Signatura electrònica de contractes

Josep Lluís Ferrer Gomila
Llorenç Huguet Rotger
M. Magdalena Payeras Capellà

PID_00199771

UOC
Universitat Oberta de Catalunya

Índex

Introducció

El cicle de vida d'un procés de contractació comprèn diferents fases, entre les quals podem trobar: negociació, signatura del contracte (perfeccionament) i execució del contracte. La primera i última no són objecte d'aquesta secció, i en general no requereixen elevades mesures de seguretat per a portar-les a terme de manera electrònica. En aquest mòdul ens centrem en la fase de signatura del contracte. S'entén que un contracte s'ha perfeccionat (terme jurídic que significa que es considera signat el contracte) quan les parts implicades s'han compromès a executar-lo.

Des de la perspectiva tècnica, la signatura electrònica de contractes forma part d'un conjunt de problemes que de manera genèrica rep el nom d'*intercanvi equitatiu de valors*. La idea substancial és que dues o més parts tenen un objecte (un cada una) que volen intercanviar pel de les altres parts intervinents en la transacció, però volen tenir la garantia que si ells proporcionen el seu objecte, rebran l'objecte o objectes que esperen a canvi. Arribem així, de manera ràpida, a la propietat fonamental que demanarem a qualsevol protocol per a la signatura electrònica de contractes: l'equitat. És a dir, que o totes les parts implicades reben el que esperen al final de l'intercanvi, o cap no estarà en una posició avantatjosa al final d'aquest intercanvi.

La propietat fonamental dels protocols de signatura electrònica de contractes és l'equitat: que cap de les parts no es trobi en una situació de desavantatge després de finalitzar l'execució del protocol de signatura de contractes.

La solució més habitual en el món en paper és gairebé trivial, ja que moltes signatures de contractes es fan de manera presencial, i això garanteix que les parts obtindran el que estan esperant, sense assumir el risc de proporcionar la seva sense rebre la de la contrapart. Òbviament, en el món electrònic l'intercanvi cara a cara no és possible, i per tant haurem de proporcionar solucions alternatives però que permetin mantenir (o millorar) els nivells de seguretat del món en paper.

Les solucions per a la signatura electrònica de contractes se solen classificar en funció de si intervé o no una tercera part de confiança (TTP, de l'anglès *trusted third party*), i si hi intervé, com ho fa (en tots els contractes, si sorgeixen problemes, etc.). Per això, exposarem solucions amb TTP i solucions sense TTP.

1. Solucions per a la signatura electrònica de contractes amb TTP

La classificació feta en la introducció respecte de les TTP es pot refinar més, segons el grau d'intervenció de la TTP en l'execució dels protocols per a la signatura electrònica de contractes. Així parlem de protocols per a la signatura electrònica de contractes amb:

- TTP *in-line*, quan la TTP intervé en tots els passos de l'execució del protocol.
- TTP *on-line*, quan la TTP intervé en totes les execucions del protocol de signatura electrònica de contractes, però no en tots els passos.
- TTP *off-line*, quan la TTP només intervé en cas que sorgeixin conflictes o problemes entre les parts, és a dir, sota demanda d'una o més de les parts; per tant, en aquest tipus de protocols la TTP no intervé de manera genèrica, i també reben el nom de *protocols optimistes*.

Ara ens centrarem en la signatura electrònica de contractes entre dues parts, és a dir, dues parts contractants volen intercanviar les seves signatures sobre el text d'un contracte.

Abans de començar amb la descripció dels protocols, procedirem a descriure la notació per a designar els actors que usarem. Els actors que hi intervindran són, per una part, els signants del contracte, que denominarem A(lice) i B(ob), i de l'altra part tenim la tercera part de confiança, que denominarem T.

Solució amb TTP *in-line*

Començarem amb la que es pot considerar com la solució més senzilla possible, utilitzant una TTP *in-line*. A i B han d'enviar el contracte i la seva signatura sobre aquest a la TTP. Una vegada la TTP disposa de les dues còpies signades, procedeix a verificar que les signatures són correctes i, si és el cas, retransmet la còpia signada per A a B, i la de B a A:

A $\rightarrow$ T: M,$Sign_A(M)$
B $\rightarrow$ T: M,$Sign_B(M)$
T : verifica les signatures de A i B
T $\rightarrow$ A: M,$Sign_B(M)$
T $\rightarrow$ B: M,$Sign_A(M)$

Noteu que en aquest senzill protocol la TTP és la que proporciona les garanties que l'intercanvi sigui equitatiu. Fins que no disposa de les còpies de totes dues parts, no les retransmet. Per tant, es compleix la propietat d'equitat. No

obstant això, amb aquest senzill exemple detectem alguns inconvenients d'aquest tipus de solucions.

D'una banda, el fet que la TTP hagi d'intervenir en cada intercanvi pot representar un problema de coll d'ampolla pel que fa a les comunicacions amb aquesta, o en qualsevol cas, la seva intervenció permanent pot encarir el procés de contractació. Per això se solen preferir aquelles solucions que no requereixen una participació en totes les execucions de la TTP.

Un altre aspecte que s'ha de considerar és la confiança que cal dipositar en la TTP, com a mínim des de dos punts de vista. D'una banda, és fàcil observar que la TTP pot afavorir una de les parts, i pot proporcionar la còpia, per exemple, de B a A i no la de A a B. Això conduiria a una situació no equitativa per culpa de la TTP.

D'altra banda, tal com s'ha dissenyat el protocol, la TTP té accés a tota la informació relacionada amb el contingut del contracte; per tant, la privacitat i el secret de la transacció queda també en mans de la TTP, que la podrà difondre a tercers sense el coneixement de A i B.

Solució amb TTP *off-line* asíncrona

Per a resoldre alguns dels problemes indicats anteriorment, explicarem a continuació una solució optimista o amb TTP *off-line*. En aquesta solució apareixen tres subprotocols: un subprotocol d'intercanvi, un de cancel·lació i un de finalització. Començarem explicant el d'intercanvi. Els passos que cal seguir són els següents:

A → B: M,$Sign_A(M)$
B → A: $Sign_B(M)$
A → B: $Sign_A[Sign_B(M)]$

Un cop intercanviada la informació anterior, el contracte està signat. Per a A la prova del contracte és senzillament la signatura de B sobre el contracte, i per a B la signatura del contracte consisteix en la signatura de A i l'acusament de recepció (el tercer pas) que li ha enviat A. Noteu que si totes dues parts actuen correctament (i no hi ha problemes de comunicacions) la TTP no ha d'intervenir.

En una solució optimista o amb TTP *off-line* si les parts actuen correctament i no es produeixen problemes de comunicació s'aconsegueix la propietat d'equitat sense que hagi d'intervenir la TTP.

Però, què passa si alguna de les dues parts intenta fer trampes? Més concretament, què passa si A, un cop disposa de la signatura de B, no envia l'acusament de recepció del tercer pas? Si A actués d'aquesta manera, tindria còpia signa-

da del contracte de B, i B només disposaria de la prova que A volia signar el contracte, però no de la prova plena que el contracte està signat. Per tant, la situació no seria equitativa. Per això cal un subprotocol perquè, si es dóna aquesta situació, permeti a B restablir l'equitat de l'intercanvi. Es tracta del subprotocol de finalització:

B → T: M,$Sign_A(M)$,$Sign_B(M)$
T → B: $Sign_T[Sign_B(M)]$

En primer lloc, B ha de procedir a enviar a la TTP la informació que li va proporcionar A en el primer pas. Amb això (comprovant la signatura feta per A) la TTP pot verificar que A va proposar a B signar el contracte M. Si és així, i si la TTP no disposa de més informació, la TTP envia a B un acusament de recepció equivalent al que hauria d'haver enviat A (però ara signat per la TTP). D'aquesta manera hem restablert l'equitat per a B.

Però a continuació podem observar que es pot donar una situació no desitjable per a A. Amb el subprotocol anterior hem aconseguit que B, un cop ha rebut el primer pas de A, pugui finalitzar l'execució del protocol de signatura de contractes en qualsevol moment, fins i tot si no envia (sense ni tan sols fer l'intent) la informació del segon pas a A.

A priori això no representa un problema greu d'equitat per a A, perquè si B no contacta amb la TTP, cap dels dos no té res que comprometi l'altra part, i si B contacta amb la TTP, l'únic que cal fer és permetre que A també pugui contactar amb la TTP, la qual li hauria de proporcionar la signatura de B (que aquest havia remès a la TTP quan hi va contactar):

A → T: M,$Sign_A(M)$
T → A: $Sign_B(M)$

Però en el cas que B no hagi contactat amb la TTP en el moment en què ho faci A, què ha de fer la TTP? Una primera opció seria que no fes res, però d'aquesta manera A quedaria en una situació no desitjable: A sap que B pot contactar amb la TTP en qualsevol moment i obtenir l'acusament de recepció de la TTP. Per a solucionar aquesta situació conflictiva hi ha dues solucions.

En primer lloc, podem establir una data d'expiració després de la qual la TTP no ha de resoldre noves peticions, o bé es pot afegir una resposta nova de la TTP per a cancel·lar l'intercanvi:

A → T: M,$Sign_A(M)$
T → A: $Sign_T(cancel - signatura - M)$

És a dir, que si la TTP no ha rebut cap petició de B (per tant, la TTP pensa que de moment per a B tot és correcte), el que ha de fer és enviar un missatge de cancel·lació vinculat al contracte amb A. Ara hauríem de combinar els dos casos anteriors en un únic subprotocol per a A:

A → T: M,$Sign_A(M)$
Si B_ha_contactat_amb_T then T → A: $Sign_B(M)$
Else T → A: $Sign_T(cancel - signatura - M)$

D'aquesta manera preveiem les dues situacions possibles que es poden donar quan A contacta amb la TTP. Quedaria ara per completar el subprotocol de finalització que havíem explicat per a B. Ara cal preveure el cas en què A hagi contactat amb la TTP abans que B:

B → T: M,$Sign_A(M)$,$Sign_B(M)$
Si A_ha_contactat_amb_T then T → B: $Sign_T(cancel - signatura - M)$
Else T → B: $Sign_T[Sign_B(M)]$

Ara el protocol de manera global compleix la propietat fonamental d'equitat. Per comprovar-ho, analitzem les situacions possibles que es poden donar:

- A i B finalitzen l'execució del subprotocol d'intercanvi. En aquest cas queda clar que després d'executar els tres passos, totes dues parts acaben obtenint prova que el contracte està signat.

- A envia el primer missatge i no rep res de B (perquè B no ha enviat la resposta, o perquè s'ha perdut mentre estava en trànsit). A ha de contactar amb la TTP, i si B no havia contactat amb T (això significa que B no pot tenir l'acusament de recepció) rebrà un missatge de cancel·lació; per tant, cap de les dues parts no tindrà prova que el contracte està signat (perquè de fet no està signat). En el cas que B ja hagi contactat amb la TTP, aquesta hauria proporcionat un acusament de recepció a B, i ara ha de proporcionar la signatura de B a A: per tant, totes dues parts tindran prova que el contracte està signat.

- B envia el seu missatge i no rep l'acusament de recepció. De manera anàloga a l'anterior, es pot demostrar que o tots dos reben una cancel·lació, o tots dos disposaran d'una prova que el contracte està signat.

Ja hem demostrat que el protocol anterior és equitatiu, però ara explicarem una situació aparentment contradictòria que es pot donar. Suposem que s'executa el protocol d'intercanvi sense cap problema, però a més *a posteriori* A contacta amb la TTP per cancel·lar l'intercanvi. Com la TTP no disposa de més informació (B no hi ha contactat), ha de proporcionar un missatge de cancel·lació a A. D'aquesta manera A disposa d'una doble prova: que el contracte està signat (amb la prova aportada per B) i que el contracte està cancel·lat (amb la prova aportada per T).

Però en el cas que sorgeixin litigis entre A i B, per a un àrbitre extern quedarà clar que A és trampós si intenta al·legar que el contracte està cancel·lat. Si A intenta fer prevaler la prova de cancel·lació davant d'un àrbitre, B aportarà els dos missatges enviats per A (la signatura de A i l'acusament de recepció de A).

En concret, l'acusament de recepció de A demostra que aquest és trampós, ja que en cap cas A no hauria enviat aquest acusament de recepció sense haver disposat abans de la signatura de B. En conclusió, A va contactar amb la TTP després d'haver rebut la còpia signada de B, i va intentar fer trampes.

Un dels problemes que plantejava la solució amb TTP *in-line* era que calia dipositar molta confiança en la TTP. Ens hauríem de preguntar ara com és el nou protocol des del punt de vista de possibles aliances de la TTP amb les diferents parts. Tècnicament denominem *verificabilitat de la TTP* aquesta propietat.

Direm que un protocol de signatura electrònica de contractes amb TTP compleix la propietat que la TTP és verificable si les parts intervinents poden demostrar un mal comportament de la TTP si aquesta no segueix els passos previstos en el protocol.

Per a fer l'anàlisi del protocol proposat anteriorment observarem els dos possibles atacs que pot portar a terme la TTP. Una assumpció que farem és que la TTP sempre respon a les peticions de les parts (en cas contrari seria possible demostrar aquest mal comportament de la TTP amb un àrbitre extern).

El primer atac que es podria fer és una aliança de la TTP amb A. Així, l'objectiu seria proporcionar còpia del contracte a A, i deixar sense prova B. Però aquesta situació no és possible perquè, si B contacta amb la TTP, aquesta o bé li ha de proporcionar l'acusament de recepció de A, o l'acusament de recepció de T, o el missatge de cancel·lació. Si es donen les dues primeres situacions, l'atac ha fracassat. I si es dóna la tercera situació, B disposarà d'una prova que li permetrà demostrar *a posteriori* (si A intenta fer valer la còpia signada de B) que o A o T van fer trampes.

El segon atac que es podria produir és l'aliança de T amb B. En aquest cas, l'objectiu és proporcionar còpia del contracte a B, i intentar deixar A sense proves. Per a això, la TTP proporciona l'acusament de recepció de A i el de cancel·lació a B, i un missatge de cancel·lació a A. Aquesta situació no pot ser detectada *a priori* per A. En aquest moment, B es troba en una situació de privilegi perquè si *a posteriori* li interessa afirmar que el contracte està signat, aportarà l'acusament de recepció de A, i en cas contrari només aportarà la cancel·lació de T. Al contrari, des del punt de vista de A, el contracte està cancel·lat. Com es pot donar aquesta situació? Perquè recordem que hi havia la possibilitat que després de finalitzar l'intercanvi A contacti amb la TTP per obtenir un missatge de cancel·lació i intentar fer trampes. Aquesta possible situació ara s'ha tornat en contra seva.

També hem dit en el protocol amb TTP *in-line* que les parts havien de dipositar molta confiança en la TTP perquè tenia accés al contingut del missatge. En la nova proposta hem millorat el problema una mica, però no totalment.

Si la TTP no ha d'intervenir, no tindria accés al missatge, però si la TTP ha d'intervenir, observem que les parts envien el text del missatge, M, a la TTP.

Un protocol per a la signatura electrònica de contractes compleix la propietat de confidencialitat si només els signants del contracte tenen accés al contingut del missatge, i ni tan sols la TTP té accés a aquest contingut.

Per tant, el protocol plantejat és millorable des del punt de vista de la confidencialitat de la informació intercanviada, perquè no solament la TTP pot arribar a tenir accés a la informació del contracte, sinó també qualsevol espia que tingui accés al canal de comunicacions (el contracte M es transmet en clar entre A i B). Ara introduirem unes petites millores per aconseguir la propietat de confidencialitat. El primer que farem és xifrar la comunicació entre A i B (això es podria aconseguir de manera senzilla en la pràctica utilitzant protocols ja estandarditzats com SSL/TLS):

A → B: $E_k(M)$,$PU_B(k)$,$Sign_A(M)$
B → A: $Sign_B(M)$
A → B: $Sign_A[Sign_B(M)]$

El mecanisme utilitzat és senzill i consisteix a xifrar el contracte M amb un criptosistema simètric (per exemple, AES) amb una clau secreta generada per A. Però B necessitarà conèixer aquesta clau secreta, i per això xifrem la clau secreta, k, amb la clau pública de B d'un criptosistema de clau pública (per exemple, RSA). Quan B rep la informació del primer pas, la primera operació que ha de fer és un desxifratge amb la seva clau privada per a recuperar la clau k. Amb aquesta clau k pot fer el desxifratge del criptograma que conté el text del contracte. Finalment, amb el text del contracte i la signatura de A sobre aquest, B pot verificar si aquesta signatura és correcta.

Un cop resolta la confidencialitat davant tercers, quedaria resoldre la confidencialitat davant la TTP. En realitat, la TTP no necessita estrictament tenir accés al contingut del contracte, ja que la TTP només necessita verificar que les signatures de les parts són correctes. Per a això n'hi ha prou que proporcionem a la TTP el resum del contracte que s'ha utilitzat per a fer la signatura electrònica:

A → T: $H(M)$,$Sign_A(M)$
Si B_ha_contactat_amb_T then T → A: $Sign_B(M)$
Else T → A: $Sign_T(cancel - signatura - M)$

B → T: $H(M)$,$Sign_A(M)$,$Sign_B(M)$
Si A_ha_contactat_amb_T then T → B: $Sign_T(cancel - signatura - M)$
Else T → B: $Sign_T[Sign_B(M)]$

Per a fer la verificació de si les signatures són correctes, l'únic que ha de fer la TTP és aplicar les claus públiques de A i B (segons pertoqui) i a continuació comparar el valor obtingut amb el resum que li han transmès (si coincideixen, la signatura és correcta, i en cas contrari, la signatura no es correspon amb el resum rebut). Ara sí que podem afirmar que el protocol compleix la propietat de confidencialitat.

Una propietat interessant del protocol que acabem de presentar és que no s'han introduït restriccions temporals. Les parts poden contactar amb la TTP quan vulguin per finalitzar l'execució del protocol. Per això diem que el protocol és asíncron (no requereix la sincronització dels rellotges de cap de les parts que hi intervenen).

Un protocol de signatura electrònica de contractes compleix la propietat de temporalitat (*timeliness* en anglès) si les parts poden tenir la garantia que l'execució del protocol es pot finalitzar quan elles vulguin, sense assumir el risc de perdre l'equitat de l'intercanvi.

Solució amb TTP *off-line* síncrona

Hem vist que el fet que el protocol fos asíncron (les parts podien contactar amb la TTP en qualsevol moment) obligava a introduir un subprotocol de cancel·lació per a A. Ara veurem com podríem modificar la proposta si establim un temps límit, és a dir, que un cop transcorregut un cert temps ja només es puguin fer consultes a la TTP de quin és l'estat de la transacció, però no demanar-li que faci cap acció de modificar l'estat de l'intercanvi. Suposarem que el subprotocol d'intercanvi coincideix amb l'anterior:

A → B: M,$Sign_A(M)$
B → A: $Sign_B(M)$
A → B: $Sign_A[Sign_B(M)]$

Tal com succeïa abans, es pot donar la situació que B enviï el compromís, però no rebi l'acusament de recepció de A. Per això hem d'establir un subprotocol perquè B pugui restablir l'equitat de l'intercanvi. Però ara imposem una condició addicional a B, i és que ha de contactar amb la TTP abans d'un determinat valor temporal t. Si es troba en una situació no equitativa i no contacta amb la TTP abans de t, el problema ja no tindrà solució: quedarà en desavantatge davant A (però a causa de la seva inacció). Per tant, el subprotocol queda com segueix:

B → T: M,$Sign_A(M)$,$Sign_B(M)$
Si t-actual<t then T → B: $Sign_T[Sign_B(M)]$
Else T → B: $Sign_T[fora-de-termini]$

En aquest cas no té sentit que A contacti amb la TTP per cancel·lar l'intercanvi, ja que A sap que si no rep la signatura de B després de transcórrer el temps t (que ella ha triat) podrà consultar la TTP per saber l'estat de l'intercanvi. El subprotocol per a A seria:

A → T: M,$Sign_A(M)$
Si B_ha_contactat_amb_T then T → A: $Sign_B(M)$
Else T → A: $Sign_T(cancel - signatura - M)$

De fet, en cas que B no hagi contactat amb la TTP, no caldria que T enviés un missatge de cancel·lació a A, perquè després del temps t, B ja no podrà aconseguir l'acusament de recepció de A ni de T i, per tant, cap de les parts no tindrà res que comprometi l'altra.

La introducció d'una restricció temporal simplifica el protocol plantejat (i les accions que ha de fer la TTP), però pot representar un inconvenient per a les parts contractants (en aquest cas per a B). D'una banda, B ha de tenir el rellotge sincronitzat amb la TTP, ja que si no, assumeix el risc que la petició sigui rebutjada per haver arribat fora de termini. D'altra banda, encara que els rellotges estiguin sincronitzats, en cas que surtin problemes amb el canal de comunicacions, l'equitat de l'intercanvi es pot veure compromesa per a B, per la impossibilitat de contactar amb la TTP dins del termini establert.

Una altra solució amb TTP *off-line*

Arribats a aquest punt, en el qual ja s'han presentat dues solucions (síncrona i asíncrona) per a la signatura electrònica de contractes, es podria plantejar la qüestió de per què un protocol amb tres passos: seria possible un protocol en dos passos? La resposta és que no, perquè la TTP no tindria manera de prendre una decisió que garantís l'equitat a la part que estigui actuant de manera honesta.

D'altra banda, es podria pensar en una solució en quatre passos. Òbviament, l'eficiència disminuiria, però potser s'aconseguiria alguna millora respecte al protocol de tres passos. Un protocol de quatre passos seria de la manera següent:

A → B:M,$Sign_A(M)$
B → A: $Sign_B(M)$
A → B: $Sign_A[Sign_B(M)]$
B → A: $Sign_B(Sign_A[Sign_B(M)])$

Es pot observar que s'ha aconseguit una certa simetria entre les dues parts que han de signar el contracte. Ara la prova per a A és la signatura de B i l'acusament de recepció de B, tal com per a B la prova és la signatura de A i l'acusament de recepció de A. Quedaria ara per veure com queden alterats els subprotocols per als casos de conflicte.

Noteu que ara el que ràpidament pot quedar en desavantatge és A (a diferència d'abans), perquè un cop que A ha enviat l'acusament de recepció queda en mans de B. Per això hem d'establir un protocol de finalització per a A amb la TTP:

A → T: M,$Sign_A(M)$,$Sign_B(M)$,$Sign_A[Sign_B(M)]$
T → A: $Sign_T(Sign_A[Sign_B(M)])$

De manera anàloga al que hem explicat en el primer protocol optimista, ara seria B el que podria quedar en una situació no equitativa, si no dissenyem un subprotocol de finalització per a ell. Es podria donar el cas que A fes trampes, i sense haver enviat el tercer missatge contactés amb la TTP. D'aquesta manera A aconsegueix les proves que necessita (la signatura de B i l'acusament de recepció de T) i B només té la signatura de A, que no és suficient. Per això necessitem el subprotocol següent:

B → T: M,$Sign_A(M)$,$Sign_B(M)$
T → B: $Sign_T(Sign_A[Sign_B(M)])$

Ara hauríem de decidir si volem que la solució sigui síncrona (establint un termini per a poder contactar amb la TTP), o si volem una solució asíncrona (i llavors hauríem de dissenyar un subprotocol de cancel·lació per a B, amb l'objectiu que no hagi d'esperar indefinidament a veure què fa A).

Aquest protocol en quatre passos (tant la versió síncrona com l'asíncrona) introdueix una millora respecte de les versions en tres passos, i és que el comportament de la TTP és verificable, és a dir, que si la TTP intenta fer trampes, aquesta situació podrà ser detectada i demostrada.

2. Solucions per a la signatura electrònica de contractes sense TTP

Si hem dit que les solucions amb TTP *on-line* o *in-line*, no són convenients perquè la TTP es pot convertir en un coll d'ampolla pel que fa a les comunicacions, o no volem assumir el cost que pugui comportar la mateixa crítica, encara que en menys grau, es pot adreçar a les solucions amb TTP *off-line*. Per això cal preguntar-nos si hi ha propostes de solucions sense TTP. La resposta és que sí, i que un dels tipus de solucions sense TTP es basa en el que es coneix com l'intercanvi gradual de secrets. De manera senzilla podem dir que la idea consisteix a intercanviar la signatura sobre el contracte a trossos (a continuació veurem que això és una simplificació poc elaborada).

Protocol de cara o creu

Per explicar un exemple concret de protocol basat en l'intercanvi gradual de secrets, primer hem d'explicar un protocol per a jugar a cara o creu a través de les xarxes de comunicacions. Per a això suposem que dos usuaris volen jugar al joc de cara o creu en versió electrònica i un d'ells, l'iniciador (Alice), disposa d'un parell de claus de criptografia asimètrica, PR_1-PU_1 i PR_2-PU_2. En primer lloc, remet els components públics a l'altre jugador (Bob):

$$\text{A} \rightarrow \text{B: } PU_1, PU_2$$

A continuació B genera una clau de criptografia simètrica (k), i tria de manera aleatòria una de les dues claus públiques que va rebre procedents de A, que etiquetarem com a PU_j. Amb la clau pública triada xifrarà la clau k, i transmetrà el resultat a A:

$$\text{B} \rightarrow \text{A: } PU_j(k)$$

Ara A ha d'escollir de manera també aleatòria (no sap quina ha triat B) una de les dues claus privades. Suposem que tria PR_1 i fa el desxifratge amb aquesta clau del que ha rebut de B:

$$X = PR_1(PU_j(k))$$

Noteu que si $j = 1$ llavors X és exactament el valor de la clau triada per B:

$$X = PR_1(PU_1(k)) = k$$

I al contrari, si $j = 2$ llavors el desxifratge donarà com a resultat un valor arbitrari:

$$X = PR_1(PU_2(k)) = ?$$

La probabilitat que A es trobi en una situació o l'altra és del 50%, perquè A no té manera de conèixer la clau pública que ha triat B (perquè A no coneix la clau k que ha generat B).

A continuació A xifra un missatge "creu" amb el valor obtingut X, i transmet el resultat a B:

$$\text{A} \rightarrow \text{B: } E_X(creu)$$

B fa l'operació de desxifratge:

$$R = D_k(E_X(creu))$$

En el cas que j sigui 1 (B ha triat la primera clau pública) tindrem que X es correspon amb k, i per tant:

$$R = D_k(E_k(creu)) = creu$$

En cas contrari ($j = 2$) tindrem que:

$$R = D_k(E_X(creu)) = ????$$

I per convenció suposarem que aquest resultat es correspon amb la "cara".

Per finalitzar l'execució del protocol A i B intercanvien la informació següent:

A → B: PR_1,PR_2
B → A: k

Perquè tots dos puguin comprovar que cap dels dos no va fer trampa, és a dir, que el 50% de les vegades sortirà cara i el 50% de les vegades creu, sense que cap de les dues parts pugui saber quin serà el resultat de l'execució del protocol (ni influir-hi).

Més enllà del joc de "cara o creu" el que interessa del protocol anterior és que una de les parts, A, ha proporcionat a l'altra un de dos possibles valors (cara o creu) amb la mateixa probabilitat, i sense que A sàpiga quin dels dos li ha proporcionat. Aquesta característica es denomina transferència inconscient. Per a explicar un protocol de signatura de contractes sense TTP, suposarem que disposem d'un protocol similar, que etiquetarem com a:

$$\text{A} \rightarrow \text{B: } trans - transc(x,y)$$

que significa que A transmet a B un de dos possibles valors (x o y), de manera equiprobable i sense que A sàpiga quin dels dos ha transferit.

Protocol de signatura de contractes sense TTP

Expliquem aquí un protocol de signatura de contractes sense TTP; és a dir, en l'execució d'aquest protocol només intervindran els dos signants del contracte.

El protocol l'inicia A generant de manera aleatòria $2N$ claus de criptografia simètrica (per exemple, d'AES o DES), i les agrupa formant parelles:

$$\text{A: } (a_1,a_{N+1}),(a_2,a_{N+2}),\ldots,(a_N,a_{2N})$$

A continuació xifra un text arbitrari S amb les $2N$ claus que ha generat anteriorment, i envia els $2N$ criptogrames a B:

$$\text{A} \rightarrow \text{B: } C_i = E_{a_i}(S) \text{ (per a } i = 1 \text{ fins a } 2N)$$

A acorda amb B que quedarà vinculat a un contracte M si B pot obtenir una o més de les parelles de claus que ha generat en el primer pas (i que en aquest moment només ella coneix). Aquest pas queda fora de l'explicació detallada del protocol, però apuntem que implicaria enviar un missatge signat, en què s'indica aquesta qüestió.

En aquest moment B procediria de manera anàloga als passos que ha fet A. En primer lloc, generaria $2N$ claus de criptografia simètrica i les agruparia en parelles:

$$\text{B: } (b_1,b_{N+1}),(b_2,b_{N+2}),\ldots,(b_N,b_{2N})$$

A continuació també xifra el text arbitrari S amb les $2N$ claus que ha generat anteriorment, i envia els $2N$ criptogrames a A:

$$\text{B} \rightarrow \text{A: } D_i = E_{b_i}(S) \text{ (per a } i = 1 \text{ fins a } 2N)$$

B també acorda amb A que quedarà vinculat a un contracte M si A pot obtenir una o més de les parelles de claus que ha generat (i que en aquest moment només B coneix).

En el pas següent és quan necessitem el protocol de transferència transcordada que hem apuntat anteriorment. A i B l'utilitzaran per a transmetre una de les dues claus de cada un dels N parells de claus que han generat respectivament. Comencem per A:

$$\text{A} \rightarrow \text{B: } trans-transc(a_i,a_{i+N}) \text{ (per a totes les parelles de claus)}$$

En finalitzar aquest pas, A haurà transferit a B una de les dues claus de cada parell, amb dues característiques molt importants per a la seguretat del protocol: A no sap quina de les dues claus de la parella ha proporcionat a B, i el fet que n'hagi proporcionat una o l'altra (de cada parell) és equiprobable.

B ha de fer el mateix pas amb els seus parells de claus:

$$B \rightarrow A: \textit{trans} - \textit{transc}(b_i, b_{i+N}) \text{ (per a totes les parelles de claus)}$$

En aquest punt de l'execució del protocol A i B disposen de la meitat dels secrets de l'altra part. Això no representa un compromís per a cap dels dos, ja que han acordat que per considerar signat el contracte han de disposar com a mínim d'un dels parells de les claus (per tant, en aquest moment no disposen de prou informació).

A continuació A i B intercanvien bit per bit totes les claus que van generar en la primera fase del protocol. Aquest intercanvi l'han de fer de forma intercalada, de manera que en tot moment tots dos disposin aproximadament de la mateixa informació de l'altra part. Òbviament, la part que inicïi l'intercanvi estarà en desavantatge, però si l'intercanvi és bit per bit aquest desavantatge serà molt petit. Vegem com quedaria el protocol:

FOR i = 1 TO L DO (en què L és la longitud de cada clau)
A → B: el bit *i* de totes les claus a_i
B → A: el bit *i* de totes les claus b_i

Si cap de les dues parts atura l'execució d'aquest pas, al final A i B disposaran dels *N* parells de claus de l'altra part. Totes dues parts poden verificar que els parells són correctes fent el desxifratge dels criptogrames que han intercanviat a l'inici de l'execució del protocol. N'hi havia prou amb un parell de claus per a considerar signat el contracte, i per tant, el contracte quedarà signat.

Analitzem ara què passa si una de les dues parts intenta fer trampes. El primer intent de fer un frau es podria produir en el moment de transferir una de les dues claus amb la transferència inconscient, és a dir, intentar enviar una clau que no es va utilitzar per a xifrar el text arbitrari *S*. D'aquesta manera, la part fraudulenta el que intenta és obtenir un parell de claus de l'altra part, i deixar-la sense la possibilitat d'obtenir els seus parells de claus. La transferència inconscient no permet aquest atac, ja que és clar que l'atacant no pot canviar les dues claus. Una de les dues és transferida en el pas de transferència inconscient, i per tant, si les dues claus són "falses" l'altre extrem ho detectarà ràpidament, ja que aquesta clau no es correspondrà amb cap dels criptogrames rebut en el primer pas del protocol.

Per tant, l'atacant només pot canviar un dels dos components de cada parell. Però tampoc no pot fer aquest atac, ja que recordeu que es transfereix una clau de cada parell de manera inconscient, és a dir, l'emissor no sap quin dels dos components ha rebut l'altre extrem. Si l'emissor canvia un dels dos components aleatòriament, en un parell té el 50% de probabilitats de ser detectat per l'altra part (que s'hagi transferit la clau canviada). Ara s'observa la importància que siguin *N* parells de claus. Hem vist que la probabilitat de no ser detectat en una transferència és del 50% (que tingui la sort que l'altre extrem

rebi la clau correcta, la no canviada). En un segon parell de claus, la probabilitat de no ser detectat també serà del 50%, però la probabilitat acumulada de no ser detectat és del $0,5 \cdot 0,5 = 0,25$, és a dir, el 25%, i així successivament, i arribem a una probabilitat per a N parells de:

$$\text{Prob\_no\_detecció} = 2^{-N}$$

Per a un valor relativament petit de N la probabilitat de no ser detectat es pot fer menyspreable. Com a conclusió, no és possible aquesta via d'atac.

Una segona possibilitat de fer trampes que tenen A i B és intentar enviar bits incorrectes durant la fase d'intercanvi de les claus bit per bit. Veiem que tampoc no és possible aquest atac. En cada pas de l'intercanvi bit per bit de les claus, A i B proporcionen un bit de cada clau de les $2N$ claus que han generat en el primer pas del protocol. Però en aquest punt l'altre extrem disposa d'una de les dues claus de cada parell de manera segura (com acabem de veure). Si A o B intenten enviar bits incorrectes de les dues claus de cada parell, seran immediatament detectats, ja que l'altra part disposa d'una de les dues.

Si intenten enviar bits incorrectes d'una de les dues claus de cada parell, també seran detectats, ja que no saben de quina de les dues claus disposa l'altra part. Si una part detecta que rep bits incorrectes, aturarà immediatament l'execució del protocol per no quedar en situació de desavantatge (proporcionar un parell a l'altra part i quedar-se sense la possibilitat de disposar d'un parell de l'altra part). Per tant, tampoc no és possible aquest atac.

Com a conclusió, podem afirmar que A i B no poden fer cap dels dos atacs plantejats, sense assumir un risc molt elevat que l'intent de fer trampes sigui detectat.

Es podria concloure que el protocol presentat és prou segur, i sense necessitat de TTP, però per a això cal assumir que totes dues parts disposen de la mateixa potència de càlcul. Per a observar la necessitat d'aquesta assumpció, ens hem de plantejar què passa si una de les dues parts atura l'execució del protocol en la fase d'intercanvi de bits quan ja s'han intercanviat X bits de cada clau. És cert que cada part disposa aproximadament de la mateixa informació que l'altra (com a molt hi ha la diferència d'un bit més per a B, si A és el primer que transmet els seus bits).

Un cop aturada l'execució del protocol, l'única possibilitat que queda a totes dues parts és el que es coneix com un atac per força bruta, és a dir, provar les combinacions de bits possibles dels que no han estat proporcionats per l'altra part. Si han aturat l'execució quan només faltava un bit, totes dues parts podran deduir el bit que falta sense més problema, ja que només han de fer dues proves possibles (0 o 1). Però si resten t bits, han de fer 2^t proves, i en funció de t això pot requerir una elevada potència de càlcul perquè l'atac pugui fructificar. Aquí és on hi ha el problema del protocol: el que per a una part

pot ser un problema trivial (disposa de la potència de càlcul necessària per a fer les 2^t proves en un temps raonable) per a l'altra part pot ser un problema irresoluble (pot necessitar temps i recursos de què no disposa). Suposar que les parts que intervenen en la signatura d'un contracte disposen de la mateixa potència de càlcul és poc realista i perillós des del punt de vista de la seguretat (compareu la potència de càlcul d'un particular enfront de la potència de càlcul de Google). Per aquest motiu, no se solen recomanar les solucions sense TTP, malgrat l'avantatge aparent de no necessitar la intervenció d'actors diferents dels signants del contracte.

Les solucions sense TTP basades en l'intercanvi gradual de secrets se solen descartar perquè per garantir-ne la seguretat fan l'assumpció que les parts disposen de la mateixa potència de càlcul, assumpció que és poc realista en la pràctica.

3. Solucions per a la signatura electrònica de contractes multipart

Fins ara hem presentat solucions per al cas que siguin dues les parts contractants, però també es pot donar el cas que siguin més de dos usuaris els que hagin de signar un mateix contracte. Parlem així de protocols multipart per a la signatura electrònica de contractes.

Solució per a la signatura electrònica de contractes multipart amb TTP *in-line*

Com en el cas dels contractes entre dues parts, començarem amb la solució més senzilla possible. Es tracta d'una solució amb TTP *in-line*, en la qual totes les parts intervinents (C_1 a C_N) envien la còpia signada a la TTP:

$$C_i \rightarrow \text{T: } Sign_i(M)$$

Un cop que la TTP disposa de les còpies de totes les parts, en verifica la correcció, i en cas que siguin totes correctes, ha de retransmetre les $N - 1$ còpies de les altres parts a cada participant:

$$\text{T} \rightarrow C_i\text{: } Sign_j(M) \text{ (per a } j = 1 \text{ fins a } N, \, j \text{ diferent de } i)$$

Amb aquesta senzilla solució aconseguim la propietat més important de la signatura electrònica de contractes: l'equitat. Però aquesta solució comparteix els mateixos defectes que l'equivalent per a dues parts, especialment el possible cost d'una TTP que ha d'intervenir en totes les execucions del protocol, i amb un cost computacional considerable.

Solució per a la signatura electrònica de contractes multipart amb TTP *off-line* síncrona

Continuem considerant que les solucions optimistes són preferibles, ja que reduir la intervenció de la TTP ha de significar reduir el risc que es converteixi en un coll d'ampolla, o com a mínim reduir els costos associats. Per això a continuació exposem una solució optimista síncrona, la qual cosa significa que s'estableix un termini, una data límit per a portar a terme la signatura del contracte. De fet, veurem que la data límit, més que amb la signatura del contracte directament, té a veure amb la intervenció de la TTP.

Ja hem vist que un protocol optimista ha de disposar d'un subprotocol en el qual només intervenen els signants (subprotocol d'intercanvi) i un subprotocol (o més d'un) en el qual ha d'intervenir la TTP.

Començarem explicant el subprotocol d'intercanvi:

$$C_i \rightarrow C_j\text{: } Sign_i(M) \text{ (per a } i,j = 1 \text{ fins a } N, \ i \text{ diferent de } j)$$

En un primer pas cada signant ha d'enviar l'exemplar signat a totes les altres parts. A continuació cada signant ha d'enviar un acusament de recepció a la resta de signants, per confirmar que ha rebut la còpia signada dels altres:

$$C_i \rightarrow C_j\text{: } Sign_i(Sign_j(M)) \text{ (per a } i,j = 1 \text{ fins a } N, \ i \text{ diferent de } j)$$

Els acusaments de recepció són necessaris com a element de prova per a demostrar que el contracte està signat. Per tant, no n'hi ha prou de rebre les còpies signades, sinó que també són necessaris els acusaments de recepció.

Com en el cas de la signatura entre dues parts, ara cal proporcionar un subprotocol per al cas que alguna de les parts no acompleixi el que està previst en el subprotocol d'intercanvi. Establim que les parts tenen un temps t per a contactar amb la TTP, temps després del qual la TTP no acceptarà peticions de resolució amb relació a aquest contracte, i l'únic que farà és notificar la resolució a la qual va arribar abans d'aquest període, si escau.

Si una part ha enviat els acusaments de recepció, i no està rebent els d'una o més de les contraparts, ha d'enviar a la TTP totes les proves acumulades de l'intercanvi, que com a mínim han de contenir les signatures sobre el contracte de la resta, ja que en cas contrari aquest usuari no hauria d'haver enviat els acusaments de recepció:

$$C_i \rightarrow \text{T: proves acumulades}$$

Les proves aportades per C_i demostren que totes les parts s'han compromès a signar el contracte, i per tant saben que tenen iniciada l'execució del protocol. Essent així, la TTP ha de proporcionar un acusament de recepció equivalent als que haurien d'haver enviat les altres parts contractants a C_i:

$$\text{T} \rightarrow C_i\text{: } Sign_T \text{ (acusament de recepció equivalent)}$$

Si alguna de les altres parts tampoc no ha rebut tots els acusaments de recepció, també sap que ha de contactar amb la TTP abans que venci t, sota el risc de quedar-se sense proves de la signatura del contracte.

És senzill veure que el protocol proposat compleix la propietat d'equitat.

Solució per a la signatura electrònica de contractes multipart amb TTP *off-line* asíncrona

De la mateixa manera que hem presentat una solució asíncrona per al cas de dos signants, també hi ha propostes de solució per al cas que siguin múltiples

els signants. Les solucions per al cas de múltiples signants solen ser complexes i, per això, en lloc de proporcionar una solució genèrica per al cas de N signants, explicarem una solució per a un cas particular de tres signants.

El protocol consta de tres subprotocols, com en el cas de dues parts: un subprotocol d'intercanvi, un de cancel·lació i un de finalització. Començarem descrivint el subprotocol d'intercanvi. L'intercanvi d'informació entre els actors es podria fer de múltiples maneres. Una podria controlar l'intercanvi, i adoptar una topologia en estrella. També es podria optar per una solució similar a la del cas síncron, és a dir, que tots envien la informació a tots. Aquí optem per utilitzar una topologia en anell, en la qual cada part assumeix una posició en l'anell, rep una informació del predecessor, i transmet part d'aquesta informació i la que ell genera a l'usuari que va després.

En aquest escenari apareixerà un actor nou C (Charles). Vegem com és el subprotocol d'intercanvi en la topologia en anell:

A → B: M,$Sign_A(M,1)$
B → C: M,$Sign_A(M,1)$,$Sign_B(M,1)$
C → A: $Sign_B(M,1)$,$Sign_C(M,1)$
A → B: $Sign_C(M,1)$,$Sign_A(M,2)$
B → C: $Sign_A(M,2)$,$Sign_B(M,2)$
C → A: $Sign_B(M,2)$,$Sign_C(M,2)$
A → B: $Sign_C(M,2)$,$Sign_A(M,3)$
B → C: $Sign_A(M,3)$,$Sign_B(M,3)$

El subprotocol consta de tres rondes, i les proves que s'intercanvien en cada transmissió incorporen un índex que indica a quina ronda corresponen. Veurem que això serà necessari perquè la TTP pugui resoldre les peticions de resolució, en cas que sigui necessària la seva intervenció. Les proves que el contracte està signat són l'acumulació de les signatures fetes i rebudes dels altres actors.

Es pot demostrar que en un escenari de signatura electrònica de contractes asíncrona i amb TTP *off-line* són necessàries més de $N - 1$ rondes si tenim N participants, i per això és complex explicar la solució per al cas genèric.

Reprenent l'explicació del cas de tres usuaris, si cap de les parts no atura l'execució del subprotocol d'intercanvi, i si no es presenten problemes de comunicacions, cada part disposarà de les proves necessàries per a demostrar que el contracte està signat, i sense necessitat que intervingui la TTP.

Però com en casos anteriors, ara ens hem de plantejar què passa si sorgeixen problemes de comunicacions, o alguna de les parts intenta fer trampes. Òbviament, els contractants honestos han de contactar amb la TTP per recuperar l'equitat de l'intercanvi. Però, quines són les regles que ha de seguir la TTP per garantir l'equitat de totes les parts honestes?

Construirem els subprotocols de cancel·lació i finalització pas per pas (simplificant o unint casos sempre que sigui possible), considerant les possibles situacions que es poden donar. Des del moment que A, B i C tinguin proves del compromís de les altres dues parts, podran acudir a la TTP perquè aquesta pugui donar per finalitzat l'intercanvi. És a dir, en tots els casos següents:

A → T: $Sign_B(M,1)$,$Sign_C(M,1)$
B → T: $Sign_A(M,1)$,$Sign_C(M,1)$
C → T: $Sign_A(M,1)$,$Sign_B(M,1)$
B → T: $Sign_A(M,1)$,$Sign_C(M,1)$,$Sign_A(M,2)$,$Sign_C(M,2)$
C → T: $Sign_A(M,1)$,$Sign_B(M,1)$,$Sign_A(M,2)$,$Sign_B(M,2)$

la TTP els torna un acusament de recepció, que indica que es considera signat el contracte M:

$$T \rightarrow A/B/C: Sign_T(M)$$

Aquesta decisió de la TTP és irrevocable, és a dir, un cop que la TTP hagi decidit que el contracte es considera signat, no hi ha cap petició posterior a la TTP que pugui fer variar aquesta decisió. Aquesta regla de la TTP es basa en el fet que les proves aportades per les parts demostren que tots havien manifestat la intenció de signar el contracte, i que, per tant, si no havien rebut les proves necessàries de la resta sabien que havien de contactar amb la TTP, que els aporta la prova que el contracte es considera signat.

Queden per resoldre les possibles peticions de A i B, en el cas que afirmin que han enviat els missatges de la primera ronda i no rebin les proves que esperen a canvi. Com que es tracta d'un protocol asíncron, però en el qual no volem que les parts implicades hagin d'esperar un temps indefinit per saber si el contracte serà signat o no, hem de permetre que contactin amb la TTP:

A → T: $Sign_A(M,1)$
B → T: $Sign_A(M,1)$,$Sign_B(M,1)$

Amb la informació anterior, i si ningú no havia contactat prèviament amb la TTP per finalitzar l'intercanvi (decisió que hem dit que és irrevocable) la TTP no pot fer res més que cancel·lar l'intercanvi:

$$T \rightarrow A/B: Sign_T(signatura-de-M-cancel{\cdot}lada)$$

No té sentit permetre que C pugui cancel·lar l'intercanvi, perquè quan C pot contactar amb la TTP ja té prou informació perquè la TTP pugui decidir que s'ha de considerar signat el contracte (vegeu el cas anterior).

A diferència de la decisió de la TTP que el contracte està signat, la de cancel·lació sí que la pot revocar la TTP. Això és perquè podria ser que A o B estiguessin fent trampes i això es pogués detectar *a posteriori* (amb la informació

aportada per altres parts). Per a això hem d'analitzar les possibles situacions de contacte amb la TTP que es puguin donar després d'una cancel·lació de A o B. Farem l'anàlisi per al cas en què A és la primera part que cancel·la (queda com a exercici el cas en què sigui B el primer que cancel·la):

- A cancel·la, B cancel·la, C finalitza amb la informació de la primera ronda o no contacta amb la TTP

  En aquest cas, i en tots els que segueixen, davant la petició de A, la TTP inicialment considera l'intercanvi cancel·lat. La petició de cancel·lar de B és compatible amb l'anterior de A, i per tant, la TTP continua mantenint l'estat de l'intercanvi com a cancel·lat. Finalment, la petició de C també és compatible amb l'anterior (pot ser que C hagi enviat la informació de la primera ronda, però per problemes amb el canal de comunicacions aquesta informació no va arribar a A), i per tant la TTP manté l'estat de l'intercanvi a cancel·lat.

- A cancel·la, B cancel·la, C finalitza amb la informació de la segona ronda

  En aquest cas, la petició de C demostra a la TTP que A i B havien rebut la informació que C havia enviat, perquè en cas contrari A i B no hauri-en proporcionat a C la informació de la segona ronda. Essent així la TTP conclou que A i B han mentit i canvia l'estat de l'intercanvi a finalitzat i proporciona la signatura del contracte a C:

  $$\mathrm{T} \rightarrow \mathrm{C}\text{: } Sign_T(M)$$

- A cancel·la, B finalitza amb la informació de la segona ronda, independentment del que faci C

  Com en el cas anterior, el fet que B aporti proves de la segona ronda demostra que A va fer trampes (disposava de més informació de la que va aportar a la TTP, ja que en cas contrari no hauria enviat les proves de la segona ronda a B). La TTP considerarà el contracte signat:

  $$\mathrm{T} \rightarrow \mathrm{B}\text{: } Sign_T(M)$$

- A cancel·la, C finalitza amb la informació de la primera ronda, B cancel·la o no contacta amb la TTP

  Com en el primer cas analitzat, el fet que C aporti les proves de la primera ronda és compatible amb la petició de A, i per tant la TTP mantindrà l'estat de l'intercanvi com a cancel·lat.

- A cancel·la, C finalitza amb la informació de la segona ronda, independentment del que faci B

Les proves que aporta C de la segona ronda demostren que A va fer trampes (disposava de més informació de la que va aportar a la TTP, ja que en cas contrari no hauria enviat les proves de la segona ronda a B, i aquest a la vegada a C). La TTP considerarà el contracte signat.

- A cancel·la, C finalitza amb la informació de la primera ronda, B contacta amb la informació de la segona ronda

  Com ja hem dit, la informació aportada per C és compatible amb la petició inicial de A, i per tant la TTP manté l'estat de cancel·lat, i envia a C la cancel·lació:

  $$\mathrm{T} \rightarrow \mathrm{C}: \mathit{Sign}_T(\mathit{signatura-de-M-cancel{\cdot}lada})$$

  A continuació B contacta amb la TTP i li aporta una informació que demostra que A ha fet trampes (tenia més informació de la que va aportar), però la TTP no pot revocar la decisió presa, perquè en cas contrari podria estar perjudicant C. Per tant, malgrat que A havia fet trampes, com pot ser que C no hagi fet trampes, la TTP manté l'estat de cancel·lat i envia aquesta decisió a B.

L'últim cas (l'encadenament de cancel·lacions) és el que justifica la necessitat de tres rondes. En cap cas no s'ha de permetre que una part que sigui honesta perdi l'equitat de l'intercanvi per una decisió incorrecta de la TTP. Es pot provar fàcilment que amb dues rondes la intervenció de la TTP no permetria resoldre adequadament totes les situacions possibles.

Per finalitzar, podem concloure que el protocol presentat garanteix l'equitat per a les parts honestes, i és del tipus optimista i asíncron.

Exercicis d'autoavaluació

1. Convertiu la solució amb TTP *in-line* per a la signatura electrònica de contractes entre dues parts, en una solució amb TTP *on-line*.

2. Verifiqueu si en la solució per a la signatura electrònica de contractes (entre dues parts) amb TTP *off-line* de quatre passos, efectivament el comportament de la TTP és verificable.

3. Proporcioneu un protocol per a la signatura electrònica de contractes amb TTP *off-line* en el cas en què siguin quatre les parts que han de signar el mateix contracte.