Aplicació descentralitzada per fer revisions peer-review de manera anònima

Autors: Helena Rifà i Joan Manuel Marquès

Descripció general de l'aplicació

Volem construir una aplicació descentralitzada per a fer revisions cegues (peer-review) que siguin anònimes de manera que no hi hagi cap servidor central que gestioni la informació de les revisions.

El sistema ha de garantir l'anonimat del servei de manera que un autor no pugui saber qui li revisarà el treball, un revisor no conegui de qui són els treballs que revisa, i els nodes que formen part de la xarxa distribuïda no puguin trobar el vincle entre l'autor i el revisor dels missatges que gestionen.

Utilitzarem l'aplicació en un context docent on hi haurà el rol de professor i el rol d'alumne.

El professor podrà saber l'autoria de qualsevol treball i qualsevol revisió.

Quan s'hagi acabat el termini d'introducció de revisions de treballs, es faran públics els treballs – fent-ne públic l'autor – i les revisions – que continuaran sent anònimes.

Context

- Ha de funcionar en un conjunt d'ordinadors contribuïts de manera voluntària a una comunitat.
- Disposem d'un sistema (CoDeS) que permet desplegar serveis en ordinadors contribuïts voluntàriament a una comunitat.

Consideracions

- Els ordinadors estaran connectats durant un temps que no es coneix a priori, encara que suposem que hi haurà un conjunt prou gran d'ordinadors que estarà connectat prou temps per a que el sistema funcioni.
- Qualsevol ordinador es pot desconnectar en qualsevol moment de manera abrupta (sense avisar ni seguir cap procediment de desconnexió)
- El sistema de desplegament (CoDeS) garanteix que si l'ordinador que hostatja un servei es desconnecta, el servei es reubica de manera automàtica en un altre ordinador contribuït a la comunitat.
    - Pot haver-hi més d'una instància del servei. És responsabilitat del servei mantenir les diferents instàncies sincronitzades per a garantir que no es perdi informació en cas que una de les instàncies deixi d'estar activa.
- Interessa que els serveis necessitin el mínim possible de recursos per garantir que es podran executar en els recursos donats pels membres de la comunitat.
    - Com a norma general és millor tenir més serveis que necessitin menys recursos que tenir menys serveis però que aquests necessitin més recursos.
- Cada usuari, tant professors com alumnes, tindrà un pseudònim que l'identificarà unívocament i que no es podrà falsejar.
- Cada treball tindrà un identificador únic que no tindrà relació amb la identitat de l'autor del treball i que s'assignarà en el moment de lliurar el treball.

Funcionament de l'aplicació

Les operacions han de ser anònimes.

Els alumnes no han de poder saber qui ha fet ni el treball ni cap de les seves revisions (que no siguin les seves aportacions). Tampoc han de poder saber què està publicant, revisant o llegint cap dels usuaris. També cal protegir-se contra atacs d'anàlisi del tràfic de xarxa per interceptar les accions dels usuaris i així saber qui ha fet el treball o revisió.

Els professors han de poder accedir en tot moment a qualsevol treball i a qualsevol revisió. També han de poder saber en tot moment qui ha fet cada treball i cada revisió. Les operacions dels professors també són anònimes, en el sentit que ningú ha de poder saber quan i com realitzen una operació (sinó algú que estigui espiant sabria quines Entity són reals i quines pot ser que no tinguin dades).

Funcionalitats

- publicació d'un treball: permet que un alumne publiqui de manera anònima el seu treball.
- obtenció d'un treball: permet a un alumne obtenir un treball de manera anònima.
- introducció revisió d'un treball: permet que un alumne introdueixi de manera anònima les revisions dels treballs que ha de revisar.
- obtenir revisions d'un treball: permet a un alumne autor d'un treball obtenir de manera anònima les revisions fetes sobre aquest treball. També permet a un alumne que hagi revisat d'un treball obtenir la revisió que ell ha fet sobre aquest treball.
- obtenció de tots els treballs amb les seves revisions: permet que un professor accedeixi de manera anònima a cadascun dels treballs i a cadascuna de les revisions fetes de cada treball.

Arquitectura de l'aplicació

La funcionalitat s'oferirà a partir de les següents aplicacions i serveis:

- *Entity*: servei desplegat a CoDeS que gestionarà un treball i les revisions fetes sobre aquest treball.
    - Cada Entity haurà d'estar replicat en vàries instàncies per garantir la disponibilitat de la informació en cas de fallades.
- *User's Application*: cada alumne tindrà instal·lada una aplicació que li ha de permetre:
        - introduir el seu treball al sistema
        - obtenir un treball
        - introduir una revisió
        - obtenir les revisions fetes sobre els seus treballs.
    - Aquesta aplicació pot ser una aplicació java senzilla o un servidor web amb una interfície html i uns servlets que siguin els que envien els missatges. Tant en un cas com en l'altre s'instal·laria a l'ordinador de l'alumne.
- *Directory*: servidor ubicat a la UOC (extern a CoDeS) que conté els pseudònims dels usuaris del sistema i els identificadors únics dels treballs. També s'utilitza per fer les assignacions entre usuaris i Entities.
    - Internament guarda la relació entre usuaris i identificadors. Aquesta relació només ha de poder ser visible pels professors.

- *Instructor's Application*: aplicació web ubicada a un servidor de la UOC que permet als professors accedir de manera anònima però coneixent-ne l'autoria a tots els treballs i a totes les revisions de cada treball.

Per poder posar en funcionament els serveis descrits el sistema requereix els serveis d'una infraestructura de clau pública (PKI). En concret s'haurà de desplegar una autoritat de certificació:

- *Certification Authority (CA)*: servidor ubicat a la UOC (extern a CoDeS) que certifica la identitat dels alumnes, professors i serveis que participen en la publicació i revisió de treballs.

Protocol anonimat

Inicialització

Inicialització de la CA i el Directory

1. Crear un certificat autofirmat per la CA amb una parella de claus RSA de 2048 bits. El Distinguished Name (DN) de la CA serà: "CN=CA, OU=CoDeS, O=UOC, C=ES"
2. La CA genera una parella de claus RSA de 2048 bits i emet un certificat a nom de: "CN=Directory, OU=TFM_PLL, O=UOC, C=ES". Les claus, el certificat, i el certificat de la CA s'encapsulen en un repositori de claus de tipus PKCS#12 o bé Java Key Store (JKS).
3. Quan es crea la instància de Directori, s'hi instal.la el repositori de claus del punt anterior.

Inicialització de les Entities

1. En iniciar una experiència de peer-review anònim el Directory crea $N=A*R+M$ identificadors aleatoris, on A és el nombre d'alumnes de l'aula i R i M serveixen per a que el nombre final identificadors sigui prou gran per:
    - millorar l'anonimat pel fet de garantir que hi ha un nombre mínim d'Entities desplegades.
    - que no s'acabin els identificadors degut a fallades durant el procés de publicació de treballs, que comporten l'assignació d'un identificador a la identitat d'un usuari.

    Aquest identificador s'usarà com a nom de cadascuna de les Entity que es desplegaran a CoDeS.
2. El Directory demana a la CA una clau privada i un certificat per cada identificador creat, amb les dades: "CN=Entity <id>, OU=TFM_PLL, O=UOC, C=ES".
    - La CA retorna un repositori de claus (PKCS#12 o JKS) per cada Entity que incorpora: parella de claus de l'entity, certificat de l'entity, certificat de la CA.
    - El Directory incorpora al repositori de claus de les Entities (rebut en el pas anterior) el certificat del Directory (d'aquesta manera les Entities poden verificar la signatura dels missatges que rebin del Directory ràpidament).
3. Genera una clau simètrica per a cada Entity, que s'usarà per xifra les dades a emmagatzemar.
4. El Directory desplega a CoDeS tantes Entity (amb les seves corresponents rèpliques) com indiqui N.
    - Les rèpliques d'una Entity que gestionen la informació d'un treball comparteixen el mateix nom de l'Entity, que és l'identificador esmentat anteriorment.

Inicialització del User Application

Els alumnes han d'aconseguir un certificat digital i un pseudònim per poder operar amb el sistema de gestió de treballs i revisions.

1. L'Usuari accedeix a l'User Application que té instal·lada en el seu ordinador.
2. L'User Application es connecta a la CA i envia una petició de firma de certificat (de tipus PKCS#10 o CSR) amb les següents dades: "CN=<nom d'usuari de la UOC>, OU=TFM_PLL, O=UOC, C=ES". Les claus tindran una longitud de 1024 bits.
3. La CA emet un certificat X.509 per l'usuari -*userCert*- i envia un correu electrònic a l'usuari a l'adreça <nom d'usuari de la UOC>@uoc.edu amb els fitxers: (1) certificat d'usuari, *userCert,* (2) certificat de la CA, *CACert, (3)* pseudònim de l'usuari, *pseudonym.*

Inicialització de l'Instructor Application

Els professors han d'aconseguir un certificat digital i un pseudònim per poder operar amb el sistema de gestió de treballs i revisions.

1. El professor accedeix a l'Instructor Application.
2. L'Instructor Application es connecta a la CA i envia una petició de firma de certificat (de tipus PKCS#10 o CSR) amb les següents dades: "CN=Instructor, OU=TFM_PLL, O=UOC, C=ES". Les claus tindran una longitud de 1024 bits.
3. La CA emet un certificat X.509 per l'Instructor Application i li retorna un missatge amb les dades: (1) certificat d'Instructor, (2) certificat de la CA, CACert, (3) pseudònim lligat a l'Instructor Application.

Publicació d'un treball

1. L'Usuari introdueix el seu treball a l'User Application que té instal·lada en el seu ordinador.
2. L'User Application contacta amb el Directory a través d'una connexió SSL amb autenticació mútua per a obtenir l'identificador de l'Entity on es publicarà el treball i l'autorització corresponent.
    - El Directory mira els identificadors que té lliures i n'hi assigna un. Aquesta assignació consisteix en associar l'usuari amb l'identificador de l'Entity. Inicialment el Directory etiqueta aquesta assignació com "en curs" (o pendent de confirmació). Quan acabi el procés de publicació del treball rebrà una confirmació que farà l'assignació definitiva.
        - El Directory programa un timeout (per defecte 5 minuts, però hauria de ser un paràmetre de configuració). Si s'esgota el timeout sense haver rebut la confirmació de la correcta realització de l'operació, l'entrada es torna a marcar com a "lliure".
    - En cas que l'usuari ja hagi publicat el seu treball i, per tant, aquesta publicació sigui una modificació del treball ja publicat, l'identificador retornat serà l'identificador on l'usuari ja ha publicat la versió anterior del seu treball.
    - El Directory emet una autorització de Publicació del treball per l'usuari, amb el format:

        $$ps, profile, Id\_Entity, Time, Signature_{pvk\_D}$$

        amb *ps*: pseudònim de l'usuari, *profile*: perfil de l'usuari, en aquest cas autor, *Id_Entity*: identificador de l'entity, *Time*: data d'emissió i expiració de l'autorització, *Signature*: signatura dels camps anteriors amb la clau privada del Directoy.
3. L'User Application executa el Protocol d'Encaminament Anònim posant com a Entity destí l'identificador de l'Entity obtingut en el pas 2.

4. L'Entity destí (que és la responsable del treball) agafa el treball i l'emmagatzema a CoDeS. la informació a publicar ha d'incloure l'identificador de l'alumne que publica el treball per poder comprovar més endavant que l'usuari que consulta o modifica l'operació ho pot fer.
    - La publicació d'un treball sobreescriu qualsevol publicació anterior.
5. Quan el missatge de confirmació arriba a l'User Application, aquest envia un missatge al Directory confirmant que l'operació s'ha fet correctament i a continuació indica a l'usuari que l'operació s'ha realitzat correctament.
    - En cas que la connexió entre l'User Application i la primera Entity del camí es tanqui sense haver rebut la confirmació de la correcta publicació o rebi un missatge indicant que l'operació no s'ha dut a terme correctament, l'User Application envia un missatge al Directory indicant que no es pot assegurar que la publicació s'hagi fet correctament, cosa que fa que el Directory marqui l'entrada com "lliure". Havent-hi el timeout al Directory no seria estrictament necessari enviar aquest missatge però pot anar bé per alliberar l'identificador i, sobretot, per evitar problemes si l'usuari reintenta l'enviament.
        - En aquest cas, la User Application mostra en pantalla a l'usuari un missatge indicant que la publicació no s'ha pogut fer i demana que aquest ho reintenti.
6. El Directory fa l'assignació entre l'identificador i l'usuari que ha publicat el treball definitiva.
    - Mentre no rebi aquesta confirmació o no confirmació o hagi saltat el timeout el Directory té aquesta assignació marcada com "en curs".

Fins que no s'hagi esgotat el termini de publicació de treballs, un usuari pot publicar tantes versions del seu treball com vulgui. El protocol de publicació és el mateix per totes les publicacions.

Obtenció d'un treball

Quan acaba el termini de publicació de treballs, els professors executen un procés al Directory que genera una llista que, per cada alumne que ha publicat un treball, se li assigna un nombre de revisions a fer i una autorització de revisió per obtenir els treballs a revisar i publicar les revisions. Aquest nombre és un paràmetre que introdueix el professor. El valor per defecte són 3 però el professor ho pot canviar. Un cop generada aquesta llista, s'envia a cada alumne per correu electrònic els identificadors dels treballs que ha de revisar i les corresponents autoritzacions.

- En cas que algun alumne no hagi de fer el treball però si hagi de fer les revisions, el professor notificarà manualment a l'alumne els treballs que ha de revisar. El sistema ha de funcionar correctament també si això passa.

Les autoritzacions de revisió tenen el format $<ps, profile, Id\_Entity, Time, Signature_{pvk\_D}>$, amb un valor de *profile* igual a reviewer.

Els passos per l'obtenció del treball són:

1. L'Usuari introdueix a l'User Application que té instal·lada en el seu ordinador:
    - identificador del treball (i.e. l'identificador de l'Entity que conté el treball) que ha de revisar.
    - autorització de revisió.
2. L'User Application executa el Protocol d'Encaminament Anònim posant com a Entity destí l'identificador de l'Entity que conté el treball a revisar.
3. L'Entity entrega els treballs a revisar sempre i quan l'usuari tingui una autorització de revisió vàlida (dades de l'autorització correctes, data no expirada, firma digital vàlida).
4. Quan l'User Application rep la confirmació de la transacció, aquest presenta la informació a

l'usuari.

En cas que el missatge no arribi al cap d'un cert temps, l'usuari pot tornar a fer la petició.

Introducció revisió d'un treball

1. L'Usuari introdueix a l'User Application que té instal·lada en el seu ordinador:
    - identificador del treball (i.e. l'identificador de l'Entity que conté el treball) que ha revisat.
    - autorització de revisió.
2. L'User Application executa el Protocol d'Encaminament Anònim posant com a Entity destí l'identificador de l'Entity sobre el que s'ha fet la revisió.
3. L'Entity on s'ha de posar la revisió agafa la revisió del missatge, comprova que l'usuari està autoritzat a fer la revisió del treball (dades de l'autorització correctes, data no expirada, firma digital vàlida) i la desa a CoDeS.
    - L'enviament d'una revisió sobreescriu qualsevol revisió anterior del mateix usuari.
    - L'Entity desa la revisió i el pseudònim de l'usuari que ha realitzat la revisió.
4. Quan el missatge de confirmació de transacció arriba a l'User Application aquest indica a l'usuari que la revisió s'ha desat correctament.
    - En cas que la connexió entre la User Application i la primera Entity del camí es tanqui sense haver rebut la confirmació de la correcta publicació de la revisió, l'User Application notifica a l'usuari que pot ser que la seva revisió no s'hagi pogut fer correctament, que torni a intentar-ho.
        - Una optimització podria ser reintentar-ho transaparentment (tornar al pas 2)
5. En cas que el missatge no arribi al cap d'un cert temps, l'usuari pot tornar a fer la publicació de la revisió.

Obtenir revisions d'un treball

1. L'Usuari introdueix a l'User Application que té instal·lada en el seu ordinador:
    - identificador de l'Entity de la qual en vol obtenir la revisió.
    - autorització conforme l'usuari és l'autor del treball (obtinguda en la fase de publicació del treball)
2. L'User Application executa el Protocol d'Encaminament Anònim posant com a Entity destí l'identificador de l'Entity de la revisió.
    - L'Entity responsable del treball obté la revisió de CoDeS i l'afegeix al missatge.
        - En cas que l'usuari sigui l'usuari que ha publicat el treball, s'afegeixen totes les revisions que s'han fet del seu treball sense indicar qui les ha fet.
        - En cas que l'usuari sigui un usuari que hagi fet una revisió del treball, s'afegeix al missatge la darrera revisió que aquest hagi fet del treball.
        - En cas que no estigui autoritzar per veure aquestes revisions caldria enviar un missatge d'error i desar una traça indicant que l'usuari ha intentat obtenir revisions d'aquest treball.
3. Quan el missatge de confirmació arriba a l'User Application aquest presenta la informació l'usuari.

4. En cas que el missatge no arribi al cap d'un cert temps, l'usuari pot tornar a fer la petició.

Operacions professors

Els professors poden efectuar les operacions d'Obtenció d'un treball i Obtenció de les revisions d'un treball de forma anàloga a com ho fan els estudiants.

1. El professor es connecta a l'aplicació web Instructor's Application a través d'una connexió HTTPS amb autenticació mútua. El professor introdueix al formulari web l'identificador del treball sobre el que vol fer el seguiment així com l'operació que es vol portar a terme (obtenció de treball, obtenció de revisions, obtenció de treball i revisions).
2. L'Instructor Application es connecta al Directory a través d'una connexió SSL amb autenticació mútua per a obtenir una autorització de professor.
3. L' Instructor's Application executa el Protocol d'Encaminament Anònim posant com a Entity destí l'identificador del treball objecte de la petició.
4. A través de les dades de l'autorització, l'Entity destí coneix que l'emissor de la petició pertany a un usuari amb perfil de professor i per tant, quan torna les dades a l'emissor, també hi incorpora informació sobre el pseudònim de l'autor del treball i dels revisors.
5. L' Instructor's Application contacta amb el Directory a través d'una connexió SSL amb autenticació mútua per aconseguir la identitat dels pseudònims obtinguts en el pas 4.
6. L' Instructor's Application mostra la informació dels treballs, revisions, autors i revisors al professor.

Nota: Si es volen obtenir tots els treballs i tots les seves revisions, es pot fer un bucle on es repeteixin les operacions per cadascun dels alumnes.

Implementació interfícies

User Application

Pot ser tant un aplicació java com un servidor web amb servlets. Cal pensar quina és la millor opció.

S'ha d'executar a l'ordinador de l'alumne.

Tant el treball com les revisions del treball s'han d'enviar en format text.

Hi haurà un espai per introduir cadascuna de les parts del treball i cadascuna de les parts de la revisió.

Instructor Application

Una aplicació web instal·lada en un servidor de la UOC.

Protocol d'Encaminament Anònim

El protocol d'encaminament anònim segueix un format d'*onion routing* en el qual un missatge és xifrat en múltiples capes. Cada capa conté la identitat i un paquet xifrat pel pròxim encaminador de la xarxa.

En concret, els passos del protocol són els següents:

1. L'User Application contacta amb el Directory a través d'una connexió SSL amb autenticació mútua per a obtenir *N* identificadors d'Entities i les seves corresponents claus públiques.
2. L'User Application traça un camí a través de les *N* Entities que ha obtingut del Directory

passant per l'Entity destí, és a dir, el node on vol enviar o d'on vol obtenir les dades.

Per fer-ho, crea una cadena de *M* Entities ($M \geq N+2$) ordenant de forma aleatòria les *N* Entities obtingudes del Directory (1, 2, .. N) i introduint enmig d'aquestes l'Entity destí, una repetició de l'Entity origen (la situada en la posició 1), i, opcionalment, repeticions d'algunes de les *N* Entities de la llista inicial. La posició de l'Entity destí i la repetició de l'Entity origen és aleatòria, però s'ha de complir que la posició del destí precedeixi a la segona aparició de l'origen.

3. L'User Application genera *M* claus simètriques associades a cada una de les Entities de la cadena creada en el pas 2.
4. L'User Application construeix el paquet a enviar començant per el missatge que s'enviarà a la última Entity de la cadena:

   $Paquet_M = Id\_Entity_M, E_{pbk\_M}(k_M, 0), padding$

5. Per la resta d'Entities, excepte l'Entity origen i la destí, de la ruta segueix el procediment següent:

   $Paquet_x = Id\_Entity_x, E_{pbk\_x}(k_x, len_1), E_x(role, Id\_Entity_{x+1}, Paquet_{x+1})$

   amb $E_{pbk\_x}(.)$ el xifrat amb clau pública de la Entity *x*, $E_x(.)$ el xifrat amb clau simètrica *x*, len_1 la longitud del xifrat $E_x(.)$ i *role=router* una indicació que la funció de la Entity és reenviar les dades rebudes a la Id_Entity_{x+1} de continuació.

   La len_1 serveix per determinar on acaba exactament el text xifrat ja que durant la retransmissió del missatge, les Entities intermitges del camí poden afegir dades de farciment (*padding*) als paquets per tal d'alterar la seva aparença i augmentar l'anonimat.

6. Si $Entity_x$ és l'Entity origen:

   $Paquet_x = Id\_Entity_x, E_{pbk\_x}(k_x, len_1), E_x(role, messageId, Id\_Entity_{x+1}, Paquet_{x+1})$

   Amb *role=origen* un paràmetre que indica a l'Entity que ho rep que aquest missatge l'ha originat ell mateix, i que té l'identificador intern *messageId.*

7. Si $Entity_x$ és l'Entity destí:

   $Paquet_x = Id\_Entity_x, E_{pbk\_x}(k_x, len_1), E_x(role, authz, op, len_2, <data>, Id\_Entity_{x+1}, Paquet_{x+1})$

   Amb *role=destiny* un paràmetre que indica que l'Entity que ho rep és el destí d'aquest missatge, *authz* el codi d'autorització de l'operació, *op* la llista d'operació que s'està portant a terme, len_2 la longitud de les dades *data* que es transporten ($len_2=0$ si no es porten dades), i *data* les dades a publicar.

   El paràmetre *op* conté una llista de les següents operacions:

   - *0*: Publicació d'un treball
   - *1*: Obtenció d'un treball
   - *2*: Revisió d'un treball
   - *3*: Obtenció de les revisions d'un treball

8. Finalment, el missatge a enviar és

   $Paquet = Paquet_1, EC, <padding>$

   amb *EC* un codi d'error d'1 byte (inicialment 0), i *padding* és una cadena de farciment opcional.

9. L'User Application fa l'enviament anònim del *Paquet* a la primera Entity, l'Entity origen.

- L'enviament es fa utilitzant l'encaminament de CoDeS, és a dir, el client pregunta a CoDeS per les adreces IP i port de les rèpliques de l'Entity origen i n'escull una, a qui li envia el *Paquet*.
- Quan un node origen rep per primer cop un missatge amb un *messageId* desconegut, deixa oberta la connexió amb l'User Application que ha enviat la petició i engega un timeout.
  - Si abans que expiri el timeout l'Entity origen rep un missatge amb el mateix *messageId* anterior, significa que el paquet ha estat enviat correctament al destí i per tant pot confirmar la transmissió a l'User Application. L'Entity també envia el *padding* del missatge a l'User Application.
  - Si el *timeout* expira sense que l'Entity hagi rebut per segon cop el *messageId*, l'Entity torna un missatge d'error al User Application i tanca la connexió.

10. Quan una $Entity_x$ rep un Paquet, desxifra el text xifrat del missatge utilitzant la seva clau privada pvk_x, i comprova:
    - si $len_1=0$, $Entity_x$ és el node final de la transmissió i no ha de reenviar el paquet a ningú
    - si *role=origen*, $Entity_x$ és el node origen de la transmissió i es comporta tal com hem explicat en el punt 12.
    - si *role=router*, $Entity_x$ és un node retransmissor de la informació. El missatge que la $Entity_x$ ha rebut és:

      $$Paquet = Id\_Entity_x,\ E_{pbk\_x}(k_x,\ len_1),\ E_x(role=router,\ Id\_Entity_{x+1}, Paquet_{x+1}),\ EC,\ <padding_x>$$

      El missatge que enviarà és:

      $$Paquet = Id\_Entity_{x+1}, Paquet_{x+1}, EC,\ <padding_{x+1}>$$

      El $padding_{x+1}$ que el node posarà el missatge serà:

      $$padding_{x+1}=E_x(padding_x), padding$$

      És a dir, el *padding* rebut en el missatge anterior xifrat amb la clau simètrica de l'Entity i, opcionalment, alguna cadena més de farciment.
    - si *role=destiny*, $Entity_x$ és el node destí de la transmissió. En aquest cas l'$Entity_x$ comprova que l'usuari tingui autorització per fer l'operació que vol. En cas que l'usuari no estigui autoritzat, desar una traça indicant que l'usuari ha intentat fer una operació il·lícita i canviar el missatge d'error del paquet per $EC=7$:
      - si el missatge porta dades ($len_2 \neq 0$), l'Entity emmagatzema el camp <*data*>.
      - Si l'operació que es porta a terme és d'obtenció de treballs o revisions, l'Entity incorpora al missatge a reenviar un paquet $padding_{x+1}=E_x(data)$, amb *data* les dades que es volen enviar.

      Si l'operació ha anat bé, posar $EC=1$. En cas contrari, posar el codi d'error corresponent a la incidència ocurreguda (resta a l'estudiant definir els codis d'error).

11. Quan l'User Application rep el missatge de confirmació de transmissió, si el mode d'operació era *Obtenció* (*mode=1*), desxifrarà les dades rebudes fent:

    $$data = D_{x+t}(...D_{x+1}(D_x(padding)))$$

    amb x posició de la Entity destí, i $x+t$ posició de l'Entity origen.

Emmagatzemament de les dades

Cada emmagatzema tota la informació que ha de gestionar a CoDeS, o sigui, les Entity són serveis sense estat. La informació al sistema d'emmagatzemat de CoDeS utilitzant com a identificador de la informació l'identificador de l'Entity. Per tal que només l'Entity responsable de les dades pugui accedir al contingut d'aquestes, les dades es desen xifrades amb la clau pública que es passa com a paràmetre a l'Entity en inicialitzar-la.

En rebre un missatge d'escriptura, l'Entity destí:

1. Obté del sistema de d'emmagatzemat la informació associada a l'Entity.
2. Modifica la informació d'acord amb el contingut del missatge.
3. Desa la informació al sistema d'emmagatzemat.

En rebre un missatge de lectura, l'Entity destí:

1. Obté del sistema de d'emmagatzemat la informació associada a l'Entity.

Per tal que l'accés al sistema de fitxers no delati quina Entity és la destinatària del missatge cal que les Entity del camí anònim també accedeixin al sistema d'emmagatzemat de CoDeS fent peticions de la seva informació.

Propostes de millora al Protocol d'Encaminament Anònim

Per tal d'optimitzar una mica més les operacions del protocol i les transmissions es proposa la inclusió d'un nou paràmetre en els paquets a transmetre que ajudarà a fer una gestió del *padding* dels missatges més eficient.

El Protocol d'Encaminament Anònim s'utilitza en dos modes de funcionament:

- *Publicació*: el node emissor envia unes dades a un node destí. El protocol funciona sota aquest mode quan s'executa l'aplicació de peer-review anònim en les operacions de: Publicació d'un treball i Introducció de la revisió d'un treball.
- *Obtenció*: el node emissor obté unes dades d'un node destí. El protocol funciona sota aquest mode quan s'executa l'aplicació de peer-review anònim en les operacions de: Obtenció d'un treball, i Obtenció de les revisions d'un treball.

1. Quan una entitat prepara els paquets a enviar a tots els nodes intermitjos d'una ruta fins a arribar a una Entity destí (punt 5 del protocol), incorpora en la informació per cada node intermig un valor de la modalitat de funcionament del protocol.

   $Paquet_x = Id\_Entity_x, E_{pbk\_x}(k_x, len_1), E_x(mode, role, Id\_Entity_{x+1}, Paquet_{x+1})$

   amb *mode* la modalitat de funcionament del protocol (0:*Publicació* o 1:*Obtenció*)

   En el punt 5 del Protocol d'Encaminament Anònim, quan es preparen els paquets a enviar, es configurarà un valor del *mode* aleatori <0,1>, independentment de la modalitat de funcionament real del protocol, quan es preparen els paquets de les següents Entities:
   - Entities que precedeixen a l'Entity destí en la cadena de *M* Entities.
   - Entities que segueixen a l'Entity destí en la cadena de *M* Entities sempre i quan la modalitat de funcionament sigui Publicació.

   En la resta de casos, el valor del *mode* serà el real:
   - *mode=0,* el protocol funciona en mode *Publicació*
   - *mode=1,* el protocol funciona en mode *Obtenció*

   El fet d'introduir el paràmetre *mode* en els paquets ens permet que els nodes intermitjos

d'una transmissió tinguin poca informació sobre l'operació que s'està fent però alhora que no destrueixin les dades a transmetre.

2. Quan una $Entity_x$ rep un Paquet, desxifra el text xifrat del missatge utilitzant la seva clau privada pvk_x, i comprova:
   - si $len_1=0$, $Entity_x$ és el node final de la transmissió i no ha de reenviar el paquet a ningú
   - si *role=origen*, $Entity_x$ és el node origen de la transmissió i es comporta tal com hem explicat en el punt 12.
   - si *role=router*, $Entity_x$ és un node retransmissor de la informació. El missatge que la $Entity_x$ ha rebut és:

     $$Paquet = Id\_Entity_x, E_{pbk\_x}(k_x, len_1), E_x(mode, role=router, Id\_Entity_{x+1}, Paquet_{x+1}), EC, <padding_x>$$

     El missatge que enviarà és:

     $$Paquet = Id\_Entity_{x+1}, Paquet_{x+1}, EC, <padding>$$

     El $padding_{x+1}$ que el node posarà el missatge serà:

     - si *mode=0*: qualsevol cadena $\neq padding_x$ (pot ser *null*)
     - si *mode=1*: $padding=E_x(padding_x)$

   - si *role=destiny*, $Entity_x$ és el node destí de la transmissió. En aquest cas l'$Entity_x$ comprova que l'usuari tingui autorització per fer l'operació que vol. En cas que l'usuari no estigui autoritzat, desar una traça indicant que l'usuari ha intentat fer una operació il·lícita i canviar el missatge d'error del paquet per *EC=7*:
     - si *mode=0*: agafa les dades <*data*> del paquet
     - si *mode=1*: Incorpora al paquet un $padding_x=E_x(data)$, amb *data* les dades que es volen enviar.

     Si l'operació ha anat bé, posar *EC=1*. En cas contrari, posar el codi d'error corresponent a la incidència ocurreguda (resta a l'estudiant definir els codis d'error).

3. Quan l'User Application rep el missatge de confirmació de transmissió, si el mode d'operació era *Obtenció* (*mode=1*), desxifrarà les dades rebudes fent:

   $$data = D_{x+t}(...D_{x+1}(D_x(padding)))$$

   amb *x* posició de la Entity destí, i *x+t* posició de l'Entity origen.