *Oportunitate*: Din diverse motive, o tranzacție din protocolul de non-repudiere poate fi întârziată sau chiar stopată intenționat de una dintre părțile implicate.
  Acest lucru nu trebuie să dezavantajeze cealaltă parte.

Cerințe:

- *Politică*: Toate regulile și toți parametrii necesari în cadrul serviciului de non-repudiere trebuie definite corect și complet.

Cerințe:

- *Politică*: Toate regulile și toți parametrii necesari în cadrul serviciului de non-repudiere trebuie definite corect și complet.
- *Transparența TTP-ului*: În anumite situații este de dorit ca implicarea *TTP*-ului în cadrul protocolului (sau a rezolvării disputelor) să fie invizibilă.
  Astfel, dovezile obținute prin implicarea *TTP*-ului vor fi similare celor obținute fără ajutorul acestuia.

Cerințe:

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

Rolul *TTP*-urilor în protocoalele de non-repudiere

Din punct de vedere al implicării *TTP*-ului în cadrul derulării unui protocol de non-repudiere, pot fi identificate trei situații:

Rolul *TTP*-urilor în protocoalele de non-repudiere

Din punct de vedere al implicării *TTP*-ului în cadrul derulării unui protocol de non-repudiere, pot fi identificate trei situații:

1. *Protocoale de non-repudiere bazate pe TTP-uri in-line*. Acționează ca intermediari în toate tranzacțiile efectuate între entități. Toate mesajele schimbate în cadrul protocolului trec pe la *TTP*.

Rolul *TTP*-urilor în protocoalele de non-repudiere

Din punct de vedere al implicării *TTP*-ului în cadrul derulării unui protocol de non-repudiere, pot fi identificate trei situații:

1. *Protocoale de non-repudiere bazate pe TTP-uri in-line*. Acționează ca intermediari în toate tranzacțiile efectuate între entități. 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 și verificarea dovezilor de non-repudiere.

Rolul *TTP*-urilor în protocoalele de non-repudiere

Din punct de vedere al implicării *TTP*-ului în cadrul derulării unui protocol de non-repudiere, pot fi identificate trei situații:

1. *Protocoale de non-repudiere bazate pe TTP-uri in-line*. Acționează ca intermediari în toate tranzacțiile efectuate între entități. 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 și verificarea dovezilor de non-repudiere.
3. *Protocoale de non-repudiere bazate pe TTP-uri off-line*. Terții de încredere nu participă activ în cadrul serviciului de non-repudiere (vor fi invocați doar în anumite situații de excepție).

Un *TTP* implicat în gestionarea dovezilor de non-repudiere se poate afla în una din situațiile:

Un *TTP* implicat în gestionarea dovezilor de non-repudiere se poate afla în una din situațiile:

- *Ca Autoritate de Certificare*.

Un *TTP* implicat în gestionarea dovezilor de non-repudiere se poate afla în una din situațiile:

- *Ca Autoritate de Certificare*.
  Un *CA* generează certificate digitale pentru cheile utilizatorilor, autentificându-le pentru a fi folosite în protocoalele de non-repudiere.
  Furnizează de asemenea listele de certificate revocate.

Un *TTP* implicat în gestionarea dovezilor de non-repudiere se poate afla în una din situațiile:

- *Ca Autoritate de Certificare*.
  Un *CA* generează certificate digitale pentru cheile utilizatorilor, autentificându-le pentru a fi folosite în protocoalele de non-repudiere.
  Furnizează de asemenea listele de certificate revocate.
  Autoritățile de Certificare sunt folosite întotdeauna în situațiile când pentru generarea dovezilor de non-repudiere sunt utilizate semnăturile digitale.

Un *TTP* implicat în gestionarea dovezilor de non-repudiere se poate afla în una din situațiile:

- *Ca Autoritate de Certificare*.
  Un *CA* generează certificate digitale pentru cheile utilizatorilor, autentificându-le pentru a fi folosite în protocoalele de non-repudiere.
  Furnizează de asemenea listele de certificate revocate.
  Autoritățile de Certificare sunt folosite întotdeauna în situațiile când pentru generarea dovezilor de non-repudiere sunt utilizate semnăturile digitale.

  În general, *TTP*-urile cu rol de *CA* sunt utilizate off-line. Pot acționa on-line, dacă semnăturile electronice folosesc formate avansate de semnătură electronică și conțin informații 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.

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 poate fi activat pentru a genera dovezile de non-repudiere în numele participanților.

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 poate fi activat pentru a genera dovezile de non-repudiere în numele participanților.

Dacă dovezile de non-repudiere se obțin pe bază de semnături digitale, notarul ar trebui să furnizeze mărci temporale privind momentul generării dovezilor.

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 poate fi activat pentru a genera dovezile de non-repudiere în numele participanților.

Dacă dovezile de non-repudiere se obțin 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 Autoritate de Expediere

O astfel de Autoritate constituie un terț de încredere în ceea ce privește expedierea mesajelor.

Ca Autoritate de Expediere

O astfel de Autoritate constituie un terț de încredere în ceea ce privește expedierea mesajelor.

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

Ca Autoritate de Arbitrare

Un *TTP* cu acest rol nu va fi implicat în protocoale decât în situațiile în care apar dispute, principalul său scop fiind judecarea și rezolvarea acestora.

Ca Autoritate de Arbitrare

Un *TTP* cu acest rol nu va fi implicat în protocoale decât în situațiile în care apar dispute, principalul său scop fiind judecarea și rezolvarea acestora.

Judecarea disputelor presupune evaluarea dovezilor puse la dispoziție de participanți și luarea în considerație a unei politici de non-repudiere.

Protocoale de non-repudiere fără *TTP*-uri

Pot asigura doar probabilistic cerințele de non-repudiere.
Chiar dacă în implementare se aleg corect parametrii protocolului, gradul de risc este foarte mic iar probabilitatea de nesoluționare a disputelor este neglijabilă, totuși aceste protocoale sunt ineficiente, fiind greu de aplicat.

Protocoale de non-repudiere fără *TTP*-uri

Pot asigura doar probabilistic cerințele de non-repudiere.
Chiar dacă în implementare se aleg corect parametrii protocolului, gradul de risc este foarte mic iar probabilitatea de nesoluționare a disputelor este neglijabilă, totuși aceste protocoale sunt ineficiente, fiind greu de aplicat.

Exemplu

Primele protocoale fără TTP apar la jumătatea anilor 80, *dezvoltate inițial pentru schimbul de chei criptografice.*
Ideea de bază era ca fiecare entitate să transmită pe rând biți succesivi din informația secretă care trebuia furnizată celeilalte entități, până la epuizarea informației.

Protocoale de non-repudiere fără *TTP*-uri

Pot asigura doar probabilistic cerințele de non-repudiere.
Chiar dacă în implementare se aleg corect parametrii protocolului, gradul de risc este foarte mic iar probabilitatea de nesoluționare a disputelor este neglijabilă, totuși aceste protocoale sunt ineficiente, fiind greu de aplicat.

Exemplu

Primele protocoale fără TTP apar la jumătatea anilor 80, *dezvoltate inițial pentru schimbul de chei criptografice.*
Ideea de bază era ca fiecare entitate să transmită pe rând biți succesivi din informația secretă care trebuia furnizată celeilalte entități, până la epuizarea informației.
Dacă o entitate stopa procesul, cealaltă proceda similar.

Protocolul Markowitch - Roggeman

Scopul este de a obține dovezile de non-repudiere *NRO* și *NRR* fără a utiliza un terț de încredere.

Protocolul Markowitch - Roggeman

Scopul este de a obține dovezile de non-repudiere *NRO* și *NRR* fără a utiliza un terț de încredere.

1. În prima rundă, *Alice*:
   1. Criptează mesajul M folosind cheia simetrică k și obține $c = E_k(M)$;

Protocolul Markowitch - Roggeman

Scopul este de a obține dovezile de non-repudiere *NRO* și *NRR* fără a utiliza un terț de încredere.

1. În prima rundă, *Alice*:
   1. Criptează mesajul M folosind cheia simetrică k și obține $c = E_k(M)$;
   2. Generează dovada originii lui c:

$$EOO_c = Sig_{Alice}(Bob, l, c)$$

unde $l = Hash(M, k)$;

Protocolul Markowitch - Roggeman

Scopul este de a obține dovezile de non-repudiere *NRO* și *NRR* fără a utiliza un terț de încredere.

1. În prima rundă, *Alice*:
   1. Criptează mesajul M folosind cheia simetrică k și obține $c = E_k(M)$;
   2. Generează dovada originii lui c:

$$EOO_c = Sig_{Alice}(Bob, l, c)$$

      unde $l = Hash(M, k)$;
   3. Trimite lui *Bob* (EOO_C, l, c).

Protocolul Markowitch - Roggeman

Scopul este de a obține dovezile de non-repudiere *NRO* și *NRR* fără a utiliza un terț de încredere.

1. În prima rundă, *Alice*:
   1. Criptează mesajul M folosind cheia simetrică k și obține $c = E_k(M)$;
   2. Generează dovada originii lui c:

$$EOO_c = Sig_{Alice}(Bob, l, c)$$

      unde $l = Hash(M, k)$;
   3. Trimite lui *Bob* (EOO_C, l, c).
2. *Bob* răspunde cu dovada recepției mesajului criptat c:

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

3 În rundele $i = 2, \ldots, n-1$

1. *Alice* trimite lui *Bob* cuplul $(EOO_{r_{i-1}}, r_{i-1})$ unde r_{i-1} este o valoare aleatoare de rundă, iar

$$EOO_{r_{i-1}} = Sig_{Alice}(Bob, i, r_{i-1})$$

3 În rundele $i = 2, \ldots, n-1$

1. *Alice* trimite lui *Bob* cuplul $(EOO_{r_{i-1}}, r_{i-1})$ unde r_{i-1} este o valoare aleatoare de rundă, iar

$$EOO_{r_{i-1}} = Sig_{Alice}(Bob, i, r_{i-1})$$

2. *Bob* răspunde cu dovada recepției valorii r_{i-1}:

$$EOR_{r_{i-1}} = Sig_{Bob}(Alice, i, r_{i-1})$$

3 În rundele $i = 2, \ldots, n-1$

1. *Alice* trimite lui *Bob* cuplul $(EOO_{r_{i-1}}, r_{i-1})$ unde r_{i-1} este o valoare aleatoare de rundă, iar

$$EOO_{r_{i-1}} = Sig_{Alice}(Bob, i, r_{i-1})$$

2. *Bob* răspunde cu dovada recepției valorii r_{i-1}:

$$EOR_{r_{i-1}} = Sig_{Bob}(Alice, i, r_{i-1})$$

4 În ultima rundă, *Alice* trimite cheia simetrică k și dovada $EOO_{r_{n-1}}$ a originii acesteia:

$$EOO_{r_{n-1}} = Sig_{Alice}(Bob, n, k)$$

3 În rundele $i = 2, \ldots, n-1$

1. *Alice* trimite lui *Bob* cuplul $(EOO_{r_{i-1}}, r_{i-1})$ unde r_{i-1} este o valoare aleatoare de rundă, iar

$$EOO_{r_{i-1}} = Sig_{Alice}(Bob, i, r_{i-1})$$

2. *Bob* răspunde cu dovada recepției valorii r_{i-1}:

$$EOR_{r_{i-1}} = Sig_{Bob}(Alice, i, r_{i-1})$$

4 În ultima rundă, *Alice* trimite cheia simetrică k și dovada $EOO_{r_{n-1}}$ a originii acesteia:

$$EOO_{r_{n-1}} = Sig_{Alice}(Bob, n, k)$$

5 *Bob* răspunde cu dovada recepției cheii k:

$$EOR_{r_{n-1}} = Sig_{Bob}(Alice, n, k)$$

După epuizarea rundei n și primirea lui $EOR_{r_{n-1}}$, *Alice* anunță terminarea protocolului, dezvăluind practic numărul de runde ales și faptul că în această ultimă rundă a trimis cheia simetrică k prin care *Bob* poate obține mesajul $M = D_k(c)$.

După epuizarea rundei n și primirea lui $EOR_{r_{n-1}}$, *Alice* anunță terminarea protocolului, dezvăluind practic numărul de runde ales și faptul că în această ultimă rundă a trimis cheia simetrică k prin care *Bob* poate obține mesajul $M = D_k(c)$.

Dovada de non-repudiere a originii va fi:

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

După epuizarea rundei n și primirea lui $EOR_{r_{n-1}}$, *Alice* anunță terminarea protocolului, dezvăluind practic numărul de runde ales și faptul că în această ultimă rundă a trimis cheia simetrică k prin care *Bob* poate obține 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 recepției va fi

$$NRR = (EOR_c, EOR_{r_{n-1}})$$

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c: dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c: dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;
  - $EOO_{r_{n-1}}$: dovedește că *Alice* i-a trimis și cheia simetrică de decriptare.

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c: dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;
  - $EOO_{r_{n-1}}$: dovedește că *Alice* i-a trimis și cheia simetrică de decriptare.
- *Alice* poate dovedi non-repudierea recepției:

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c: dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;
  - $EOO_{r_{n-1}}$: dovedește că *Alice* i-a trimis și cheia simetrică de decriptare.
- *Alice* poate dovedi non-repudierea recepției:
  - EOR_c: dovedește că *Bob* i-a confirmat primirea mesajului criptat cu o cheie simetrică;

În cazul unei dispute:

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

Protocolul se bazează pe păstrarea secretului privind numărul de runde și cheia secretă k, alese de *Alice* până la epuizarea întregului set de runde.

Protocolul se bazează pe păstrarea secretului privind numărul de runde și cheia secretă k, alese de *Alice* până la epuizarea întregului set de runde.

Bob, neștiind dinainte numărul de runde fixat de *Alice*, în ultima rundă nu va ști că a primit cheia simetrică decât după ce va fi anunțat de *Alice*, furnizând la rândul lui informația necesară de non-repudiere.

Securitate

La fiecare din ultimele $n-1$ runde, înainte de a trimite dovada non-repudierii recepției corespunzătoare, *Bob* ar putea încerca decriptarea mesajului primit în prima rundă; dacă reușește, poate alege să nu mai trimită această dovadă, ceea ce l-ar pune în avantaj față de *Alice*.

Securitate

La fiecare din ultimele $n-1$ runde, înainte de a trimite dovada non-repudierii recepției corespunzătoare, *Bob* ar putea încerca decriptarea mesajului primit în prima rundă; dacă reușește, poate alege să nu mai trimită această dovadă, ceea ce l-ar pune în avantaj față de *Alice*.
Această problemă se poate rezolva printr-o implementare care să contorizeze timpii de răspuns ai lui *Bob*.
Astfel, dacă apare o întârziere mai mare în cadrul unui răspuns de la *Bob*, protocolul eșuează.

Securitate

La fiecare din ultimele $n-1$ runde, înainte de a trimite dovada non-repudierii recepției corespunzătoare, *Bob* ar putea încerca decriptarea mesajului primit în prima rundă; dacă reușește, poate alege să nu mai trimită această dovadă, ceea ce l-ar pune în avantaj față de *Alice*.
Această problemă se poate rezolva printr-o implementare care să contorizeze timpii de răspuns ai lui *Bob*.
Astfel, dacă apare o întârziere mai mare în cadrul unui răspuns de la *Bob*, protocolul eșuează.
Pentru asta, trebuie ales un algoritm de criptare în care operația de decriptare să dureze un timp suficient ca să poată fi detectat de *Alice*.

Securitate

La fiecare din ultimele $n-1$ runde, înainte de a trimite dovada non-repudierii recepției corespunzătoare, *Bob* ar putea încerca decriptarea mesajului primit în prima rundă; dacă reușește, poate alege să nu mai trimită această dovadă, ceea ce l-ar pune în avantaj față de *Alice*.
Această problemă se poate rezolva printr-o implementare care să contorizeze timpii de răspuns ai lui *Bob*.
Astfel, dacă apare o întârziere mai mare în cadrul unui răspuns de la *Bob*, protocolul eșuează.
Pentru asta, trebuie ales un algoritm de criptare în care operația de decriptare să dureze un timp suficient ca să poată fi detectat de *Alice*.
Soluția este ineficientă practic.

- Pentru ca protocolul să fie funcțional, valorile aleatoare $r_1, \ldots, r_{n-1}$ trebuie să fie independente, egal distribuite, de lungimi aproximativ echivalente cu lungimea lui k.

- Pentru ca protocolul să fie funcțional, 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 și/sau un mod de utilizare al acestui algoritm care să nu permită decriptarea doar a unei părți a mesajului.

- Pentru ca protocolul să fie funcțional, 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 și/sau un mod de utilizare al acestui algoritm care să nu permită decriptarea doar a unei părți a mesajului.
- Oricare dintre entități poate întrerupe protocolul în rundele intermediare; în această situație, nici *Alice* și nici *Bob* nu deține un avantaj, deoarece nu va avea dovada privind trimiterea/recepția cheii simetrice de decriptare.

Există însă o probabilitate ca *Bob* să intuiască corect numărul de ordine al rundei finale și să întrerupă protocolul înainte de epuizarea corectă a acestei runde finale, ceea ce îi conferă un avantaj.

Există însă o probabilitate ca *Bob* să intuiască corect numărul de ordine al rundei finale și să întrerupă protocolul înainte de epuizarea corectă a acestei runde finale, ceea ce îi conferă un avantaj.

În această situație *Bob* va avea dovada completă a non-repudierii originii în timp ce lui *Alice* îi lipsește dovada recepției de către *Bob* a cheii de decriptare, deci nu poate dovedi că *Bob* a avut acces la mesaj.

Protocoale bazate pe *TTP*-uri in-line

În 1996 Coffey si Saidha au propus un protocol de non-repudiere bazat pe un *TTP* in-line utilizat pe post de *Server de Non-Repudiere* (*NRS - Non-Repudiation Server*).

Acest terț colectează dovezile de non-repudiere și le transmite apoi entităților care își dispută tranzacția.

Protocoale bazate pe *TTP*-uri in-line

În 1996 Coffey si Saidha au propus un protocol de non-repudiere bazat pe un *TTP* in-line utilizat pe post de *Server de Non-Repudiere* (*NRS - Non-Repudiation Server*).

Acest terț colectează dovezile de non-repudiere și le transmite apoi entităților care își dispută tranzacția.

Practic terțul de încredere funcționează ca un *Agent de Expediere* (*AE*) pentru mesajele schimbate între entitățile participante.

Protocoale bazate pe *TTP*-uri in-line

În 1996 Coffey si Saidha au propus un protocol de non-repudiere bazat pe un *TTP* in-line utilizat pe post de *Server de Non-Repudiere* (*NRS - Non-Repudiation Server*).

Acest terț colectează dovezile de non-repudiere și le transmite apoi entităților care își dispută tranzacția.

Practic terțul de încredere funcționează ca un *Agent de Expediere* (*AE*) pentru mesajele schimbate între entitățile participante.

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

Serverul de Non-Repudiere trebuie să aibă următoarele caracteristici:

Serverul de Non-Repudiere trebuie să aibă următoarele caracteristici:

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

Serverul de Non-Repudiere trebuie să aibă următoarele caracteristici:

- este independent de cele două entități, fiind însă de încredere pentru acestea;
- nu distribuie nici o informație legată de dovezi către vreo entitate participantă până când nu deține și dovada corespunzătoare pentru a putea fi distribuită către cealaltă entitate;

Serverul de Non-Repudiere trebuie să aibă următoarele caracteristici:

- este independent de cele două entități, fiind însă de încredere pentru acestea;
- nu distribuie nici o informație legată de dovezi către vreo entitate participantă până când nu deține și dovada corespunzătoare pentru a putea fi distribuită către cealaltă entitate;
- primește dovada de non-repudiere a originii direct de la *Alice* și cooperează cu *Bob* pentru a genera dovada de non-repudiere a recepției;

Serverul de Non-Repudiere trebuie să aibă următoarele caracteristici:

- este independent de cele două entități, fiind însă de încredere pentru acestea;
- nu distribuie nici o informație legată de dovezi către vreo entitate participantă până când nu deține și dovada corespunzătoare pentru a putea fi distribuită către cealaltă entitate;
- primește dovada de non-repudiere a originii direct de la *Alice* și cooperează cu *Bob* pentru a genera dovada de non-repudiere a recepției;
- odată ce deține toată informația necesară de non-repudiere pentru ambele entități, nu se va opune procesului de distribuție a acestei informații către cele două entități.

Protocolul Coffrey - Saidha

Se desfășoară în două faze:

Protocolul Coffrey - Saidha

Se desfășoară în două faze:

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

Protocolul Coffrey - Saidha

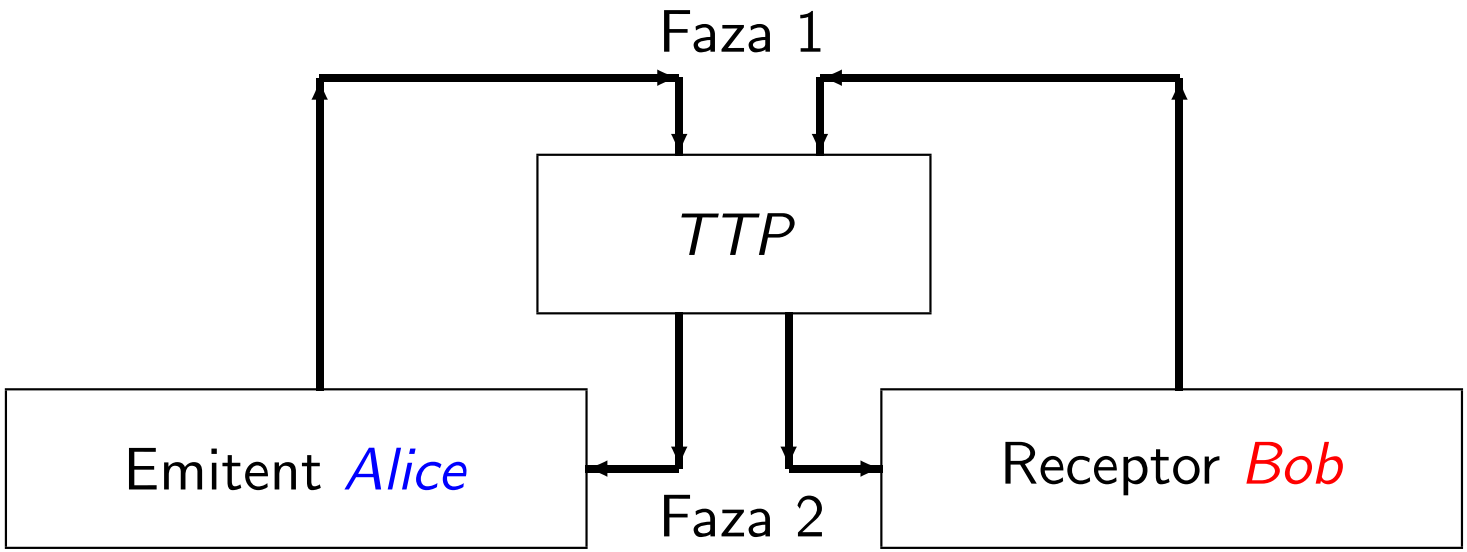
Se desfășoară în două faze:

1. (Faza 1) Generarea dovezilor de non-repudiabilitate; are loc la *TTP*.
2. (Faza 2) Distribuirea dovezilor de non-repudiabilitate.

Protocolul Coffrey - Saidha

Se desfășoară în două faze:

1. (Faza 1) Generarea dovezilor de non-repudiabilitate; are loc la *TTP*.
2. (Faza 2) Distribuirea dovezilor de non-repudiabilitate.



Toată comunicarea dintre *Alice* și *Bob* are avea loc prin intermediul Serverului de Non-Repudiere, iar mesajele schimbate sunt semnate digital și însoțite de o informație privind momentul semnării.

Toată comunicarea dintre *Alice* și *Bob* are avea loc prin intermediul Serverului de Non-Repudiere, iar mesajele schimbate sunt semnate digital și însoțite de o informație privind momentul semnării.

De aceea – pe lângă Serverul de Non-Repudiere – protocolul necesită și prezența unei *Autorități de Marcare Temporală* care să marcheze momentul semnării și să poată astfel demonstra că semnarea mesajelor s-a făcut în perioada de valabilitate a certificatelor digitale corespunzătoare cheilor de semnătură.

Descrierea formală

1. $Alice \longrightarrow TSA : \quad PK_{TSA}(NROp)$
2. $TSA \longrightarrow Alice : \quad PK_{Alice}(NRO)$

Alice – ca entitate inițiatoare – generează prin intermediul terțului de încredere dat de Autoritatea de Marcă Temporală (*TSA* – *Timestamp Authority*) dovada de non-repudiere a originii (*NRO*).
Alice trimite către *TSA* dovada parțială a originii:

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

și primește de la *TSA* marca temporală peste această semnătură:

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

unde *TSA* și ts_1 reprezintă identificatorul *TSA*-ului și respectiv timpul aplicat.

3. $Alice \longrightarrow NRS : \ "NR - req"$

La pasul 3 *Alice* inițiază o sesiune de lucru cu Serverul de Non-Repudiere (*NRS*), transmițându-i o cerere de non-repudiere $NR - req$.

$$\begin{array}{lll} 4. & NRS \longrightarrow Alice: & PK_{Alice}(n_1) \\ 5. & Alice \longrightarrow NRS: & PK_{NRS}(Sig_{Alice}(n_1, NRO, NRRp)) \end{array}$$

Alice transmite către *NRS* dovada de non-repudiere a originii (*NRO*) obținută la primii doi pași, precum și o dovadă parțială de non-repudiere a recepției:

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

unde h este o funcție de dispersie criptografică.

$$\begin{array}{lll} 4. & NRS \longrightarrow Alice: & PK_{Alice}(n_1) \\ 5. & Alice \longrightarrow NRS: & PK_{NRS}(Sig_{Alice}(n_1, NRO, NRRp)) \end{array}$$

Alice transmite către *NRS* dovada de non-repudiere a originii (*NRO*) obținută la primii doi pași, precum și o dovadă parțială de non-repudiere a recepției:

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

unde h este o funcție de dispersie criptografică.

Pentru a evita atacuri de tip replay, se folosește o secvență de tip provocare - răspuns bazată pe un nonce n_1, generat de serverul *NRS* la pasul 4.

$$\begin{array}{lll} 4. & NRS \longrightarrow Alice: & PK_{Alice}(n_1) \\ 5. & Alice \longrightarrow NRS: & PK_{NRS}(Sig_{Alice}(n_1, NRO, NRRp)) \end{array}$$

Alice transmite către *NRS* dovada de non-repudiere a originii (*NRO*) obținută la primii doi pași, precum și o dovadă parțială de non-repudiere a recepției:

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

unde h este o funcție de dispersie criptografică.

Pentru a evita atacuri de tip replay, se folosește o secvență 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*.

$$6.\quad NRS \longrightarrow Bob: \quad PK_{Bob}(Sig_{NRS}(n_2, NRRp))$$

Serverul *NRS* inițiază generarea dovezii de non-repudiere a recepției transmițându-i lui *Bob* un nonce n_2 (necesar mai târziu) și dovada parțială de non-repudiere *NRRp* primită de la *Alice*.

7.	$Bob \longrightarrow TSA:$	$PK_{TSA}(NRRps)$
8.	$TSA \longrightarrow Bob:$	$PK_{Bob}(NRR)$

Bob generează – prin intermediul terțului de încredere dat de *TSA* – dovada de non-repudiere a recepției (*NRR*).

Bob trimite spre *TSA* o dovadă parțială de non-repudiere a recepției, semnată de el:

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

și primește de la *TSA* marca temporală peste această semnătură:

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

9. $Bob \longrightarrow NRS : \quad PK_{NRS}(Sig_{Bob}(n_2, NRR))$

Bob trimite către Serverul de Non-Repudiere (*NRS*), dovada de non-repudiere a recepției *NRR*, însoțită de nonce-ul n_2 (pentru a evita atacurile de tip replay).

În acest moment, Faza 1 s-a încheiat și *NRS* – având ambele dovezi de non-repudiere – este în măsură să treacă la Faza 2 a protocolului: transmiterea acestor dovezi către *Alice* și *Bob*.

În acest moment, Faza 1 s-a încheiat și *NRS* – având ambele dovezi de non-repudiere – este în măsură să treacă la Faza 2 a protocolului: transmiterea acestor dovezi către *Alice* și *Bob*.

10.	$NRS \longrightarrow Bob:$	$PK_{Bob}(NRO)$
11.	$NRS \longrightarrow Alice:$	$PK_{Alice}(NRR)$

Astfel *Bob* va primi dovada de non-repudiere a originii mesajului (NRO) și – împreună cu acesta – și mesajul propriu zis M.

În acest moment, Faza 1 s-a încheiat și *NRS* – având ambele dovezi de non-repudiere – este în măsură să treacă la Faza 2 a protocolului: transmiterea acestor dovezi către *Alice* și *Bob*.

10.	$NRS \longrightarrow Bob:$	$PK_{Bob}(NRO)$
11.	$NRS \longrightarrow Alice:$	$PK_{Alice}(NRR)$

Astfel *Bob* va primi dovada de non-repudiere a originii mesajului (*NRO*) și – împreună cu acesta – și mesajul propriu zis *M*. Similar, *Alice* primește dovada de non-repudiere a recepției mesajului (*NRR*).

Securitate

- Dovezile de non-repudiere a originii și respectiv a recepției nu sunt trimise celor două entități până ce acestea nu depun aceste dovezi semnate la Serverul de Non-Repudiere.

Securitate

- Dovezile de non-repudiere a originii și respectiv a recepției nu sunt trimise celor două entități până ce acestea nu depun aceste dovezi semnate la Serverul de Non-Repudiere.
- Dovezile de non-repudiere deținute de *Alice* și *Bob* nu conțin semnătura *NRS*-ului.

Securitate

- Dovezile de non-repudiere a originii și respectiv a recepției nu sunt trimise celor două entități până ce acestea nu depun aceste dovezi semnate la Serverul de Non-Repudiere.
- Dovezile de non-repudiere deținute de *Alice* și *Bob* nu conțin semnătura *NRS*-ului.
  Acestea sunt:
  - $NRO = Sig_{TSA}(Sig_{Alice}(Alice, Bob, M), TSA, ts_1)$;
  - $NRR = Sig_{TSA}(Sig_{Bob}(Bob, Alice, h(NRO)), TSA, ts_2)$.

Securitate

- Dovezile de non-repudiere a originii și respectiv a recepției nu sunt trimise celor două entități până ce acestea nu depun aceste dovezi semnate la Serverul de Non-Repudiere.
- Dovezile de non-repudiere deținute de *Alice* și *Bob* nu conțin semnătura *NRS*-ului.
  Acestea sunt:
  - $NRO = Sig_{TSA}(Sig_{Alice}(Alice, Bob, M), TSA, ts_1)$;
  - $NRR = Sig_{TSA}(Sig_{Bob}(Bob, Alice, h(NRO)), TSA, ts_2)$.
- Cele două entități nu comunică niciodată direct; schimbul de mesaje (mesajul M și dovezile de non-repudiere) se face numai prin intermediul Serverului de Non-Repudiere.

- Mesajul *M* ajunge la *Bob* odată cu dovada de non-repudiere a originii *NRO* și nu înainte ca *NRS* să dețină dovada de non-repudiere a recepției *NRR*.

- Mesajul *M* ajunge la *Bob* odată cu dovada de non-repudiere a originii *NRO* și nu înainte ca *NRS* să dețină dovada de non-repudiere a recepției *NRR*.
- În cazul soluționării unei dispute în care *Alice* susține că *Bob* a primit mesajul *M*, arbitrul cere *NRS*-ului dovada de non-repudiere a recepției (*NRR*) precum și dovada de non-repudiere a originii (*NRO*).

- Mesajul *M* ajunge la *Bob* odată cu dovada de non-repudiere a originii *NRO* și nu înainte ca *NRS* să dețină dovada de non-repudiere a recepției *NRR*.
- În cazul soluționării unei dispute în care *Alice* susține că *Bob* a primit mesajul *M*, arbitrul cere *NRS*-ului dovada de non-repudiere a recepției (*NRR*) precum și dovada de non-repudiere a originii (*NRO*).
  Dacă aceste dovezi nu pot fi oferite, afirmația lui *Alice* este respinsă.

- Mesajul *M* ajunge la *Bob* odată cu dovada de non-repudiere a originii *NRO* și nu înainte ca *NRS* să dețină dovada de non-repudiere a recepției *NRR*.
- În cazul soluționării unei dispute în care *Alice* susține că *Bob* a primit mesajul *M*, arbitrul cere *NRS*-ului dovada de non-repudiere a recepției (*NRR*) precum și dovada de non-repudiere a originii (*NRO*).
  Dacă aceste dovezi nu pot fi oferite, afirmația lui *Alice* este respinsă.
  Altfel, arbitrul verifică semnătura lui *Alice* și marca temporală aplicată de *TSA* pe această semnătură.
  De asemenea verifică valoarea $h(NRO)$ din cadrul dovezii *NRR*, semnătura lui *Bob* și marca temporală aplicată peste aceasta.

- Mesajul *M* ajunge la *Bob* odată cu dovada de non-repudiere a originii *NRO* și nu înainte ca *NRS* să dețină dovada de non-repudiere a recepției *NRR*.
- În cazul soluționării unei dispute în care *Alice* susține că *Bob* a primit mesajul *M*, arbitrul cere *NRS*-ului dovada de non-repudiere a recepției (*NRR*) precum și dovada de non-repudiere a originii (*NRO*).
  Dacă aceste dovezi nu pot fi oferite, afirmația lui *Alice* este respinsă.
  Altfel, arbitrul verifică semnătura lui *Alice* și marca temporală aplicată de *TSA* pe această semnătură.
  De asemenea verifică valoarea $h(NRO)$ din cadrul dovezii *NRR*, semnătura lui *Bob* și marca temporală aplicată peste aceasta.
  În caz de succes, arbitrul aprobă afirmația lui *Alice*.

- În cazul soluționării unei dispute în care *Bob* susține că *Alice* a trimis mesajul M, arbitrul cere *NRS*-ului dovada *NRO*.

- În cazul soluționării unei dispute în care *Bob* susține că *Alice* a trimis mesajul M, arbitrul cere *NRS*-ului dovada *NRO*. Dacă aceasta nu poate fi oferită, afirmația lui *Bob* este respinsă.

- În cazul soluționării unei dispute în care *Bob* susține că *Alice* a trimis mesajul *M*, arbitrul cere *NRS*-ului dovada *NRO*.
  Dacă aceasta nu poate fi oferită, afirmația lui *Bob* este respinsă.
  Altfel, arbitrul verifică dacă mesajul din *NRO* satisface informația oferită de *Bob*.
  De asemenea, verifică semnătura lui *Alice* și marca temporală aplicată de *TSA* pe această semnătură.

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

- În cazul soluționării unei dispute în care *Bob* susține că *Alice* a trimis mesajul *M*, arbitrul cere *NRS*-ului dovada *NRO*.
  Dacă aceasta nu poate fi oferită, afirmația lui *Bob* este respinsă.
  Altfel, arbitrul verifică dacă mesajul din *NRO* satisface informația oferită de *Bob*.
  De asemenea, verifică semnătura lui *Alice* și marca temporală aplicată de *TSA* pe această semnătură.
  În caz de success, arbitrul aprobă afirmația lui *B*.
- *TSA*-ul participă activ la generarea dovezilor de non-repudiere; cei doi participanți precum și Serverul de Non-Repudiere, trebuie să aibă încredere în acesta și să verifice semnăturile digitale ale *TSA*-ului aplicate la marcarea temporală a mesajelor.

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

- *Alice* și *Bob* nu trebuie să aibă încredere unul în celălalt, dar trebuie să aibă încredere în cele două *TTP*-uri.
- Atacurile de tip replay în relația *NRS* - Entitate sunt evitate prin utilizarea unor secvențe de tip provocare -răspuns, bazate pe informații de tip nonce transmise de *NRS*.

Concluzii

Utilizarea de *TTP*-uri in-line (pentru asigurarea dovezilor necesare în serviciile de non-repudiere) poate fi o soluție viabilă.

Concluzii

Utilizarea de *TTP*-uri in-line (pentru asigurarea dovezilor necesare în serviciile de non-repudiere) poate fi o soluție viabilă.

Există câteva dezavantaje majore care descurajează punerea în practică a acestui tip de arhitectură:

Concluzii

Utilizarea de *TTP*-uri in-line (pentru asigurarea dovezilor necesare în serviciile de non-repudiere) poate fi o soluție viabilă.

Există 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 primește pentru a le retransmite.

Concluzii

Utilizarea de *TTP*-uri in-line (pentru asigurarea dovezilor necesare în serviciile de non-repudiere) poate fi o soluție viabilă.

Există 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 primește pentru a le retransmite.
2. La nivelul *TTP*-ului, "gâtuirea" tranzacțiilor este maximă.

Concluzii

Utilizarea de *TTP*-uri in-line (pentru asigurarea dovezilor necesare în serviciile de non-repudiere) poate fi o soluție viabilă.

Există 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 primește pentru a le retransmite.
2. La nivelul *TTP*-ului, "gâtuirea" tranzacțiilor este maximă.
3. Gestiunea centralizată de informații sensibile în cantități mari poate constitui o problemă, necesitând un nivel suplimentar de confidențialitate la nivelul terțului.

Protocoale bazate pe *TTP*-uri on-line

În cazul protocoalelor de non-repudiere bazate pe *TTP*-uri on-line, *TTP*-ul nu acționează ca Agent de Expediere (intermediar pentru fiecare tranzacție între entități).

Protocoale bazate pe *TTP*-uri on-line

În cazul protocoalelor de non-repudiere bazate pe *TTP*-uri on-line, *TTP*-ul nu acționează ca Agent de Expediere (intermediar pentru fiecare tranzacție între entități).
El intervine totuși în cadrul fiecărei sesiuni a protocolului.

Protocoale bazate pe *TTP*-uri on-line

În cazul protocoalelor de non-repudiere bazate pe *TTP*-uri on-line, *TTP*-ul nu acționează ca Agent de Expediere (intermediar pentru fiecare tranzacție între entități).
El intervine totuși în cadrul fiecărei sesiuni a protocolului.

Vom detalia protocolul propus de Zhou și Gollmann în 1996.
Aici *TTP*-ul funcționează ca un director de publicare read-only, de unde entitățile participante își obțin informații necesare pentru dovezile de non-repudiere.

Protocolul Zhou și Gollmann

1. *Alice* (ca entitate emitentă):

Protocolul Zhou și Gollmann

1. *Alice* (ca entitate emitentă):
   1. criptează mesajul M folosind cheia simetrică k : $c = E_k(M)$;

Protocolul Zhou și Gollmann

1. *Alice* (ca entitate emitentă):
   1. criptează mesajul M folosind cheia simetrică k : $c = E_k(M)$;
   2. generează dovada originii lui c:

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

Protocolul Zhou și Gollmann

1. *Alice* (ca entitate emitentă):
   1. criptează mesajul M folosind cheia simetrică k : $c = E_k(M)$;
   2. generează dovada originii lui c:

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

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

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

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

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

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

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

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

4. *TTP*:

1. trimite spre ambele entități confirmarea faptului că deține cheia k:

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

4. *TTP*:

1. trimite spre ambele entități confirmarea faptului că deține cheia k:
$$Con = Sig_{TTP}(Alice, Bob, l, t, k)$$
2. trimite lui *Bob* cheia simetrică k.

4. *TTP*:

1. trimite spre ambele entități confirmarea faptului că deține cheia k:
$$Con = Sig_{TTP}(Alice, Bob, l, t, k)$$
2. trimite lui *Bob* cheia simetrică k.
3. *Bob* poate recompune mesajul $M = D_k(c)$.

4. *TTP*:

1. trimite spre ambele entități confirmarea faptului că deține cheia k:
$$Con = Sig_{TTP}(Alice, Bob, l, t, k)$$
2. trimite lui *Bob* cheia simetrică k.
3. *Bob* poate recompune mesajul $M = D_k(c)$.

Dovada de non-repudiere a originii va fi:

$$NRO = (EOO_c, Con)$$

4. *TTP*:

1. trimite spre ambele entități confirmarea faptului că deține cheia k:
$$Con = Sig_{TTP}(Alice, Bob, l, t, k)$$
2. trimite lui *Bob* cheia simetrică k.
3. *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 recepției va fi:

$$NRR = (EOR_c, Con)$$

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c – dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c – dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;
  - *Con* – dovedește că *Bob* a putut obține de la *TTP* cheia simetrică de decriptare, deci a putut obține și mesajul.

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c – dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;
  - *Con* – dovedește că *Bob* a putut obține de la *TTP* cheia simetrică de decriptare, deci a putut obține și mesajul.
- *Alice* poate dovedi non-repudierea recepției:
  - EOR_c – dovedește că *Bob* i-a confirmat primirea mesajului (criptat cu o cheie simetrică);

În cazul unei dispute:

- *Bob* poate dovedi non-repudierea originii:
  - EOO_c – dovedește că *Alice* i-a trimis mesajul criptat cu o cheie simetrică;
  - *Con* – dovedește că *Bob* a putut obține de la *TTP* cheia simetrică de decriptare, deci a putut obține și mesajul.
- *Alice* poate dovedi non-repudierea recepției:
  - EOR_c – dovedește că *Bob* i-a confirmat primirea mesajului (criptat cu o cheie simetrică);
  - *Con* – dovedește că *TTP*-ul a publicat cheia simetrică, deci *Bob* a avut acces la această cheie, pentru decriptarea și obținerea lui M.

În cazul unei dispute:

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

Securitate

- După epuizarea primului pas, dacă *Bob* nu răspunde cu dovada primirii mesajului criptat EOR_c sau comunicația se întrerupe, *Alice* va opri protocolul fără a transmite cheia simetrică de decriptare către *TTP*.
  Astfel *Bob* nu va avea acces la această cheie și – implicit – nici la mesaj.

Securitate

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

Securitate

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

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

- Dacă *Alice* nu trimite către *TTP* cheia simetrică k, protocolul se încheie și niciunul din parteneri nu este în avantaj.
  Bob nu are mesajul și nici dovada completă a originii lui (îi lipseste *Con*), iar lui *Alice* îi lipsește *Con*, deci nu are dovada că *TTP*-ul a publicat cheia simetrică de decriptare și nu poate dovedi non-repudierea recepției.
- După primirea cheii k și a lui *Sub*, *TTP*-ul va publica tuplul $(Alice, Bob, l, k, Con)$ într-o intrare specifică lui *Alice*, de unde *Bob* poate obține cheia de decriptare.

- Dacă *Alice* nu trimite către *TTP* cheia simetrică k, protocolul se încheie și niciunul din parteneri nu este în avantaj.
  Bob nu are mesajul și nici dovada completă a originii lui (îi lipseste *Con*), iar lui *Alice* îi lipsește *Con*, deci nu are dovada că *TTP*-ul a publicat cheia simetrică de decriptare și nu poate dovedi non-repudierea recepției.
- După primirea cheii k și a lui *Sub*, *TTP*-ul va publica tuplul $(Alice, Bob, l, k, Con)$ într-o intrare specifică lui *Alice*, de unde *Bob* poate obține cheia de decriptare.
- Protocolul este protejat la atacuri de tip Denial-of-Service.

- Dacă *Alice* nu trimite către *TTP* cheia simetrică k, protocolul se încheie și niciunul din parteneri nu este în avantaj.
  Bob nu are mesajul și nici dovada completă a originii lui (îi lipseste *Con*), iar lui *Alice* îi lipsește *Con*, deci nu are dovada că *TTP*-ul a publicat cheia simetrică de decriptare și nu poate dovedi non-repudierea recepției.
- După primirea cheii k și a lui *Sub*, *TTP*-ul va publica tuplul $(Alice, Bob, l, k, Con)$ într-o intrare specifică lui *Alice*, de unde *Bob* poate obține cheia de decriptare.
- Protocolul este protejat la atacuri de tip Denial-of-Service. Entitățile rău intenționate nu pot trimite chei false în numele lui *Alice*, deoarece acestea trebuie semnate de ea.

Concluzii

În obținerea de servicii de non-repudiere, *TTP*-urile on-line pot constitui o soluție mai bună decât cele in-line.

Concluzii

În obținerea de servicii de non-repudiere, *TTP*-urile on-line pot constitui o soluție mai bună decât cele in-line.
Motive:

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

Concluzii

În obținerea de servicii de non-repudiere, *TTP*-urile on-line pot constitui o soluție mai bună decât cele in-line.
Motive:

1. *TTP*-ul nu acționează ca Agent de Expediere ci ca un Agent de Certificare pentru cheile de decriptare.
2. Entitățile nu schimbă mesajele prin intermediul *TTP*-ului, însă acesta participă activ în cadrul fiecărei instanțe a protocolului.

Concluzii

În obținerea de servicii de non-repudiere, *TTP*-urile on-line pot constitui o soluție mai bună decât cele in-line.
Motive:

1. *TTP*-ul nu acționează ca Agent de Expediere ci ca un Agent de Certificare pentru cheile de decriptare.
2. Entitățile nu schimbă mesajele prin intermediul *TTP*-ului, însă acesta participă activ în cadrul fiecărei instanțe a protocolului.
3. În cadrul schimbului de mesajelor există si un parametru (opțional) de timp t.
   Pentru a evita o încărcare prea mare a *TTP*-ului, parametrul t – agreat de *Alice* și *Bob* încă din primele două mesaje – specifică o perioadă de timp cât *TTP*-ul va păstra disponibilă informația de cheie și dovada publicării ei.

Protocoale bazate pe *TTP*-uri off-line

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

Protocoale bazate pe *TTP*-uri off-line

Un *TTP* este off-line într-un protocol de non-repudiere dacă nu intervine decât în situații 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 și *protocoale optimiste*.

În general toate protocoalele de non-repudiere cu *TTP*-uri off-line conțin cel puțin două sub-protocoale diferite:

În general toate protocoalele de non-repudiere cu *TTP*-uri off-line conțin cel puțin două sub-protocoale diferite:

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

În general toate protocoalele de non-repudiere cu *TTP*-uri off-line conțin cel puțin două sub-protocoale diferite:

- Un protocol principal (main) utilizat în cazurile normale – în care entitățile se comportă corect una față de cealaltă.
  TTP-ul nu intervine în acest protocol.
- Un protocol de recuperare (recovery) pentru situațiile cu probleme.
  În acest caz este necesară intervenția *TTP*-ului pentru furnizarea dovezilor de non-repudiere.

În general toate protocoalele de non-repudiere cu *TTP*-uri off-line conțin cel puțin două sub-protocoale diferite:

- Un protocol principal (main) utilizat în cazurile normale – în care entitățile se comportă corect una față de cealaltă. *TTP*-ul nu intervine în acest protocol.
- Un protocol de recuperare (recovery) pentru situațiile cu probleme.
  În acest caz este necesară intervenția *TTP*-ului pentru furnizarea dovezilor de non-repudiere.

În plus, unele protocoale conțin și:

- un protocolul adițional de ieșire forțată (abort), care poate fi declanșat de una din entități în anumite situații; în urma acestuia se va apela ulterior protocolul de recuperare (recovery) pentru rezolvarea disputelor.

Protocolul de non-repudiere Kremer - Merkowitch

În cadrul protocolului sunt generate următoarele dovezi:

Protocolul de non-repudiere Kremer - Merkowitch

În cadrul protocolului sunt generate următoarele dovezi:

- dovada originii mesajului criptat:

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

Protocolul de non-repudiere Kremer - Merkowitch

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): $Sub = Sig_{Alice}(Bob, PK_{TTP}(k))$.

Protocolul de non-repudiere Kremer - Merkowitch

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 recepției mesajului criptat și a cheii de criptare a mesajului (criptată cu cheia publică a TTP-ului):

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

Continuare

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

Continuare

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

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

Continuare

- 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:

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

1. *Alice*:
   1. Generează o cheie de sesiune k și criptează mesajul M: $c = E_k(M)$.

1. *Alice*:
   1. Generează o cheie de sesiune k și criptează mesajul M: $c = E_k(M)$.
   2. Calculează $PK_{TTP}(k)$ folosind cheia publică a *TTP*-ului;

1. *Alice*:
   1. Generează o cheie de sesiune k și criptează mesajul M: $c = E_k(M)$.
   2. Calculează $PK_{TTP}(k)$ folosind cheia publică a *TTP*-ului;
   3. Construiește dovezile *EOO* (privind originea mesajului criptat) și *Sub* (privind expedierea cheii de sesiune criptată).
   4. Trimite lui *Bob* aceste informatii: $(c, PK_{TTP}(k), EOO, Sub)$.

1. *Alice*:
   1. Generează o cheie de sesiune k și criptează mesajul M: $c = E_k(M)$.
   2. Calculează $PK_{TTP}(k)$ folosind cheia publică a *TTP*-ului;
   3. Construiește dovezile *EOO* (privind originea mesajului criptat) și *Sub* (privind expedierea cheii de sesiune criptată).
   4. Trimite lui *Bob* aceste informatii: $(c, PK_{TTP}(k), EOO, Sub)$.
2. *Bob* trimite către *Alice* dovada *NRR* de non-repudiere a recepției informațiilor.

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

Dacă *Bob* nu primește informațiile din pasul 3 al protocolului, apelează la protocolul de recuperare (recovery) a mesajului; lucru necesar deoarece *Bob* nu are încă acces la M.

Dacă *Bob* nu primește informațiile din pasul 3 al protocolului, apelează la protocolul de recuperare (recovery) a mesajului; lucru necesar deoarece *Bob* nu are încă acces la M.

4 *Bob* va trimite spre *TTP* o cerere de recuperare Y:

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

Dacă *Bob* nu primește informațiile din pasul 3 al protocolului, apelează la protocolul de recuperare (recovery) a mesajului; lucru necesar deoarece *Bob* nu are încă acces la M.

4 *Bob* va trimite spre *TTP* o cerere de recuperare Y:

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

5 *TTP*-ul:

1. Decriptează cheia de sesiune k:

Dacă *Bob* nu primește informațiile din pasul 3 al protocolului, apelează la protocolul de recuperare (recovery) a mesajului; lucru necesar deoarece *Bob* nu are încă acces la M.

4 *Bob* va trimite spre *TTP* o cerere de recuperare Y:

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

5 *TTP*-ul:

1. Decriptează cheia de sesiune k:
2. Trimite lui *Alice*: cheia k și dovada *NRR* (semnată de *Bob*) care atestă faptul că *Bob* are mesajul criptat și poate primi acum și cheia de decriptare.

Dacă *Bob* nu primește informațiile din pasul 3 al protocolului, apelează la protocolul de recuperare (recovery) a mesajului; lucru necesar deoarece *Bob* nu are încă acces la M.

4 *Bob* va trimite spre *TTP* o cerere de recuperare Y:

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

5 *TTP*-ul:

1. Decriptează cheia de sesiune k:
2. Trimite lui *Alice*: cheia k și dovada *NRR* (semnată de *Bob*) care atestă faptul că *Bob* are mesajul criptat și poate primi acum și cheia de decriptare.
3. Trimite lui *Bob* cheia de sesiune k decriptată și confirmarea dovezii Con_k pentru această cheie.

Analiza protocolului

- La primul pas *Alice* trimite mesajul criptat și cheia simetrică k – criptată cu cheia publică a *TTP*-ului.
  Acest lucru va permite *TTP*-ului – în faza de recuperare – să extragă cheia k și să o furnizeze lui *Bob*.

Analiza protocolului

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

Analiza protocolului

- La primul pas *Alice* trimite mesajul criptat și cheia simetrică k – criptată cu cheia publică a *TTP*-ului.
  Acest lucru va permite *TTP*-ului – în faza de recuperare – să extragă cheia k și să o furnizeze lui *Bob*.
  După acest mesaj, nici *Alice* și nici *Bob* nu posedă dovezi complete de non-repudiere.
- La terminarea pasului 2, deși *Bob* nu poate accesa încă mesajul, trimite lui *Alice* dovada completă de non-repudiere a recepției.

Analiza protocolului

- La primul pas *Alice* trimite mesajul criptat și cheia simetrică k – criptată cu cheia publică a *TTP*-ului.
  Acest lucru va permite *TTP*-ului – în faza de recuperare – să extragă cheia k și să o furnizeze lui *Bob*.
  După acest mesaj, nici *Alice* și nici *Bob* nu posedă dovezi complete de non-repudiere.
- La terminarea pasului 2, deși *Bob* nu poate accesa încă mesajul, trimite lui *Alice* dovada completă de non-repudiere a recepției.
  Dacă *Bob* nu primește ulterior cheia de decriptare (adică pasul 3 nu mai are loc), el o poate obține de la *TTP* prin protocolul de recuperare.

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

- Dovada completă de non-repudiere a expedierii este $\{EOO, Sub, EOO_k\}$, obținută 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 și este $\{EOO, Sub, Con_k\}$.
  În acest caz, *Bob* primește și cheia simetrică de decriptare.

- Dovada completă de non-repudiere a expedierii este $\{EOO, Sub, EOO_k\}$, obținută 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 și este $\{EOO, Sub, Con_k\}$.
  În acest caz, *Bob* primește și cheia simetrică de decriptare.
- *Bob* poate încerca să trișeze și să lanseze protocolul de recuperare mai devreme.

- Dovada completă de non-repudiere a expedierii este $\{EOO, Sub, EOO_k\}$, obținută 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 și este $\{EOO, Sub, Con_k\}$.
  În acest caz, *Bob* primește și cheia simetrică de decriptare.
- *Bob* poate încerca să trișeze și să lanseze protocolul de recuperare mai devreme.
  Imediat după pasul 1 el poate întrerupe protocolul principal fără să mai trimită lui *Alice* dovada *NRR*.

- Dovada completă de non-repudiere a expedierii este $\{EOO, Sub, EOO_k\}$, obținută 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 și este $\{EOO, Sub, Con_k\}$.
  În acest caz, *Bob* primește și cheia simetrică de decriptare.
- *Bob* poate încerca să trișeze și 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 entități 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).

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 și dovada expedierii acestei chei, *Alice* a primit dovada corectă de non-repudiere a recepției.

Securitate

Niciuna din entități nu este avantajată în ceea ce priveste expedierea și recepția informației utile și a dovezilor corespunzătoare necesare.

Securitate

Niciuna din entități nu este avantajată în ceea ce priveste expedierea și recepția informației utile și a dovezilor corespunzătoare necesare.

Există însă un dezavantaj pentru *Alice*; dacă *Bob* refuză să completeze pasul 2, *Alice* nu va primi dovada recepției.

Securitate

Niciuna din entități nu este avantajată în ceea ce priveste expedierea și recepția informației utile și a dovezilor corespunzătoare necesare.

Există însă un dezavantaj pentru *Alice*; dacă *Bob* refuză să completeze pasul 2, *Alice* nu va primi dovada recepției.
În această situație:

- *Bob* poate lansa imediat protocolul de recuperare.

Securitate

Niciuna din entități nu este avantajată în ceea ce priveste expedierea și recepția informației utile și a dovezilor corespunzătoare necesare.

Există însă un dezavantaj pentru *Alice*; dacă *Bob* refuză să completeze pasul 2, *Alice* nu va primi dovada recepției.
În această situație:

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

- *Bob* poate aștepta o perioadă de timp (mai mică sau mai mare) și abia după aceea va lansa protocolul de recuperare.

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

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

Acest lucru îl plasează pe *Bob* pe o poziție avantajată față de *Alice*, deoarece *Bob* nu poate fi niciodată în situația de a depinde de *Alice*.

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

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

Concluzii

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

Concluzii

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

- Implicarea *TTP*-ului poate fi doar ocazională.
  În majoritatea timpului, tranzacțiile se desfășoară doar între entitățile emițător și receptor.

Concluzii

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

- Implicarea *TTP*-ului poate fi doar ocazională.
  În majoritatea timpului, tranzacțiile se desfășoară doar între entitățile emițător și receptor.
- *TTP*-ul nu trebuie să stocheze informațiile primite de la entități decât în anumite situații ceea ce diminuează necesarul de resurse.

Concluzii

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

- Implicarea *TTP*-ului poate fi doar ocazională.
  În majoritatea timpului, tranzacțiile se desfășoară doar între entitățile emițător și receptor.
- *TTP*-ul nu trebuie să stocheze informațiile primite de la entități decât în anumite situații ceea ce diminuează necesarul de resurse.
- Caracterul off-line al protocoalelor elimină dezavantajele cu care se confruntă serviciile on-line (redundanța serviciului și a conectivității, amenințări cu diverse atacuri cum ar fi cele de tip Denial-of-Service, intruziune etc.).

The END